INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2010; **00**:1–6 *Prepared using* nemauth.cls *[Version: 2008/03/18 v1.00]*

Mitigating Unfairness in Locality-Aware Peer-to-Peer Networks

Frank Lehrieder[1,*], Simon Oechsner[1], Tobias Hossfeld[1], Dirk Staehle[1],
Zoran Despotovic[2], Wolfgang Kellerer[2], Maximilian Michel[2]

[1] *University of Würzburg, Institue of Computer Science, Würzburg, Germany*
[2] *DOCOMO Communications Laboratories Europe GmbH, Munich, Germany*

SUMMARY

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 to simultaneously increase the performance from the viewpoint of the P2P users, i.e., to 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 mostly assume homogeneous peer distributions. This is not the case in practice according to recent measurement studies. Therefore, we extend previous work in this paper by studying scenarios with real-life, skewed peer distributions. 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 heavily depends on the used locality-aware peer selection and the specific scenario. This contradicts the principle of economic traffic management (ETM) which aims for a solution where all involved players benefit and consequently have an incentive to adopt locality-awareness. Therefore, we propose and evaluate refinements of current proposals, achieving that all users of P2P networks can be sure that their application performance is not reduced. This mitigates the unfairness introduced by current proposals which is a key requirement for a broad acceptance of the concept of locality-awareness in the user community of P2P networks.

KEY WORDS: Peer-to-peer; locality-awareness; economic traffic management

1. 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 spent on optimizing such P2P-based content distribution networks [1, 2, 3, 4, 5]. In particular, these 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) [6] was

*Correspondence to: lehrieder@informatik.uni-wuerzburg.de

Contract/grant sponsor: This work has been performed in the framework of the EU ICT Project SmoothIT. The authors alone are responsible for the content of the paper; contract/grant number: FP7-2007-ICT-216259

established in November 2008 to standardize a protocol to guide the peer selection process and make it "better than random" as it is now.

Economic traffic management (ETM) is a recently emerging design principle which aims at traffic management solutions where all involved players participate willingly because those solutions provide incentives for collaboration [7]. In case of P2P networks, several players with different and partially contrary objectives are involved in this task. The two main groups of players are the P2P users, who desire high application performance, and the ISPs, which want to avoid unnecessary inter-domain traffic but want to keep revenue-generating users satisfied at the same time. As a consequence, optimization approaches need to be incentive-compatible, i.e., they have to provide incentives to every P2P user and every ISP to adopt or at least not object to an optimization approach. This is a key requirement for a broad acceptance and a successful deployment of a new traffic management scheme throughout the Internet.

For traffic optimization in P2P networks, locality-awareness is one of the most promising concepts. 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, 5]. 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 achieved by the users is sometimes increased while it remains almost unaffected in most of the investigated scenarios. This leads to the conclusion that locality-awareness creates a win-no lose in most cases and sometimes even a win-win situation.

This study is an extension of [8] where we show that a win-no lose situation is difficult to achieve under the real-life conditions we observe in today's Internet. In contrast to the above mentioned works, we find that some groups of peers suffer significantly in terms of decreased application performance, while others benefit from locality-awareness. The main reason is that the number of peers observed in different ASes varies heavily, i.e., a few ASes contain a large number of peers and most ASes contain only very few peers [9, 10], which is not taken into account by most of the aforementioned studies. Therefore, current locality-aware solutions are not incentive-compatible and a broad acceptance is questionable because it is difficult for the peers to know if they benefit in a specific scenario or not. We admit that further factors apart from the application performance may influence the users' decision, e.g. privacy. However, we exclude these considerations from this work since all studied mechanism have the same properties with respect to privacy as regular BitTorrent.

In this study we extend our previous work [8] in the following ways. First, we investigate whether disadvantaged peers can improve their performance by ignoring locality-awareness, even if those peers benefiting from it use locality-aware mechanisms. We show that these disadvantaged peers still have to implement locality-awareness to limit the negative effects on their application performance, although they fare worse than in a scenario with no locality-awareness. Second, we study scenarios where ISPs throttle inter-domain P2P traffic. Previous studies showed that locality-awareness creates a win-win situation in this case. We observe that all peers benefit from locality-awareness in this situation also in realistic swarms, and that therefore unfairness is no issue here. Third, we propose modifications of existing implementations of locality-awareness and show that our implementation does not discriminate some of the P2P users while it keeps the full potential of inter-domain traffic savings. In this way, we mitigate the unfairness in terms of unbalanced download times. Therefore, our solution can be seen as ETM since peers do no longer need to worry about decreased application performance owed to locality-awareness.

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 also the 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. 2 reviews related work. In Sect. 3 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. 4. In Sect. 5 we conclude the paper.

2. 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 on the limits of locality-awareness. Finally, we give a short overview of measurements of real BitTorrent swarms on which we base our evaluation scenarios.

2.1. 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 10000 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. In addition, scenarios with heterogeneous peer distributions are considered in [2]. The results show increased download times for peers in ASes with a large number of peers. To avoid that, the authors refine their original proposal by using a partition merging strategy. This shows that heterogeneous scenarios lead to different results and need to be considered when evaluating the performance of locality-awareness which is in line with our results in this study.

The P4P project [4] goes further and 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 [11]. 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 servers of content distribution networks such as Akamai as a landmarks to determine how close peers are.

Biased unchoking (BU) is a mechanism complementary to BNS proposed in [5]. 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. As in [1], the study is based on a homogeneous peer distribution and shows that BU inceases the effectiveness of BNS. Furthermore, 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 [5] is taken in [12] and the evaluation in a PlanetLab environment shows that download times can be slightly reduced on average.

The work presented in [13] 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 [9], [10], i.e, with heterogeneous peer distributions. We study their impact on the performance of a BitTorrent network for the P2P user and show that the peer distribution has a significant impact on the application performance. In particular, we show that certain user groups are penalized in terms of download speed. We explain which users can benefit from current locality-awareness solutions and which not. 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 specific implementation of locality-awareness and the scenario have a large impact on whether all or some users can benefit or not. Furthermore, we refine current proposals so that no user is disadvantaged by locality-awareness. In this way, we can achieve a guaranteed win-no lose situation and make locality-awareness conform to the ETM principle.

2.2. Limits of Locality-Awareness

In [14], 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 [14].

The Internet-draft "Mythbustering P2P Locality" [15] 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.

2.3. 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 [10]. For the measurements, movie and music .torrent-files have been downloaded from the Mininova and PirateBay index sites 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 largest 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 [9] 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.

3. 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. 2 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 [16] and [17].

3.1. 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*. A tracker is 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. Typically, a peer tries to have at least 40 neighbors. All peers keep their neighbors informed about which chunks of the file they already have. In this way, a peer knows in which neighbors 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 other peers are called *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.

3.2. 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 (BU).

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, 5], is to differentiate between peers in the 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 [18] or by contacting public databases such as [19]. More complex metrics, e.g., based on the configuration of the Border Gateway Protocol (BGP) [20] or on intra-AS topologies [4], are possible but beyond the scope of this study.

3.2.1. 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, 4, 11]. 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. In the first approach the tracker no longer returns random peers from the swarm. Instead, its 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 deprives the end user of his decision about promoting locality. Still, the tracker provider needs to cooperate with the ISP. We assume that this is unlikely to happen in practice if the tracker provider thereby acts against the interest of its users. Therefore, we believe that also in this scenario P2P-users have to be convinced of such a scheme.

Since the willing cooperation of the user in any locality-awareness approach is crucial for its broad acceptance, 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). The corresponding overhead remains small since only 26 byte per contact are required in the response of the tracker. Even for a large number of requested contacts this is negligible compared to the shared files which are typically several hundred MB in size. 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 is 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.

3.2.2. Biased Unchoking The biased unchoking (BU) mechanism evaluated here was presented in [5] and is specifically targeted at 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. If not mentioned differently, we set $l_{BU} = 0.9$. Again, that is a more conservative choice than in [5], but sufficient to show a negative impact. In addition, it leads to a similar degree of preference for local peers as $l_{BNS} = 0.9$ for BNS (cf. Sect. 3.2.1).

4. Performance Evaluation of Locality-Awareness Mechanisms

This performance evaluation differs from previous work mainly in the scenarios we consider. Therefore, we first describe the chosen settings and parameters. Like in [8] we show that locality-awareness leads to unbalanced application performance in case of real-world peer distributions. As an extension to [8], we study scenarios where only some of the peers are locality-aware. We show further that the unfairness can be reduced by less strict preference for local peers, but this is to the detriment of inter-domain traffic savings. Finally, we propose and evaluate a modification of current mechanisms that leads to more balanced download times without sacrificing inter-domain traffic savings for the ISPs.

4.1. Simulation Scenarios

The simulation setup is very similar to the one used in [5]. 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 [10], these are typical values for medium-sized swarms observed in practice. All peers are connected to their AS with an access speed of 16 Mbps downstream and 1 Mbps upstream, which are typical values for the DSL access technology.

The multi-AS network we simulate forms a star topology and consists of one transit AS and $n = 20$ stub ASes where every AS $k \in \{1, \ldots, n\}$ is connected to the transit AS via an inter-AS link (cf. Fig. 1). The tracker and the initial seed are placed in this transit AS for symmetry reasons. The transit AS does not contain any further peer besides the initial seed 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 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. If not mentioned differently, we model the inter-AS links as well dimensioned.

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 [9, 10]. [10] 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 [9] 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 than 300 peers the data presented in [9] 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

$$P(k) = \frac{1/k}{\sum_{i=1}^{n} 1/k}, k \in \{1, \ldots, n\} \tag{1}$$

which represents a common denominator of [9] and [10]. The aforementioned values presented in [10] 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 [9] 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'. Fig. 2 illustrates and compares the used distribution with an unrealistic homogeneous one, where the peer arrival process is equally distributed over all $n = 20$ stub-ASes.

The simulator used in this work is based on the P2P simulation and prototyping Java framework ProtoPeer [21, 22]. The simulator contains a flow-based network model adopting the max-min-fair-share principle [23]. On top of the ProtoPeer framework, we implemented the BitTorrent functionality and behavior as described in [16] and [17]. This implementation is available as an open-source library for ProtoPeer. It 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, between peers and the tracker as well as between the peers is simulated in detail. Our implementation is publicly available as a library for ProtoPeer [24].

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.

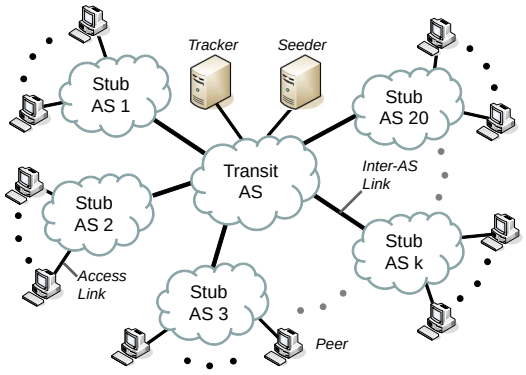


Figure 1: Simulated network topology consisting of one transit AS and $n = 20$ stub ASes. Every stub AS $k \in \{1, \ldots, n\}$ is connected to the transit AS which only forwards the traffic.

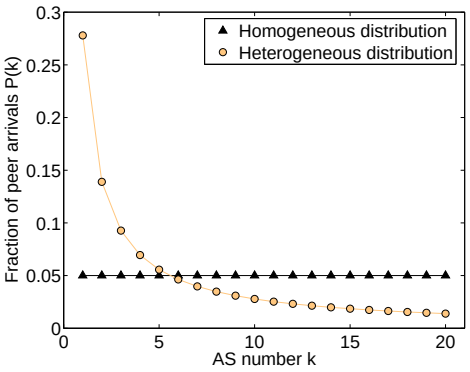


Figure 2: Fraction of peer arrivals with respect to the AS number k. In the heterogeneous case the fraction of peer arrivals decreases for larger values of k (cf. Eq. (1)), while it is constant in the homogeneous case.

4.2. Impact of Heterogeneous Peer Distributions

We start with the scenario where peers are distributed heterogeneously among the 20 ASes according to Eq. (1) as described in Sect. 4.1. The resulting average download time of the peers in the individual ASes is depicted in Fig. 3(a).

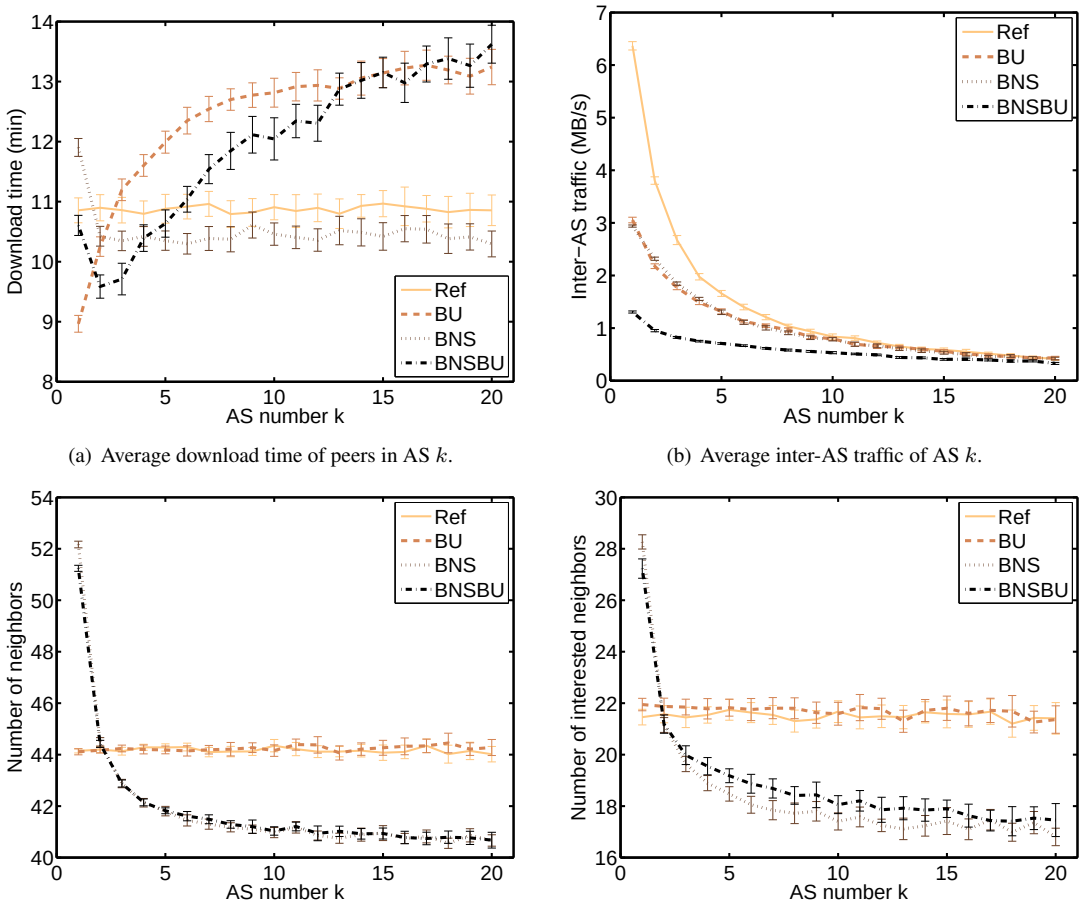
In a regular BT swarm, the AS affiliation of peers itself does not influence their download time in our scenario. Since all peers have the same access speed, this is only fair and an ideal state that should be preserved. Each peer contributes the same upload capacity and therefore experiences the same application performance. However, 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. To be more precise, in our chosen scenario, about 40% of the peers show a reduced download time, while about 50% take significantly longer to download the file on average.

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, containing about 28% of the swarm, 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. 3(c)) and therefore also more peers which are interested in downloading from them (Fig. 3(d)). 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. Thus, all locality-awareness mechanisms lead to a decrease in download performance for a share of the peers in the swarm.

Despite our focus on the user's perspective, we take a short look at the inter-AS traffic which we show as the sum of incoming and outgoing inter-AS traffic in all figures of this study. We do not differentiate between incoming and outgoing inter-AS traffic because it turned out that these are almost identical for all ASes in all studied scenarios. We can observe in Fig. 3(b) that the saving potential for inter-AS traffic grows with the share of peers in an AS. 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



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

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

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

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

Figure 3: Simulation results for the scenario with heterogeneous peer distribution and well-dimensioned inter-AS links.

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.

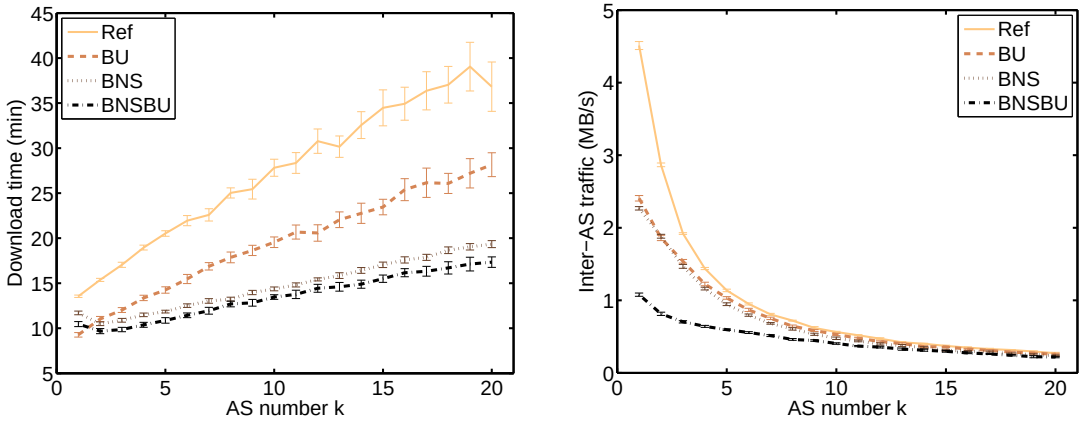
In contrast, ISPs with only a small number of peers per swarm are not likely to profit much from locality-awareness because there are only few options for peers in these ASes to choose local neighbors. The better part of such a peer's contacts have to be from remote locations even when it applies BNS. All these findings regarding the reduction of inter-AS traffic are in line with literature (cf. [5]).

4.3. *Impact of Bandwidth-Throttling or Inter-AS Bottlenecks*

In contrast to a homogeneous scenario with bottlenecks only in the access network, peers have been shown to benefit from locality-awareness if there are bottlenecks in the inter-AS links, cf. [5], [11]. These bottlenecks may be created artificially by a provider throttling P2P traffic or occur due to link or node failures and re-routed traffic. Here, locality-awareness helps to avoid the congested links and therefore leads to a higher download bandwidth. To judge whether this is still the case with a heterogeneous peer distribution, we limit the capacity of the inter-AS links in our simulation setup. To permit that all connections are throttled to the same degree, we set the capacity limit of the inter-AS links proportional to the average number of peers in the corresponding AS. As a reference value for AS 1 we use 24 Mbit/s which is slightly below the mean value observed in the Ref scenario without bottlenecks. This means the up- and downlink of AS 1 are each limited to a capacity of 24 Mbit/s, those of AS 2 are limited to 12 Mbit/s, and so on. This capacity allocation of inter-AS links gives every peer the same share of inter-AS traffic.

Fig. 4(a) shows the average download time per AS. While the download times are still distributed unfairly, every locality-aware mechanism improves the download time of the peers over the regular BitTorrent case ('Ref'). Therefore, every user gains from promoting locality, and the users in smaller ASes even save more time than the peers in larger ASes. Since regular BitTorrent prefers faster connections via the tit-for-tat strategy, and since enough local peers are available in the large ASes, locality promotion only leads to comparably small gains in terms of download speed here. Peers in smaller ASes, however, have no choice but to download content from outside their AS, and are thus slowed down significantly by the bandwidth restrictions. For these peers, utilizing the few local connections with relatively high bandwidth leads to a significant improvement in the application performance.

With respect to the traffic, the outgoing as well as the incoming inter-AS traffic of the individual ASes are limited by the bottleneck links here. Again due to the tit-for-tat strategy and enough local peers in the large ASes, the traffic to and from these ASes stays below the bottleneck capacity. While the peers in the smaller ASes can utilize local connections better by promoting locality, they still have



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

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

Figure 4: Impact of inter-AS bottlenecks on the application performance and inter-AS traffic.

to download a large share of the content from outside their AS. The locality-aware mechanisms thus can mainly save traffic for the large providers (cf. Fig. 4(b)), where they support the preference of a large number of local connections.

Thus, we conclude that all users have an incentive to implement locality-awareness under the conditions considered here. Still, the case with bottlenecks only in the access network is common in the Internet because most of the links in the core are well-dimensioned. Hence, unbalanced download times as presented in Sect. 4.2 remain a problem to be solved.

4.4. Impact of Locality-Awareness in Some ASes

The previous experiment without inter-domain bottlenecks (Sect. 4.2) shows that current locality-awareness approaches lead to a reduced application performance for a subset of the peers in the swarm. Therefore, we conclude that these peers have no incentive to promote locality. However, other peers that benefit from locality-awareness may still implement it. In the following, we return to the scenario without inter-AS bottlenecks and investigate whether it is feasible for a part of a swarm to promote locality, and what this means for the rest of the swarm.

To this end, we perform two experiments. First, we assume that the peers in only one AS use BNSBU and compare it to the default BitTorrent case. Second, we assume that the X biggest ASes already use BNSBU and investigate if peers in AS $X + 1$ should do so as well. For the sake of clarity we show only curves for BNSBU in the figures of this section and use the labels "ASX Locality" if only peers in AS X use BNSBU or "AS1-X Locality" if the X largest ASes apply the BNSBU locality-awareness mechanism.

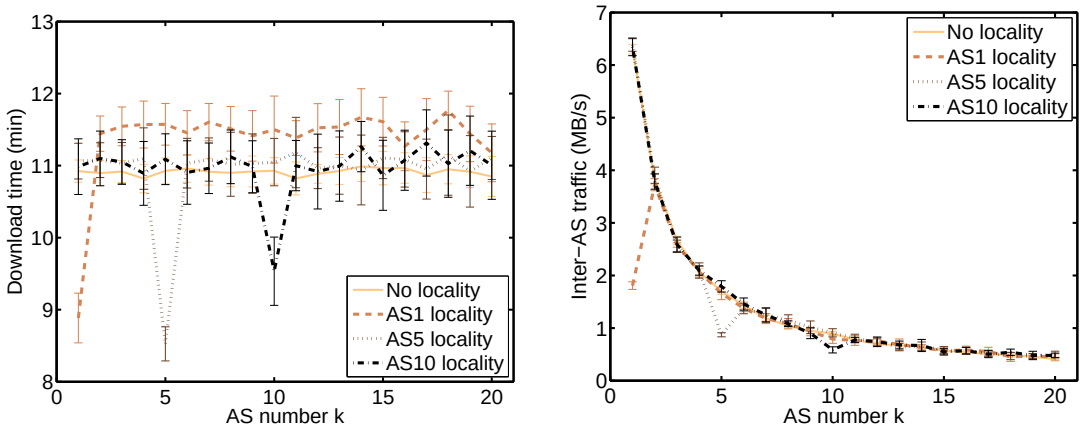
4.4.1. Locality-Awareness in a Single AS With this experiment we want to answer the question whether peers in a single AS can improve their performance through locality-awareness. For this purpose, we simulate scenarios where only the peers in one AS $k \in \{1, 5, 10\}$ use locality-awareness.

The peers in the AS which uses BNSBU reduce their average download time because they concentrate their upload capacity within the AS and receive additional upload capacity from outside. As a consequence, the average download time of peers in other ASes is increased. The impact seems to decrease for smaller ASes but there is no significant difference between "AS1 Locality" and "AS5 Locality" (cf. Fig. 5(a)). Regarding inter-AS traffic we observe in Fig. 5(b) that when peers in a single AS apply BNSBU, then the inter-AS traffic of that AS is significantly reduced while the inter-AS traffic of all other ASes remains almost unaffected.

These results show that peers promoting locality in a single provider's network are beneficial for both the provider and the peers themselves. In a swarm where no other peer implements locality-awareness, each subgroup of peers, even if it is small, has an incentive to do so.

4.4.2. Locality-Awareness in the X largest ASes As seen in the previous section, the peers in the largest AS ($k = 1$) can improve their download performance if they all use BNSBU. Consequently, peers in AS 1 have an incentive to do so. Since AS 1 holds a large share of the swarm, this leads to increased download times for peers in the other ASes. Thus, the implementation of locality-awareness in AS 1 has an effect on the application performance of peers that do not partake in the locality-promotion scheme. We now investigate whether this forces peers in other ASes to promote locality as well in order to mitigate this effect. To test this, we examine here whether there is an incentive for peers in the ASes $k \in \{2, 3\}$ to also use BNSBU.

If the top three ASes use BNSBU (labeled "AS1-3 locality" in Fig. 6(a)), they reduce their download



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

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

Figure 5: Locality-awareness in a single AS $k \in \{1, 5, 10\}$ compared to the BitTorrent reference implementation ('No Locality').

time for the same reason as described earlier. While the inter-AS traffic (Fig. 6(b)) of these ASes is reduced, it remains almost unaffected for the other ASes. Moreover, ASes 2 and 3 reduce their download time significantly in comparison to the case where only the peers in AS 1 promote locality. Similarly, the peers in AS 4 can reduce their average download time if they also apply BNSBU and this sequence can be continued for all other ASes 5, 6, 7, and so on (for the sake of readability we omit the corresponding curves in the figures). This means that in a scenario, where all ASes $Y, Y < X$ promote locality, the peers in AS X should apply BNSBU. However, there is a point when an AS still has a longer average download time in comparison to the Ref case even if it uses BNSBU. In other words, the peers in this AS can only reduce the negative effects of locality promotion in other ASes, but no longer benefit from it.

From these two experiments and the description of BNSBU we see that BNSBU tries to allocate as much as possible upload capacity to peers in the same AS while BT allocates upload capacity uniformly over all peers. Therefore, BNSBU shifts upload allocation to the local AS which leads to shorter download times for the peers in this AS. From this, we derive the following game theoretic interpretation. We identify the peers in one AS as one player which has the possible strategies "BNSBU" and "Ref". Since "BNSBU" keeps the upload capacity local, the payoff for "BNSBU" is larger (due to shorter download times) than the one for "Ref" and this holds for all players, i.e., ASes. As a consequence, the strategy "BNSBU" is a best response for every AS and every combination of the other ASes' strategies. Furthermore, the combination of strategies where all ASes use BNSBU is a (the only) Nash equilibrium where no player can improve its payoff alone. Hence, it is not possible for the peers in the small ASes to improve their performance by switching off BNSBU. On the contrary, the peers in the small ASes are "forced" by those in the large ASes to use locality-awareness although their situation would be better when all players played "Ref".

So far, we observed that locality-awareness can introduce unfairness in a swarm in terms of unbalanced download times. Moreover, we could establish that the peers that suffer from locality-awareness cannot prevent a degradation of their application performance by not implementing it. On

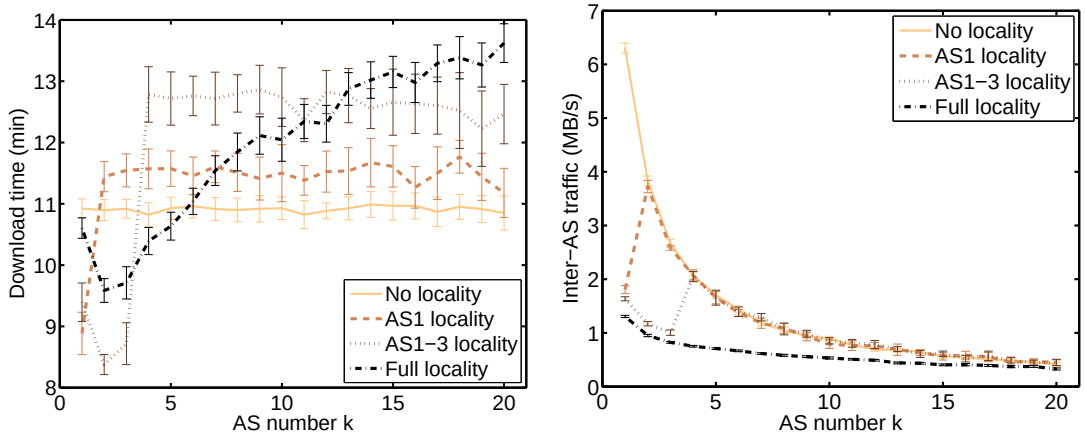
the contrary, if peers benefiting from promoting locality do so, then the rest of the swarm has to follow this scheme as well to reduce the negative consequences. This will lead to a reduced faith into such locality-awareness mechanism by users and therefore endangers its acceptance. In order to make locality-awareness incentive-compatible, a solution is required that does not introduce unfairness or that improves application performance for all users in comparison to the regular BT implementation. We will describe and compare approaches for this in the following.

4.5. Reducing the Degree of Locality-Awareness

First, we want to find out whether the degree of unfairness can be influenced simply by adapting 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.

The effect on the average download times for values of $l_{BU} \in \{0.5, 0.9\}$ is shown in Fig. 7(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. 4.2, 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. 7(b) shows that the inter-AS traffic 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.

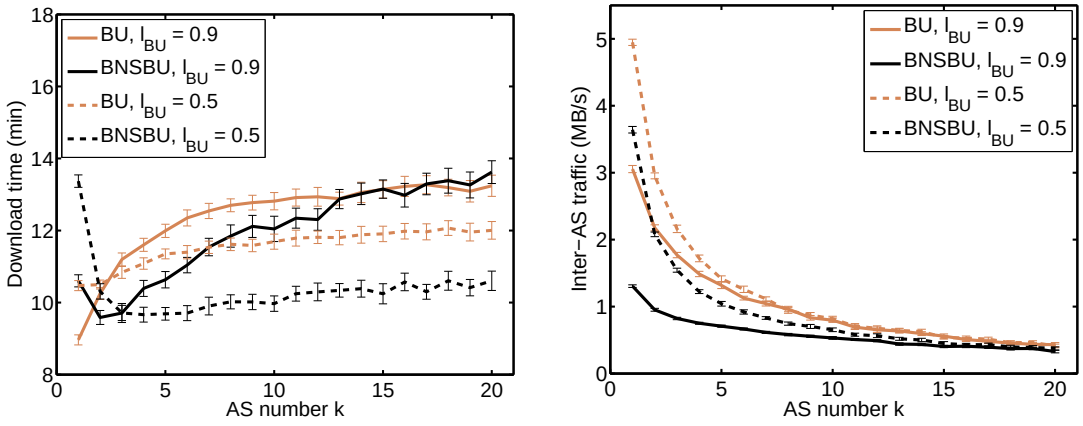
A similar effect can be achieved by reducing the degree of locality in the BNS mechanism. We compare the download times of the peers in different ASes for the locality-promotion schemes BNS



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

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

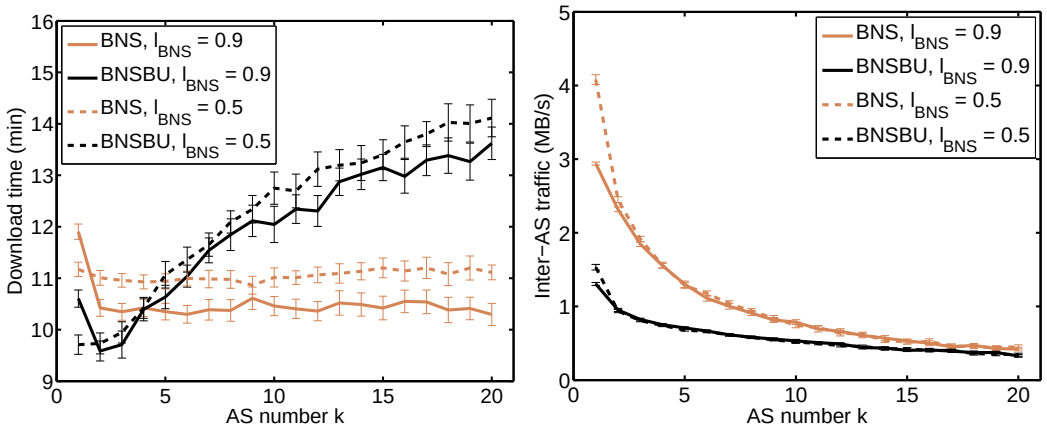
Figure 6: Locality-awareness in the X largest ASes compared to the BitTorrent reference implementation ('No Locality') and the scenario where all peers use BNSBU ('Full Locality').



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

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

Figure 7: 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 inter-AS traffic of AS k.

Figure 8: 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).

and BNSBU with the parameter $l_{BNS} \in \{0.5, 0.9\}$. The results are shown in Fig. 8(a). We observe that the significant increase in the download times for the largest AS vanishes for $l_{BNS} = 0.5$, i.e., a smaller 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. 8(b). In particular, the inter-AS traffic of AS 1 increases by about 30% (for BNS alone) or 20% (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 $l_{BNS} = 0.9$.

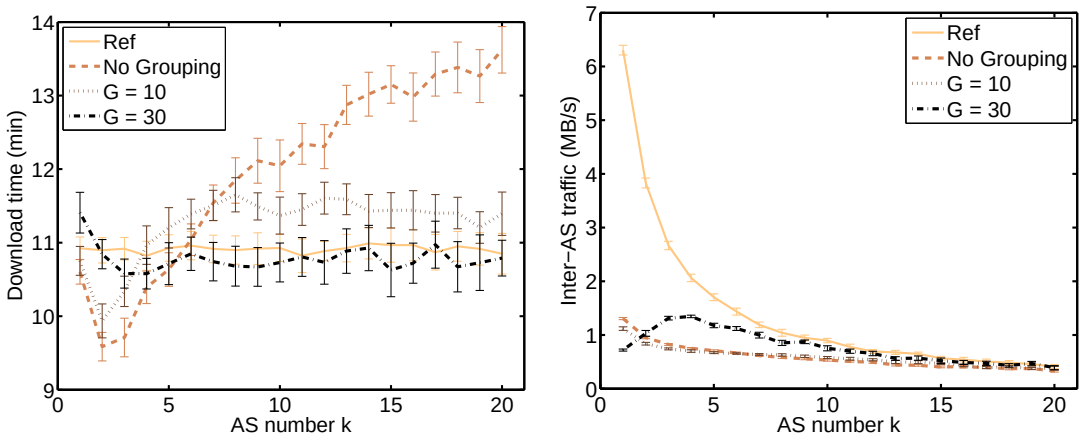
These results show that more conservative parameters might mitigate the negative effects of locality-awareness in terms of unbalanced average download times but reduce simultaneously the amount of inter-AS traffic that can be saved. Apart from this issue, there is still a significant difference in the download times of the peers with the most efficient traffic saving mechanism, BNSBU. Thus, we develop and investigate a different approach that reduces these drawbacks in the following section.

4.6. Grouping ASes

In this section we present another countermeasure for small ISPs to reduce the detrimental effects of locality awareness. The main reason why current locality-awareness approaches lead to unfairness in real swarms is that not all peers have the same number of potential local neighbors to select from. In the case of BU, this leads to a shift of upload capacity to large ASes. In the case of BNS, it leads to more neighbors for peers in large ASes and therefore less upload capacity for these peers.

To remedy this, we assign all ASes in a swarm to AS groups so that the number of peers in an AS group is limited by a given threshold G which we call "group size". For example, this could be done at an ALTO server [25] or SIS Server [7] which are intended to guide the peers in their peer selection process. For this purpose, these services need information about which peers participate in which swarms. In the BitTorrent protocol, this information could be retrieved via a communication with the tracker. We admit that this communication and the (periodic or request-based) process of grouping the ASes for every swarm could be quite a heavy burden for the locality information services. However, the AS grouping can also be shifted to the peers and be performed there in a decentralized way. For the peer-based BNS, a peer already knows the IP addresses of all other peers in the swarm or at least a large fraction of them. Via public databases, e.g. [19], each peer can perform the grouping of ASes on its own. This might lead to different AS groups at different peers and different points in time but it still achieves that a peer has a minimum number of other peers in its AS group. In this way, it prevents certain peers from being much more popular than others. For the sake of simplicity we perform the AS grouping at a central service in our simulations. This service communicates with the tracker and creates AS groups for every request in the following way. It starts the process with the smallest unassigned AS and adds the next larger AS to the group as long as less than G peers are within the AS group. This is repeated until all ASes are assigned to an AS group. The service signals to the requesting peers which other peers are in the same AS as this peer and which are in the same AS group. In the following, we describe and evaluate two possibilities how peers can include this information in their peer selection process.

4.6.1. Group-Based Peer Selection The first option to use this information within the peer selection processes BNS and BU is to simply treat peers of the same AS group as local peers and apply BNS and BU as described in Sect. 3. This avoids that peers in small ASes are allocated less bandwidth than those in large ones due to BU. The upload bandwidth is shared predominantly among a group, and the group can be chosen large enough so that there are always candidates for preferential unchoking. We test this by evaluating the previous scenario with the locality-aware BNSBU mechanism together with the grouping approach for group sizes $G \in \{10, 30\}$. Fig. 9(a) shows that the unfairness of BNSBU in terms of unbalanced download times is significantly reduced and almost eliminated for a group size of $G = 30$ peers or larger (not shown in the figures). However, a group size of 10 leads to cases with too few interested peers in the AS group to efficiently prefer them in the unchoking process. Thus,



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

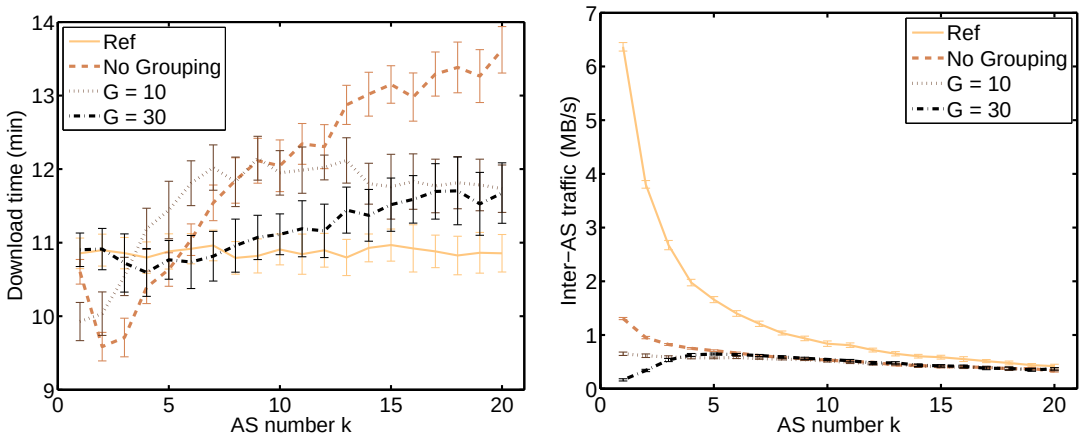
(b) Average inter-AS traffic of AS k.

Figure 9: Impact of group size $G \in \{10, 30\}$ for the group-based peer selection.

unfairness is reduced here but not eliminated. However, since now local peers are treated with the same priority as grouped peers, the traffic savings are affected. This can be seen in Fig. 9(b), where the inter-AS traffic of each AS is compared for the pure locality case, denoted by 'No Grouping', and the case with peer grouping. Slightly more inter-domain traffic is caused by the grouping approach, with larger traffic amounts being caused by larger groups. This is due to the fact that a larger group leads to a higher probability for a grouped peer to be unchoked instead of a local peer. However, the saved traffic is still significant in comparison to the reference BitTorrent case for all considered group sizes. Only the largest AS is the major exception, and saves more traffic. Since less peers upload into this AS, the incoming traffic is decreased in comparison to the case with no grouping. Due to the tit-for-tat algorithm, this results in slightly less upload traffic as well.

4.6.2. Strict Preference of Local Peers While we have achieved the goal of providing a fair service to all end users with the algorithm described above, we want to find out whether we can optimize this approach with respect to the inter-AS traffic. To this end, we define three categories of neighbors: local, grouped, and remote. Grouped neighbors are those located in an AS of the same AS group but not in the same AS. As our implementations of BNS and BU rely on the distinction between local and remote neighbors, we need to modify them slightly because we have to distinguish three different types of neighbors now. With BNS we try to fill 90% of the neighbor set first with local and then with grouped peers. With BU, we cannot use the probabilistic approach with those three groups. Therefore, we adapt the unchoking policy so that the highest ranked interested peer is always unchoked, i.e., a peer unchokes a local neighbor with probability $l_{BU} = 1$ if present. If not, it unchokes a grouped neighbor if present. If not, it unchokes a remote neighbor. Hence, as much as possible data is uploaded to local peers in contrast to the aforementioned group-based peer selection.

As a result of this stricter preference, the traffic savings increase in comparison to the case without groups for all considered group sizes and the largest ASes, cf. Fig. 10(b). This traffic optimization comes again at the price of unfairness, cf. Fig. 10(a). The peers in the smallest ASes take again longer to download the file, since now the same effect described for the original locality-aware mechanisms



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

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

Figure 10: Impact of group size $G \in \{10, 30\}$ for the peer selection with a strict preference for local peers.

occurs within the individual groups. Larger ASes within a group profit more in terms of bandwidth than smaller ones. However, since the groups are more homogeneous than the swarm as a whole, the consequences for the peers are less severe. Thus, the average download times of the ASes are distributed fairer than with pure BNSBU. These experiments show that current locality-awareness solutions can be improved by grouping ASes into groups of roughly the same size. This might increase implementation complexity, but it leads to more balanced download times. Still, a trade-off exists between balanced download times and the amount of saved inter-AS traffic. However, the AS grouping solutions mitigates the negative effects significantly.

5. Conclusions

In this study we considered the impact of locality-awareness mechanisms in P2P networks on content distribution from the P2P user's point of view. For our evaluation, we used BitTorrent as a well-known example P2P application and investigated its performance when *biased neighbor selection* and *biased unchoking*, including a third method combining the two, are used as locality promotion mechanisms. We questioned these established locality promotion mechanisms and investigated them in real-world settings.

In particular, we checked the impact of realistic, skewed peer distributions on the application performance for the mentioned locality promotion mechanisms. We showed that a win-no lose situation for ISPs and all P2P users is difficult to achieve in practice with the current locality-promotion proposals because in most cases (without inter-domain bottlenecks) some of the P2P users benefit to the detriment of other P2P users. What is the case for a specific user depends strongly on the implementation of the locality-aware peer selection mechanism and the properties of the swarm. Furthermore, we showed that peers disadvantaged by locality-awareness cannot improve their situation by using legacy BitTorrent algorithms. Instead, they are forced to promote locality as well. Finally, we proposed to group ASes and showed that this mitigates the effect of unbalanced download times in our

scenarios. At the same time, it retains the full potential of inter-domain traffic savings for the ISPs. This is an important step towards a broad acceptance of locality-awareness in the P2P user community because P2P users can be sure that their AS affiliation has no impact on their application performance.

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 *Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS)*, 2006.
2. Stevens Le Blond, Arnaud Legout, and Walid Dabbous. Pushing BitTorrent Locality to the Limit. Technical report, PLANETE - INRIA Sophia Antipolis / INRIA Rhône-Alpes - INRIA, 2009.
3. Vinay Aggarwal, Anja Feldmann, and Christian Scheideler. Can ISPs and P2P Systems Co-operate for Improved Performance? *SIGCOMM Comput. Commun. Rev.*, 37(3):29–40, 2007.
4. Haiyong Xie, Richard Y. Yang, Arvind Krishnamurthy, Yanbin G. Liu, and Abraham Silberschatz. P4P: Provider Portal for Applications. *SIGCOMM Comput. Commun. Rev.*, 38(4):351–362, 2008.
5. Simon Oechsner, Frank Lehrieder, Tobias Hoßfeld, Florian Metzger, Konstantin Pussep, and Dirk Staehle. Pushing the Performance of Biased Neighbor Selection through Biased Unchoking. In *Proc. IEEE Int'l Conf. Peer-to-Peer Computing (P2P)*, Seattle, USA, 2009.
6. Charter of the IETF Working Group on Application-Layer Traffic Optimization (ALTO). http://www.ietf.org/html.charters/alto-charter.html.
7. Tobias Hoßfeld, David Hausheer, Fabio Hecht, Frank Lehrieder, Simon Oechsner, Ioanna Papafili, Peter Racz, Sergios Soursos, Dirk Staehle, George D. Stamoulis, Phuoc Tran-Gia, and Burkhard Stiller. An Economic Traffic Management Approach to Enable the TripleWin for Users, ISPs, and Overlay Providers. In *Towards the Future Internet - A European Research Perspective, ISBN 978-1-60750-007-0*, page 24. IOS Press Books Online, 2009.
8. Frank Lehrieder, Simon Oechsner, Tobias Hoßfeld, Zoran Despotovic, Wolfgang Kellerer, and Maximilian Michel. Can P2P-Users Benefit from Locality-Awareness? In *Proc. IEEE Int'l Conf. Peer-to-Peer Computing (P2P)*, Delft, the Netherlands, 2010.
9. Haiyang Wang, Jiangchuan Liu, and Xu Ke. On the Locality of BitTorrent-based Video File Swarming. In *Proc. Int'l Workshop on Peer-to-Peer Systems (IPTPS)*, 2009.
10. Tobias Hoßfeld, David Hock, Simon Oechsner, Frank Lehrieder, Zoran Despotovic, Wolfgang Kellerer, and Maximilian Michel. Measurement of BitTorrent Swarms and their AS Topologies. Technical Report 463, University of Würzburg, 2009.
11. 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.*, 38(4):363–374, 2008.
12. Shansi Ren, Enhua Tan, Tian Luo, Songqing Chen, Lei Guo, and Xiaodong Zhang. TopBT: a Topology-Aware and Infrastructure-Independent BitTorrent Client. In *Proc. IEEE Int'l Conf. Computer Communications (INFOCOM)*, 2010.
13. Bo Liu, Yi Cui, Yansheng Lu, and Yuan Xue. Locality-Awareness in BitTorrent-Like P2P Applications. *IEEE Trans. Multimedia*, 11(3):361–371, 2009.
14. Michael Piatek, Harsha V. Madhyastha, John P. John, Arvind Krishnamurthy, and Thomas Anderson. Pitfalls for ISP-Friendly P2P Design. In *Proc. ACM Workshop on Hot Topics in Networks (HotNets)*, 2009.
15. Enrico Marocco, Antonio Fusco, Ivica Rimac, and Vijay K. Gurbani. Mythbustering Peer-to-peer Traffic Localization. IETF Internet draft (work in progress), http://tools.ietf.org/html/draft-marocco-p2prg-mythbustering-01, 2009.
16. Arnaud Legout, Guillaume Urvoy-Keller, and Pietro Michiardi. Rarest First and Choke Algorithms Are Enough. In *Proc. ACM Internet Measurement Conf. (IMC)*, pages 203–216, New York, NY, USA, 2006.
17. Bittorrent specification. http://wiki.theory.org/BitTorrentSpecification.
18. The SmoothIT project. http://smoothit.org/, 2008.
19. Team cymru. http://www.team-cymru.org/.
20. Peter Racz, Simon Oechsner, and Frank Lehrieder. BGP-based Locality Promotion for P2P Applications. In *International Conference on Computer Communication Networks (ICCCN)*, Zurich, Switzerland, August 2010.
21. Protopeer. http://protopeer.epfl.ch/index.html.
22. Wojciech Galuba, Karl Aberer, Zoran Despotovic, and Wolfgang Kellerer. ProtoPeer: A P2P Toolkit Bridging the Gap Between Simulation and Live Deployment. In *Proc. Int'l Conf. Simulation Tools and Techniques (SIMUTools)*, 2009.
23. Dimitri Bertsekas and Robert Gallagher. *Data Networks*. Prentice Hall, 1987.
24. BitTorrent Library for Protopeer. http://protopeer.epfl.ch/wiki/BitTorrent.
25. Sebastian Kiesel, Laird Popkin, Stefano Previdi, Richard Woundy, and Yang Richard Yang. Application-Layer Traffic Optimization (ALTO) Requirements. IETF Internet draft (work in progress), http://datatracker.ietf.org/doc/draft-ietf-alto-reqs/, 2010.