University of Würzburg
Institute of Computer Science
Research Report Series

Measurement of BitTorrent Swarms and their AS Topologies

Tobias Hoßfeld, David Hock, Simon Oechsner,
Frank Lehrieder, Zoran Despotovic, Wolfgang Kellerer,
Maximilian Michel

Report No. 464 January 2010

[1] University of Würzburg, Institute of Computer Science, Chair of Distributed Systems
Am Hubland, 97074 Würzburg, Germany
hossfeld@informatik.uni-wuerzburg.de

[2] DOCOMO Communications Laboratories Europe GmbH
Munich, Germany
despotovic@docomolab-euro.com

Acknowledgments. This work has been funded partially in the framework of the EU ICT Project SmoothIT (FP7-2007-ICT-216259) and by the Federal Ministry of Educations and Research of the Federal Republic of Germany (Förderkennzeichen 01 BK 0800, G-Lab). The authors alone are responsible for the content of the paper.

Measurement of BitTorrent Swarms and their AS Topologies

Tobias Hoßfeld, David Hock,
Simon Oechsner, Frank Lehrieder
University of Würzburg, Institute of Computer
Science, Chair of Distributed Systems
Am Hubland, 97074 Würzburg, Germany
hossfeld@informatik.
uni-wuerzburg.de

Zoran Despotovic, Wolfgang Kellerer,
Maximilian Michel
DOCOMO Communications Laboratories
Europe GmbH
Munich, Germany
despotovic@docomolab-euro.com

Abstract

The optimization of overlay traffic resulting from overlay applications such as BitTorrent is a challenge addressed by several recent research initiatives. Locality awareness for the selection of peers is considered as the straight forward solution in order to reduce network traffic and shorten download latency. In most related work a simple approach is taken limiting the number of links pointing outside an autonomous system, while important overlay characteristics as peer distribution over autonomous systems and time dynamics are neglected in the corresponding performance evaluations. In this work, we address this lack of realistic swarm statistics by providing our measurement results revealing real live BitTorrent swarm characteristics.

1 Introduction

Overlay traffic resulting from applications such as BitTorrent emerges as a high burden for network operators today. The problem arises of how to effectively control and manage such traffic stemming from end-to-end overlay applications from within the network. Recently this challenge is addressed by research initiatives like SmoothIT [1], P4P [2] and Oracle [3]. Though solutions are still at an early stage, its importance has already triggered standardization and the IETF working group ALTO (Application Layer Traffic Optimization) has been founded in November 2008.

Locality awareness is considered as a straight forward solution by all current research initiatives. Traffic generated by overlay applications typically crosses borders of a network operator domain (a so called autonomous system, AS) multiple times and thus causes high cost for the network provider. It is also subject to lower quality of service such as a longer delay. The concept of locality awareness is to optimize the traffic flow with information about the location of a content providing peer in the underlying network. For example, any peer might be provided with a list of peers for download that are marked according to the position inside or outside the AS of the requesting peer. Thus, quality of service and network usage can be optimized at the same time, for the benefit of the overlay application and the network provider.

Whereas locality awareness has shown its benefits in several, mostly simulation-based experiments in a controlled environment, the question arises whether a simple locality awareness approach such as sketched above is sufficient and whether locality awareness can be utilized efficiently in real network environments. Most evaluations of current approaches rely on a set of conditions which let those approaches appear as very efficient. However, the measurements that we present in this paper show that these conditions are rarely met in real networks. In particular, we corroborate and extend known results that the distribution of peers among AS's is highly skewed which means that only a very small number of peers is present in most of the AS's. This has a strong impact on the efficiency of locality awareness because the possibilities of keeping traffic within these AS's is limited. Thus, current state of the art traffic optimization solutions lack real-world scenarios to be evaluated in. To alleviate that problem, we present a large-scale measurement campaign based on BitTorrent in this paper.

The contribution of this paper is (1) providing the results on our comprehensive measurement study of BitTorrent swarms and (2) a characterization of BitTorrent swarms relevant for the quality of locality awareness solutions. The measurement results comprise a comprehensive set of swarms for different types of content listed at mininova.org and piratebay.org. We have measured the swarm size, swarm dynamics in terms of number of leechers and seeders, and the AS topology of swarms. We have also analyzed the details of individual swarms to understand content clustering (e.g., availability of certain content in specific regions only). The measurements have been performed from June 2008 to May 2009 using the PlanetLab and G-Lab experimental facilities. The characterization shows for example that real-life BitTorrent swarm distributions are highly skewed (i.e., 90% of the AS's have less than ten peers, mostly just one) and demand for more differentiated algorithms for traffic optimization. This is in contrast to most previous work where uniform distributions of peers over AS's are assumed.

The remainder of this paper is organized as follows. Related work is discussed in Sec. 2. We first describe the measurement setup in Sec. 3 and provide the measurement results and the derived characteristics for BitTorrent swarms in Sec. 4. From these results, we summarize the main observations for modeling BitTorrent swarms in Sec. 5 and conclude their relevance for traffic optimization mechanisms in Sec. 6.

2 Related work

P2P has a dominant share in the total Internet traffic [4]. The main contributor to this share is still BitTorrent [5]. Thus, it becomes critical to understand all aspects of BitTorrent's behavior, including both properties of individual swarms and global statistics, and provide good sources of information for its modeling. This is one contribution of our paper. Studies such as [6] and [7] do exactly this. [6] follows the lifetime of one specific torrent and analyzes BitTorrent's main performance indicators (e.g., download times). Besides examining its download performance, [7] makes a step further toward providing measurements useful for modeling of BitTorrent. Peer uptime distribution, their bandwidth distribution, peer arrival process properties as well as distribution of seeders across time are the main quantities [7] focuses on. The set of properties in the focus of our paper is not overlapping with these, i.e., the information we provide is complementary to the information provided by [7].

We pay considerable attention in this paper to the distribution of peers in BitTorrent swarms across AS's. The motivation for doing so is as follows. BitTorrent forms its overlay graphs and distributes content unaware of the underlying physical network properties. The same piece of a file can bounce back and forth between different peers in Europe and America creating thus enormous amount of unnecessary traffic. Given the scale of the problem, i.e., BitTorrent's immense popularity, it becomes important for network owners to manage BitTorrent traffic efficiently. At the same time, being network-agnostic, BitTorrent might be offering suboptimal performance as seen from its users point of view. Solving the two problems simultaneously (handling P2P traffic efficiently and improving performance for the end user) has become an important research area recently. Locality promotion has been so far suggested in the literature as the main solution class. It requires peers to give preference to selecting neighbors from the same AS rather than those outside the AS when forming the overlay graph. Bindal et al. [8] and Aggarwal et al. [3] were the first to analyze how locality promotion can help reduce the generated traffic and improve the performance of BitTorrent and Gnutella, respectively. They both find serious improvements of the application performance (i.e., reduction of download times) and reduction of cross-ISP traffic. [9] essentially repeats the experiments of [8] on a larger scale and comes up with the conclusion that locality can be pushed to the limit, i.e., only a minimum necessary inter-AS links can be kept, while all others should be maintained toward local peers, i.e., those within the same AS.

A similar approach is taken in [10]. Wang et al. studied around 70,000 BitTorrent swarms from btmon.org-BitTorrent site for 6 months in 2008, using 200 PlanetLab nodes with a customized BitTorrent client to retrieve the swarms' peer IP addresses. These IP addresses were run against the whois-service to resolve the IPs' autonomous systems. The paper mainly concentrates on swarms distributing video files, stating that video files show the highest regional (AS) interest, i.e. Chinese movies are mostly watched in China. The authors analyze the distribution of peers to AS's and conclude that in small swarms the application of locality awareness mechanisms is not useful, because the top AS of the swarm holds a large fraction of the whole swarm and the traffic is already naturally localized. On the other hand in large swarms the authors found no AS holding more than 6% of the whole swarm population, which makes the application of locality enhancements more favourable. Furthermore they find for large swarms, that the relation between ordered ASes of a swarm and the AS-fraction of a swarm (i.e., x-largest AS of a swarm – #peers in AS/#peers in swarm) follows the Mandelbrot-Zipf distribution. Eventually the paper argues, that AS's have a stationary property of forming a larger cluster within a swarm, and give a probabilistic approach how to predict the peers' membership in a large cluster. Peers in large clusters should apply locality aware neighbor selection, peers not in large cluster should stay with the standard random neighbor selection. In contrast to this paper, we consider more media types in our measurement and also cover more swarms from different torrent indexes. This allows us to generalize the results and to identify subgroups with special characteristics. Thus, we also provide a more differentiated view on regional content, which is mentioned in [10] but not considered in detail. Especially, we show that the share of peers in one AS can be larger for regional content.

3 Measurement Setup

The measurements described in this section aim at gathering data about live BitTorrent swarms from which we want to draw conclusions about the viability of locality awareness. First, we outline the BitTorrent protocol itself before introducing our measurement methodology.

3.1 The BitTorrent Protocol

BitTorrent's objective is to disseminate one large file to a large number of users in an efficient way. For each file an overlay network called *swarm* is created. According to the original BitTorrent specification, each overlay network consists of two different kinds of peers, the seeders and the leechers, and a so-called tracker. A *seeder* is a peer in the swarm that holds the complete file and uploads to others altruistically, whereas a *leecher* is still downloading the file.

For each swarm, a centralized component, the so-called *BitTorrent tracker*, stores information about the file itself and all peers in the swarm. This information includes the file size, the number of seeders and leechers, as well as the IP addresses of the peers. A peer joining the network asks the tracker for a list of active peers in the overlay. The tracker then returns (a) the number of seeders S and leechers L and (b) a random subset of k peers, i.e., k different IPs, to the requesting peer. Most trackers return $k = 50$ peers per default.

In order to avoid congestion at the tracker, the request rate of an individual peer is limited. The default value in the original BitTorrent tracker implementation from Cohen allows a single request every 5 min. However, in the Internet, various tracker implementations exist and in our measurements we have been able to contact various trackers every 10 s, if necessary.

For searching files to download through the BitTorrent protocol, there are several websites that list indexes and directories of `.torrent` files. Such a website is referred to as *torrent index*. A torrent index maintains a list of `.torrent` files containing metadata about the files to be shared and about the tracker, as well as additional information about the popularity of a file (in terms of number of seeders and leechers) or the date when the file was published.

3.2 Conducted Experiments

To gain a more differentiated view on the characteristics of existing swarm types than in the known work, we chose specific sets of swarms to measure. These are defined by a number of selection criteria which serve to define a number of swarm classes. In contrast to [10], we do not only want to analyze swarms found on one index and only distributing videos. Instead, we want to expand the insights gained from observing these swarms to other classes of swarms as well. According to a certain selection criterion and the desired type of content, the `.torrent` files are downloaded from a torrent index. As *selection criteria*, we consider (a) all available torrents, (b) the most popular torrents in terms of number of peers in the swarm, and (c) the most recent files which have been published in the last 24 hours. As *type of content*, we distinguish between (1) music files, (2) TV series, (3) movies, (4) so-called "regional" movies which are in a certain language (German, Spanish, French, Italian, Dutch), and (5) all media independent of the type of content. The considered *torrent index servers* cover the most popular ones in the Internet, (i) PirateBay, (ii) Mininova, and (iii) Demonoid. Here, the criteria (a)(3) and (a)(4) correspond to the class of swarms evaluated in [10]. Thus, we additionally consider other content types and indexes as well as specific subsets of swarms.

Table 1 summarizes the measurement experiments conducted over the period from June 2008 to May 2009. Each measurement experiment is assigned a unique identifier `ID`. which is used when describing the measurement results. In particular, we measure in each experiment the swarm size, the swarm dynamics, and the AS topology of swarms meaning the affiliation of peers to AS's. In order to measure the total number N of peers in a swarm and their corresponding AS's, we contacted the tracker and requested a list of peers. As a result, the number of seeders S and leechers L, and a set of k different IP addresses of peers are returned.

Since a tracker typically returns $k = 50$ IP addresses for a single request, we used a large number of machines with BitTorrent clients running on each of them. They contact the tracker simultaneously in order to get the IP addresses from all peers in the swarm at a single time instant, i.e. a snapshot of the swarm. In particular, several requests are sent within 5 minutes from all 219 nodes in PlanetLab [11] and 153 nodes in G-Lab [12], respectively, until $N = S+L$ different IP addresses are obtained. Then, the IP addresses are mapped to the origin AS using the RIPE database (`http://www.ripe.net/projects/ris/tools/riswhois.html`). This measurement method is referred to as *distributed monitoring* in the remainder of the paper. However, for measuring the swarm size only, it is sufficient to monitor the tracker (denoted as 'tracker monitored' in Table 1 for setups `Pop`. and `24h`.) or to parse the website of the torrent index ('website parsed'), as done in experiment `TV`. Additionally, we consider a publicly available dataset from Khirman [13] with measurement results of the swarm sizes of torrents on different torrent index servers (`KPi`., `KDe`., and `KMi`.).

To study the time dynamics of a swarm, several samples of the swarm size and the AS topology are captured over a longer period of time which is denoted as "xx samples every yy hours" instead of "snapshot" in the column "measurement per swarm" in Table 1. In that case, for example the average swarm size over this period of time is given, which may result in a decimal number, while a snapshot of a swarm always returns an integer value.

Table 1: Overview on conducted measurement setups.

ID	torrent index	selection criteria	type of content	meas. per swarm	#torrents	metho-dology	observed	meas. date
TV.	PirateBay	all available	TV series	96 samples over 36 hours	63,867	website parsed	swarm size	Jun. 2008
Pop.	PirateBay	most popular	movies	snapshot	4,463	tracker monitored	swarm size	Mar. 2009
24h.	PirateBay	last 24 hours	all media	snapshot	1,048	tracker monitored	swarm size	Mar. 2009
Grp.	Mininova	groups w.r.t. size & language	movies	440 samples over 88 hours	16	distributed monitoring	AS topology	Apr. 2009
Mov.	Mininova	all available	movies	snapshot	126,050	distributed monitoring	AS topology	Apr. 2009
Mus.	Mininova	all available	music	snapshot	135,679	distributed monitoring	AS topology	Apr. 2009
Reg.	PirateBay	top 30	regional movies	snapshot	120	distributed monitoring	AS topology	May 2009
KPi.	PirateBay	all available	all media	snapshot	1,682,355	data taken from [13]	swarm size	Mar. 2009
KDe.	Demonoid	community selected titles	all media	snapshot	11,759	data taken from [13]	swarm size	Mar. 2009
KMi.	Mininova	legal torrents promotion	all media	snapshot	4,514	data taken from [13]	swarm size	Mar. 2009
Ele.	open movie “Elephants Dream”			8,640 samples over 24 hours	1	distributed monitoring	AS topology	Apr. 2009

3.3 Distributed Monitoring of Tracker

The distributed monitoring of a BitTorrent tracker for obtaining the AS topology relies on experimental facilities, like PlanetLab or G-Lab, with a large number of nodes. They are controlled by a central unit C which is located at the University of Wuerzburg in our measurements. C has established connections to the used PlanetLab and G-Lab nodes Ω. C is responsible for the distribution of the `.torrent` files to these monitoring nodes Ω, the initialization of the monitoring on Ω and the collection of the created result files from Ω. The monitoring on each node itself is realized with a python script that queries a tracker n times every t seconds. In our measurements, t is set to 15 seconds to avoid overloading the tracker, while n is chosen according to N, using the analysis described below.

Figure 1 shows the number of occurences X of the same IP address in a measurement trace. The random variable X can be approximated by a binomial distribution $X \sim BINO(n, q)$, when the tracker of a swarm of size N is requested n times and returns 50 IP addresses each time, i.e. $q = 1 - (1 - 1/N)^{50}$.

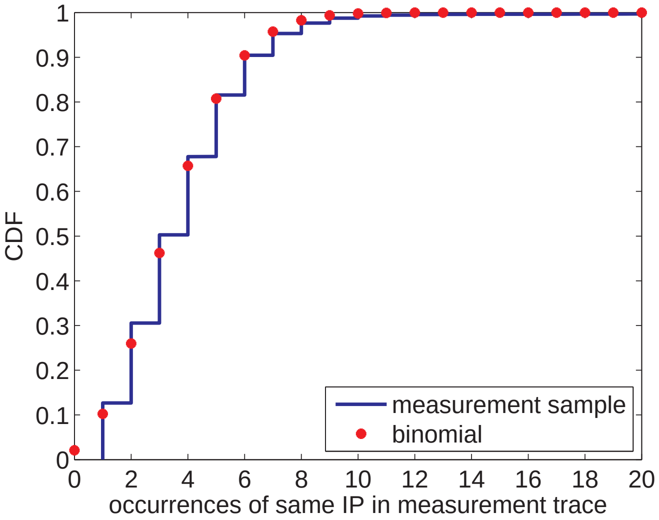

Figure 1: Occurences X of the same IP address in a measurement sample follow a binomial distribution $X \sim BINO(b, q)$, when the tracker of a swarm of size N is requested b times and returns 50 IP addresses each time, i.e. $q = 1 - (1 - 1/N)^{50}$.

In the following, we derive the number Y of required monitoring nodes in order to obtain all IP addresses of N peers in a swarm. Upon each request, the tracker returns a subset of $k = 50$ peers which are randomly chosen from all N peers. Denote by X the number of times the tracker has to be contacted to get N different IP addresses. The derivation of X is known as the *coupon collector's problem* [14]. In [15], we derived an exact solution which is given in the following.

Let $P(j, i)$ denote the probability to observe j different IPs after the i-th tracker response. It is

$$P(j, i) = 1 \text{ for } j \leq k \text{ and } i > 0, \qquad (1)$$

since the first tracker response returns k different IPs. It is

$$P(j,i) = 0 \text{ for } j > \min(ik, N), \tag{2}$$

since a maximum of ik different IPs are retrieved after the i-th tracker response and there are only N different IPs.

This allows to recursively compute $P(j,i)$ for all other cases,

$$P(j,i) = \sum_{m=0}^{k} \frac{\binom{j-m}{k-m} \cdot \binom{N-j+m}{m}}{\binom{N}{k}} \cdot P(j-m, i-1)\,, \tag{3}$$

which simply considers the number of possibilities to obtain $k-m$ old and m new IPs, normalized by the number of possibilities for k different IPs of a tracker response. As a result, we obtain the distribution X of the number of required tracker responses to get all N IPs which is $P(X=i) = P(N,i)$.

An upper bound of the average number of required tracker responses

$$E[X] = \sum_{i=0}^{\infty} iP(N,i) \tag{4}$$

can be approximated [14] using the harmonic number

$$h_N = \int_0^1 \frac{1-x^N}{1-x} dx, \tag{5}$$

such that

$$E[X] \approx \frac{N \cdot h_N}{k}\,, \tag{6}$$

which is exact for $k=1$. For example, to get a snaphot of the AS topology of a swarm with $N = 20,000$ peers, around $n = 20\,\mathrm{requests}$ have to be sent from each of the 219 used PlanetLab nodes. This takes $n \cdot t = 5\,\mathrm{minutes}$. The computation of the required number of tracker requests allows to dimension the number of monitoring nodes and to adjust appropriately the parameters t and n, if a time frame of 5 minutes is allowed for capturing the snapshot.

However, it has to be noted that Equation (6) only returns the average number of required tracker responses. Checking the percentage of missing IP addresses in our measurements, we observed that only for a small number of swarms some IP addresses are missing. In particular, we checked the percentage of missing IP addresses when observing the AS topology of a swarm. Figure 2 shows the cumulative distribution function (CDF) of the percentage of missing IP addresses when measuring the AS topology for the movies (`Mov.`) and music files (`Mus.`). For 97.5% of all movies (`Mov.`) and more than 98.5% of all music files (`Mus.`), all IP addresses in the swarm were captured. A reason for missing IPs is the fact that peers may go offline during the measurement interval of 5 minutes. This has no effect on the numerical values or on the conclusions with respect to application layer traffic optimization.

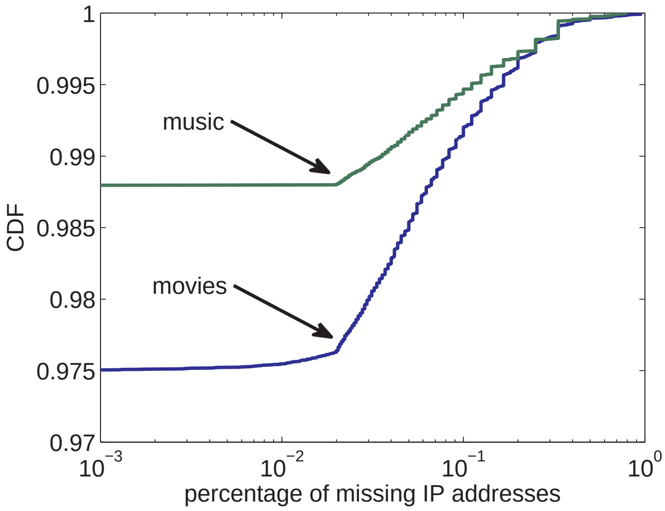

Figure 2: For 97.5% of all movies (Mov.) and more than 98.5% of all music files (Mus.), all IP addresses in the swarm were captured. The percentage of missing IP addresses is given as CDF.

4 Measurement Results

In this section, we describe the results from the measurements and draw some first conclusions from that data. We focus on observations where previous studies provide only a general impression or where the results for specific swarm types contradict the accepted knowledge. In particular, we are interested in the characteristics of the swarm size and its development over time. Additionally, we consider the mapping of swarms on the AS topology of the Internet, since this has important implications for the viability of locality-promoting mechanisms. Another important characteristic to model application layer traffic optimization schemes for BitTorrent has to take into account that within a single AS several swarms are existing in parallel. To this end, we investigate the number of parallel swarms within a single AS. Finally, we report our findings on content that is popular only in specific regions of the world.

4.1 Population Sizes in Swarms

First we take a look at the size of the measured swarms. For this, we analyzed the seeder and leecher population of swarms for different content types, e.g., movies, TV shows and music files, which are registered at different BitTorrent index websites. This allows us to avoid drawing platform- or content-specific conclusions.

Figure 3 shows the observed swarm sizes for the data sets TV., Pop., 24h., Mov., Mus., KPi., KDe. and KMi. We can see that the distribution of the number of peers is similar for all data sets except for the 24h. and Pop. set. An explanation for this divergence is the fact that these two sets feature swarms with specific characteristics due to the popularity of the shared content. While the Pop. set of swarms contains swarms with highly sought after content by definition,

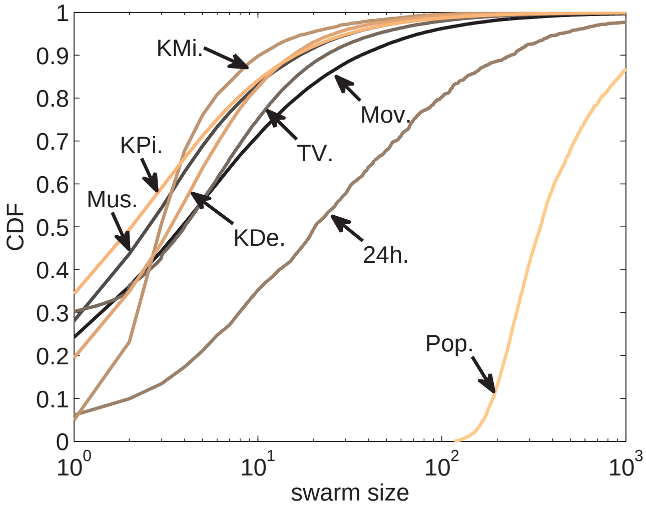

Figure 3: CDF of the number of peers in a swarm.

it is a reasonable assumption that the recently added files of the 24h. set are also more popular than the average, since users are interested in new content which is available for the first time.

To give an impression of the proportion of seeders to leechers in swarms, Figure 4 shows a scatterplot of the number of seeders in relation to the number of leechers per swarm for the TV. data set. This example, which is supported by the other measurement results, implies that the number of leechers is correlated to the number seeders.

The according data for all measured data sets is given in Table 2. It contains the statistics for the total number of observed swarms, their mean value μ and coefficient of variation c_{var} of their sizes, the skewness, kurtosis and maximum of the swarm size distribution as well as the 95th percentile both as an absolute value and normalized by the mean swarm size. Finally, the fraction of swarms π_{80} that contain 80% of the peers and the correlation $C(S, L)$ between the number of seeders and leechers in all swarms of the whole data set is shown.

The first observation we make about these results is that the swarm size depends on the content shared. This is in line with the observations for video file swarms from [10]. The swarms which distribute movies are the largest on average, while smaller music files are shared by less peers on average. This is due to the fact that larger files take longer to download, leading to a longer online time of peers and therefore a higher population in the swarm. This should be offset by the resulting additional upload bandwidth offered to the swarm. However, it can be shown analytically, e.g., by adapting the analysis of [16], that download times do increase in such swarms.

Regarding the different data sets, the coefficient of variation of the swarm size is in the same range, with the exception of the Khirman set of PirateBay swarms (KPi.). This set also differs significantly in terms of skewness, kurtosis and maximum swarm size. Although we cannot judge the source of this discrepancy with our data and the other data sets from Khirman, we still observe that at least the 95th percentile normalized by the mean value is comparable to the

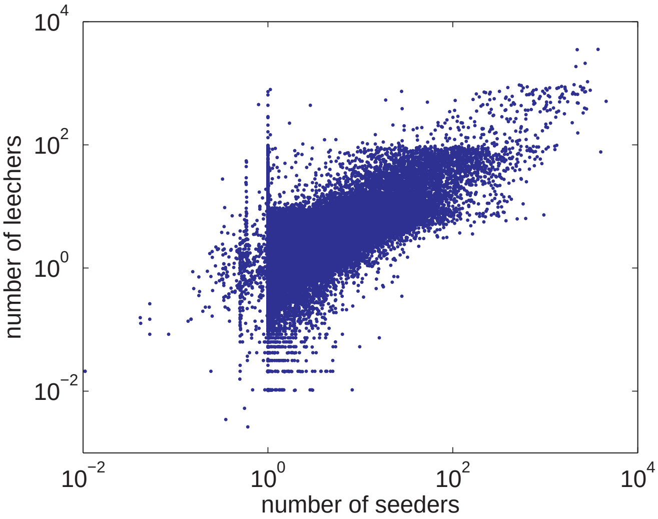

Figure 4: Scatterplot of the number of seeders and the number of leechers in a swarm. Each dot represents (#seeder,#leecher) of an individual swarm for the TV shows dataset TV.

corresponding values for the other data sets. This means that known observations for video file swarms can be extended to other media types as well.

Another general observation is that the Pareto principle holds for most of the evaluated data sets. The p_{80} value, i.e., the fraction of top swarms that contain 80% of all peers in all swarms of the set, is around 0.2 for all sets except the top movies and the Khirman data for the mininova and demonoid sites. This means that 80% of the peers belong to 20% of the swarms, therefore this fraction of the swarms generates roughly 80% of the P2P traffic. It is clear that the most popular content as covered by the Pop. data set do not show this Pareto property, since the different files here are equally popular and represent only a very specific part of the total shared content.

Finally, there is a strong correlation $C(S, L)$ between the number of seeders and the number of leechers in a swarm. This is intuitively clear, since more leechers mean a larger number of potential seeders, and swarms with only a few seeders are normally not popular due to long download times.

From these observations we draw several first conclusions for a locality-aware mechanism. The type of shared content has an impact on the swarm size and therefore on the effectiveness of different locality-promoting solutions. We will see in the next sections that this is also true for the topological characteristics of a swarm, which also depend on the content shared. In general, the swarm size distribution is heterogeneous with a Pareto-like distribution of the total peer population on the different swarms. Also, recently released and popular content leads to much larger swarms in comparison.

Also, there is a significant amount of very small swarms containing less than 40 peers. With typical BitTorrent client parameters, each peer in such a swarm will know all other peers, since it tries to have at least 40 neighbors. The result is a fully meshed swarm. Consequently, accepted solutions using Biased Neighbor Selection (BNS) as introduced in [8], where peers close in the topology are preferred as neighbors, will have a low impact on these swarms, since there is no choice to be made in the neighbor selection.

Therefore, we conclude that it would be a good strategy to concentrate locality-promoting efforts on the comparably few top swarms, including new and popular content. The share of traffic that can be influenced by targeting these swarms is significant (around 80%), while the effort to do so is much lower than when trying to cover all or at least most of the swarms. To optimize the monitoring of swarms in order to find these candidates, it may help to just keep track of the seeder population, since it is strongly correlated to the number of leechers and thus the total population of a swarm.

Table 2: Statistics on the number of peers in a swarm.

ID	# swarms	mean μ	c_{var}	skewness	kurtosis	max.
Mov.	126,049	25.46	8.48	51.89	3,573.01	20,079
TV.	63,867	15.53	6.47	29.45	1,246.99	7,276
Mus.	135,679	9.76	4.24	28.43	1,432.57	3,813
KPi.	1,682,355	11.12	13.42	216.52	69,248.60	72,988
KMi.	4,514	6.99	3.17	19.78	535.82	763
KDe.	11,759	9.73	4.64	22.90	663.79	1,883
Pop.	4,463	691.14	2.08	9.87	144.06	30,691
24h.	1,048	146.68	5.37	17.20	386.37	19,748

Table 2 (cont.): Statistics on the number of peers in a swarm.

ID	q_{95}	q_{95}/μ	π_{80}	$C(S, L)$
Mov.	76	2.98	0.13	0.84
TV.	45	2.88	0.17	0.71
Mus.	32	3.28	0.25	0.61
KPi.	31	2.79	0.18	0.85
KMi.	19	2.72	0.45	0.53
KDe.	27	2.78	0.31	0.65
Pop.	2,068	2.99	0.45	0.73
24h.	435	2.97	0.12	0.65

4.2 Time-Dynamics within a Swarm

While a snapshot of the number and size of swarms is necessary to determine good rules for traffic optimization, it is wrong to assume that a swarm can be treated as static. The population of a swarm varies over time, meaning that the performance of locality-aware mechanisms also depends on this dynamics.

While it may be efficient to promote locality in a swarm that was measured as being large at a given time instant, it may be less efficient when the swarm shrinks quickly after that snapshot.

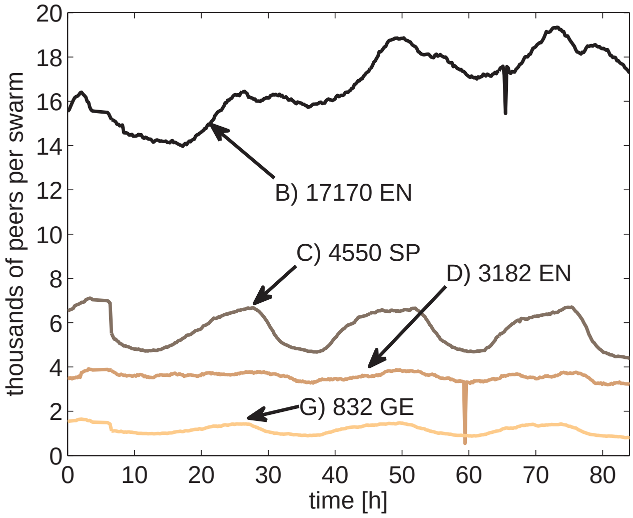

Figure 5: Total swarm size of exemplary swarms (measurement setup `Grp.`) as defined in Table 6.

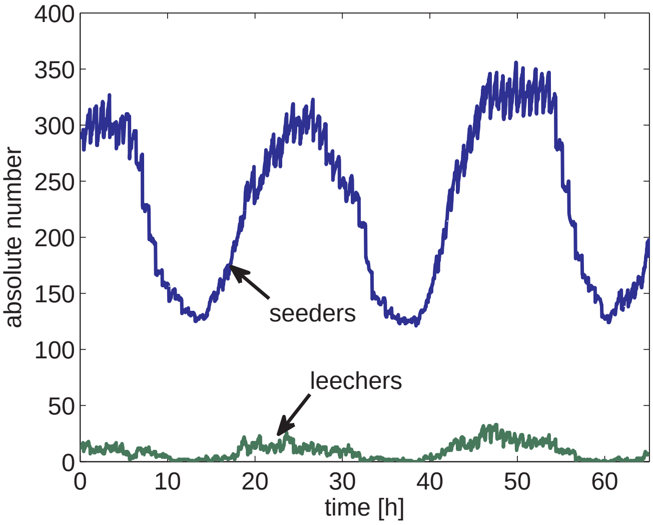

Figure 6: Number of seeders and leechers over time for a German version of open office (`GOO.`).

To gain insights into the time-dependent behavior of swarms, we observed selected swarms over a timespan of several days. The evolution of the size of four of these exemplary swarms, which are taken from the set summarized in Table 6, is depicted in Figure 5. The selection of these swarms allows us to show principal differences between swarms even if they share the same type of content.

We observe that there are variations in the population of each swarm, as well as quantitative and qualitative differences in these variations between the swarms. While swarm D), which is

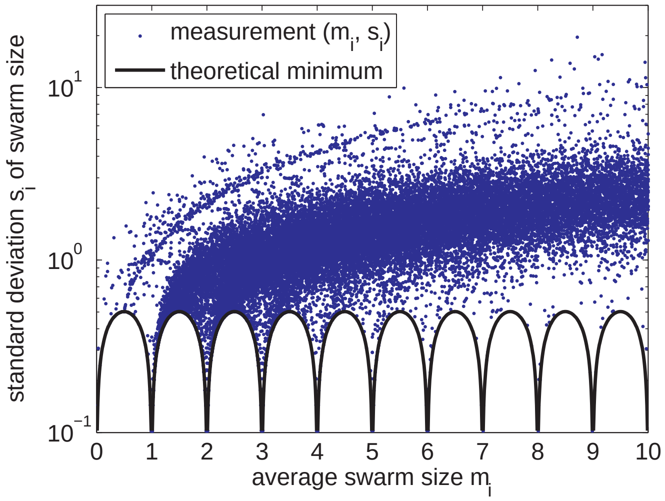

Figure 7: Standard deviation s_i of the size of swarm i vs. average size of swarm i for measurement experiment TV.

sharing a movie in English, shows only small changes in its peer population, the size of swarm C) exhibits a periodic behavior. We attribute this to the fact that in this swarm, a movie in Spanish is distributed. As a consequence, the peers in this swarm can be found mainly in Spain and South America, and therefore the swarm population increases during the daytime in these regions and decreases again afterwards. Swarm G), sharing a German movie, shows a similar characteristic, although it is less pronounced due to the fact that this swarm is smaller.

The development of the peer population of swarm B) is a superposition of a continually increasing popularity and a 24 hour cycle like for swarms C) and G). While swarm D) distributes content that seems not to be preferred regionally, the movie shared in swarm B) seems to be more popular in a specific part of the world. Another example for this dependency of content and a periodic swarm size behavior is shown in Figure 6, where the seeder and leecher population of a swarm distributing a German version of OpenOffice is plotted over time.

In order to be able to describe and compare the dynamics of swarm sizes, we use the standard deviation of the size of a single swarm measured at regular intervals. This value is plotted in Figure 7 for the swarms of the TV. data set, sorted by swarm size. Only swarms with 10 peers or less on average are shown. We see a clear trend for a higher variation of the swarm size in larger swarms. There is a theoretical lower bound for the standard deviation, leading to the peculiar shape of the plot for mean swarm sizes that are not an integer value. Since we capture $R = 96$ samples of the size of a swarm i for the TV. experiment, the minimum standard deviation s_i for a given average swarm size $m_i \in [a; a+1[$ is obtained when we measure k times a size of a and $R-k$ times a size of $a+1$ (for $a \in \mathbb{N}$). Thus, it is

$$m_i = \frac{ka + (R-k)(a+1)}{R} \tag{7}$$

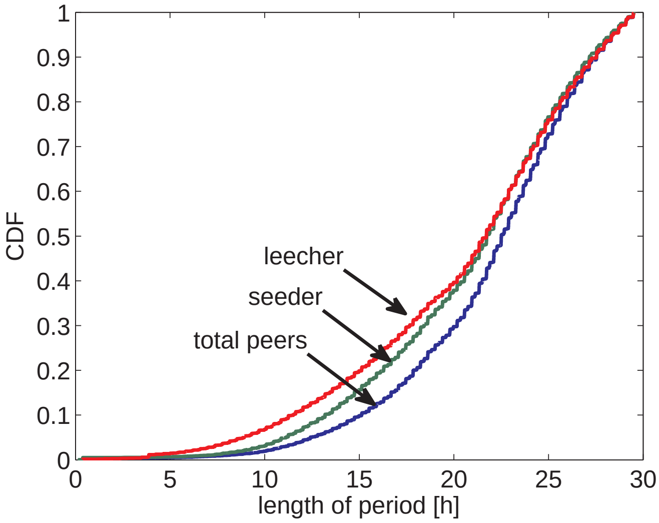

Figure 8: Length of period for TV. by calculating the periodicity transform using the *M-Best Algorithm* [17].

and

$$s_i = \sqrt{\frac{ka^2 + (R-k)(a+1)^2}{R} - m_i^2} = \frac{1}{R}\sqrt{(R-k)k}\,. \tag{8}$$

We now want to determine the amount of swarms that show a diurnal behavior similar to swarms B), C) and G), in order to judge the relevance of this effect for the performance evaluation of locality awareness mechanisms. To that end, we use a method called periodicity transform which automatically detects periodicities for a given data set. In particular, we rely on the 'M-best' algorithm as introduced in [17] that returns a list of the $M = 10$ best periodicities. From the M best periodicities that are $\{\tau_i : 1 \leq i \leq M\}$, we calculate the autocorrelation ρ_i at lag t_i and select the best period of duration τ_k with maximum, positive autocorrelation ρ_k, i.e. $k = \arg\left(\max\{\rho_i : 1 \leq i \leq M\}\right)$.

Figure 8 shows the CDF of the length of the 'best' period for the number of seeders, the number of leechers, and the entire swarm size for the TV. data set. It can be seen that the three different curves show a similar behavior. In particular, the curves for the number of leechers and the total swarm size are almost identical, showing that the leechers mainly determine the diurnal behavior. Furthermore, we observe that roughly the 'best' period for x % of all swarms is around 24 h.

Figure 9 shows the autocorrelation ρ_k to the best period of duration τ_k. Again, the three different curves are quite similar. We observe that from the swarms in the TV. data set only 8.36 % show a strong correlation $\rho_k > 0.7$. As a summary of the time-dynamics analysis, we see for roughly 5.7 % of the swarms a day-night behavior can be observed. To be more precise, for these swarms the autocorrelation is larger than 0.7 for the best period, while the duration of the period is about 1 day, i.e. between 21 hours and 27 hours.

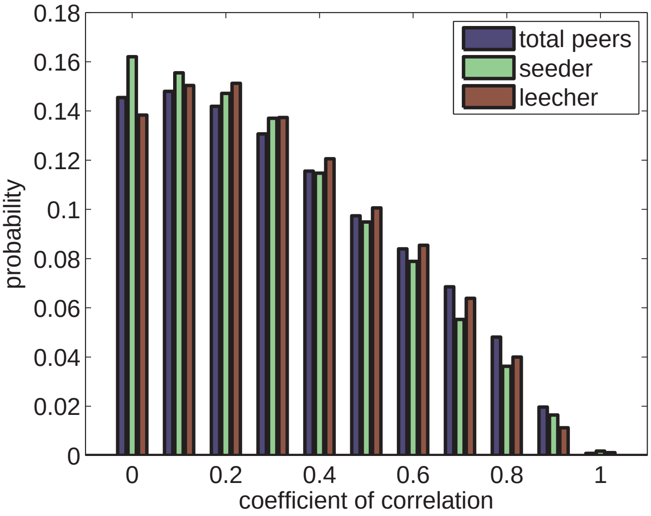

Figure 9: Autocorrelation to the best period for TV.

4.3 AS Topology of Swarms

In order to judge the potential of swarms to be optimized by locality-aware mechanisms, we have to take a look at the distribution of the peers in the Internet topology. We do so by mapping peer IPs to AS's and thus get a statistic on the number of peers per AS for a given swarm. In general, we believe that swarms that are distributed over fewer AS's but with more peers per AS can utilize locality-awareness much better than a swarm that is highly dispersed topologically. While there is already a higher probability for peers to exchange data locally in an AS containing a large share of the swarm, the potential to save traffic by systematically promoting locality is also greater.

We present the CDFs for the average number of peers per AS for swarms of the Mov. data set in Figure 10 and for the Mus. data set in Figure 11, respectively. Note that the x-axis is scaled logarithmically. The swarms are grouped according to their average size as shown in Table 3 and Table 4 together with the relative size of each group. We observe that, for an increasing mean swarm size, the average number of peers per AS grows. However, this value as well as the maximum number of peers in one AS is still small even for the largest swarms.

Table 3: Percentage of swarms grouped according to their size for movie files (Mov.).

[0; 25[	[25; 50[	[50; 100[	[100; 500[	[500; 1e3[	[1e3; ∞[
0.8580	0.0703	0.0294	0.0347	0.0040	0.0036

Table 4: Percentage of swarms grouped according to their size for music files (Mus.).

[0; 25[	[25; 50[	[50; 100[	[100; 200[	[200; 500[	[500; 1000[	[1000; 2000[	[2000; ∞[
93.28	3.80	1.29	0.94	0.52	0.11	0.04	0.01

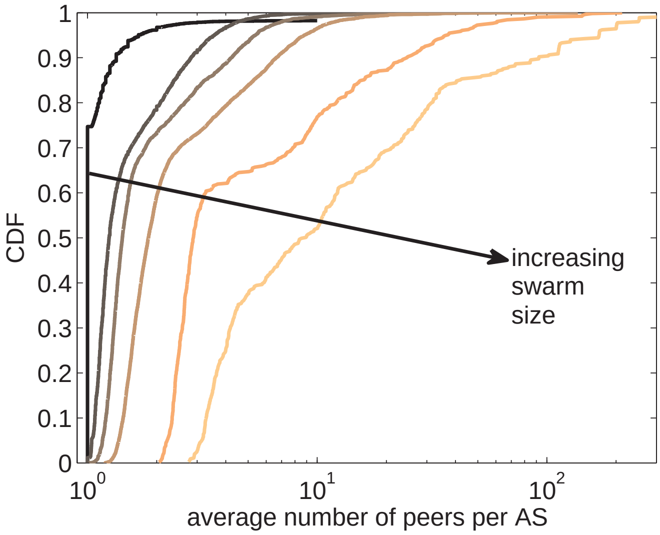

Figure 10: CDF of average number of peers per observed AS. Swarms (`Mov.`) are grouped according to their size, cf. Table 3.

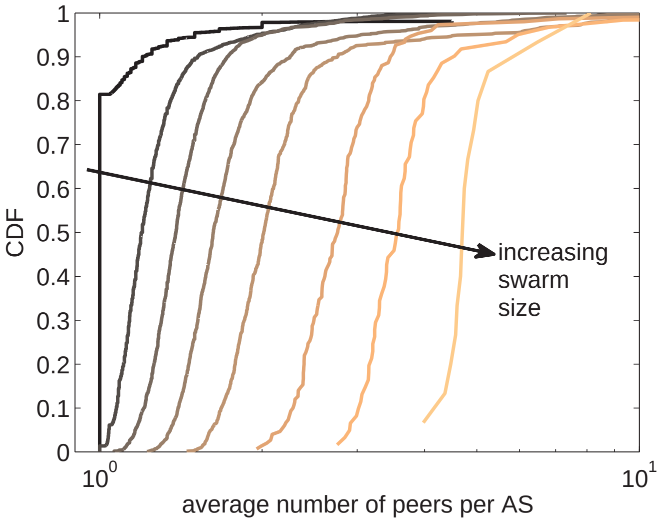

Figure 11: CDF of average number of peers per observed AS. Swarms (`Mus.`) are grouped according to their size, cf. Table 4.

The coefficient of variation also grows for larger swarms (cf. Figure 12, for the same grouping). From the results, we see that there are a only a few, if any, AS's that contain a significant fraction of the swarm, while there are still many AS's in the same swarm holding only one or two peers. For these, locality-awareness is probably only of limited use, since the few peers in these AS's do not have much choice in selecting local peers to exchange data with.

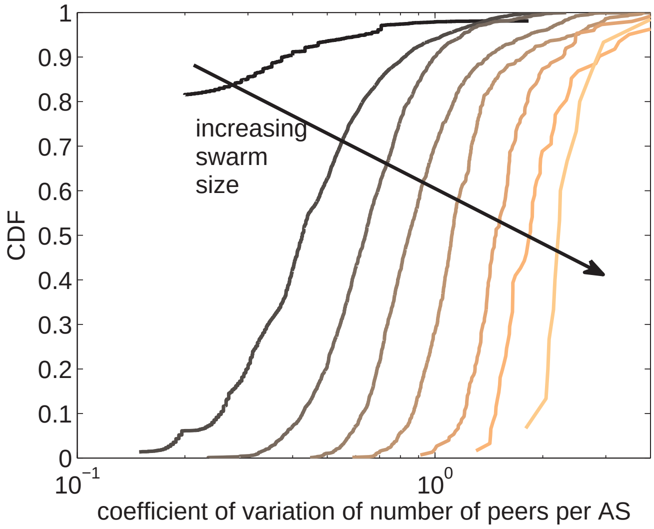

Figure 12: CDF of coefficient of variation of number of peers per observed AS per swarm. The swarms (Mus.) are grouped according to their total size, cf. Table 4.

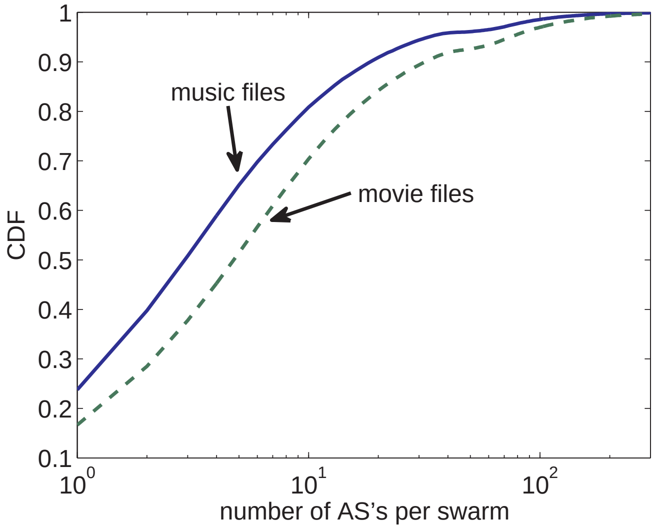

Figure 13: Number of AS's per swarm (Mus.,Mov.).

Another important characteristic of a swarm is the absolute number of AS's, since swarms that are distributed over fewer AS's but with more peers per AS can likely utilize locality promotion mechanism more efficiently. To this end, we consider the movie files (Mov.) as well as the music files (Mus.). Figure 13 shows the CDF of the number of AS's per swarm for both data sets. Since there are more peers involved in swarms offering movie contents, there are also more different AS's involved than in swarms providing music files. On average, there are 65 % more AS's involved in movie swarms than in music swarms. In particular, if the CDF of the number of AS's for movie swarms is normalized by a factor of 1.65, it is nearly identical to the CDF for music swarms. The maximum number of observed AS's is 1,744 for movie swarms and 809 for music swarms, respectively.

4.4 Multiple Swarms in a Single AS

Another important characteristic to model application layer traffic optimization schemes for BitTorrent has to take into account that within a single AS several swarms are existing in parallel. We have taken a closer look at the `Mov.` and `Mus.` data sets which have been captured at the same time. In order to determine the number of parallel swarms, we have parsed the IP addresses of any peer in all swarms of both data sets and mapped them to AS numbers. Since we only consider a subset of all existing types of content and a subset of all existing torrent index websites, the presented study here only gives a lower bound for the number of parallel swarms within an AS.

Figure 14 shows the CDF of the number of parallel swarms within a single AS. It has to be noted that about 10 % of all AS's have only a single swarm. However, the average number of parallel swarms in an AS is about 255. Since the distribution is heavily skewed, the median is only about 12 swarms. The 99% quantile lies at 6,096 parallel swarms, while the maximum number of parallel swarms is 35,327.

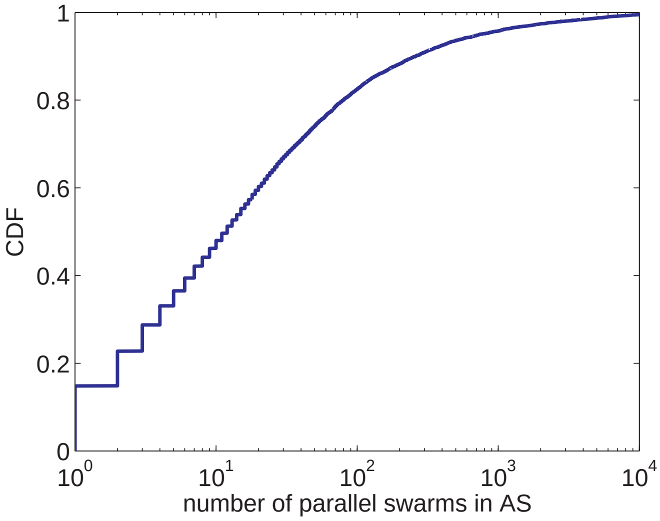

Figure 14: Number of parallel swarms within an AS (`Mus.`,`Mov.`).

The top ten of the AS's with the largest number of parallel swarms is enumerated in the following. We give the observed AS number, the number of swarms which are currently active in the AS, the AS name, and the organization name corresponding to the AS. In order to get this information, we used the ARIN WHOIS database search which is available at `http://ws.arin.net/whois` and the RIPE database at `http://www.db.ripe.net/whois`.

Table 5: Top ten of the AS's with the largest number of parallel swarms. The data sets `Mov.` and `Mus.` are taken into account only.

(1.) AS 7132 (SBIS-AS) participates in 35327 swarms: AT&T Internet Services
(2.) AS 19262 (VZGNI-TRANSIT) participates in 28776 swarms: Verizon Internet Services Inc.
(3.) AS 2856 (BT-UK-AS) participates in 25762 swarms: BTnet UK Regional network
(4.) AS 3269 (ASN-IBSNAZ) participates in 24967 swarms: Telecom Italia
(5.) AS 6327 (SHAW) participates in 24670 swarms: Shaw Communications Inc.
(6.) AS 577 (BACOM) participates in 22447 swarms: Bell Canada
(7.) AS 6830 (UPC) participates in 22244 swarms: UPC Broadband
(8.) AS 812 (ROGERS-CABLE) participates in 22230 swarms: Rogers Cable Communications Inc.
(9.) AS 5089 (NTL) participates in 21975 swarms: NTL Group Limited, United Kingdom
(10.) AS 3352 (TELEFONICA DE ESPANA) participates in 21776 swarms: Telefonica-Data-Espana

4.5 Characteristics of Regional Swarms

We have already seen the effect regional content has on the evolution of the swarm size over time. We now take a closer look at the topological characteristics of swarms sharing this content. To this end, we consider 16 individual swarms of different average sizes distributing movies in German, Spanish, Chinese or English (cf. Table 6). For these swarms, we analyze the absolute number of peers in the AS's observed over the lifetime of the swarm. The results of this analysis are shown in Figure 15. The AS ids on the x-axis are sorted by the number of peers observed in them. This means on the left side we have the AS with the minimum number of peers located in this AS, while on the right side we have the AS's with many peers. On the y-axis we have the absolute number of peers per AS; both axes are scaled logarithmically.

We see that larger swarms tend to have larger shares of the swarm in single AS's. For the largest swarm A, the most AS's are observed. Also, swarms sharing internationally interesting content, i.e., in English, are spread over a larger number of AS's than the swarms distributing more regional content. Thus, swarms C and E, although being larger than swarms J and N, are concentrated on the same or even lower number of AS's due to the fact that the users interested in that content can be found in the same region.

Table 6: Individually measured swarms over time (`Grp.`) using the following notion: *ID) average swarm size & language*

A) 21,351 EN	B) 17,170 EN	C) 4,550 SP	D) 3,182 EN
E) 1,390 SP	F) 972 GE	G) 832 GE	H) 626 GE
I) 579 SP	J) 479 EN	K) 473 GE	L) 351 GE
M) 289 GE	N) 258 EN	O) 217 SP	P) 81 CHI

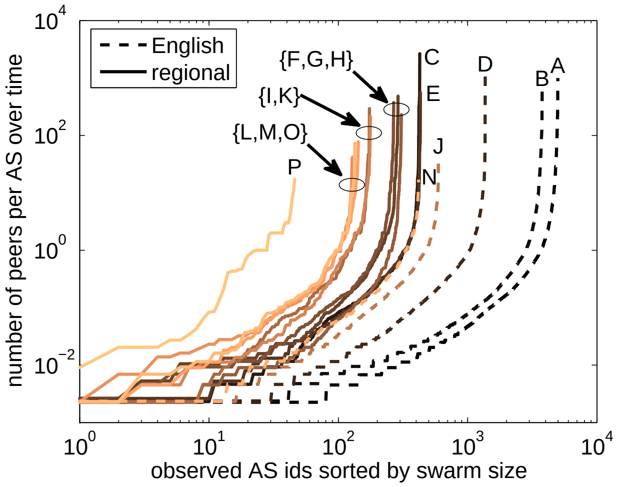

Figure 15: Number of peers per observed AS over time (Table 6).

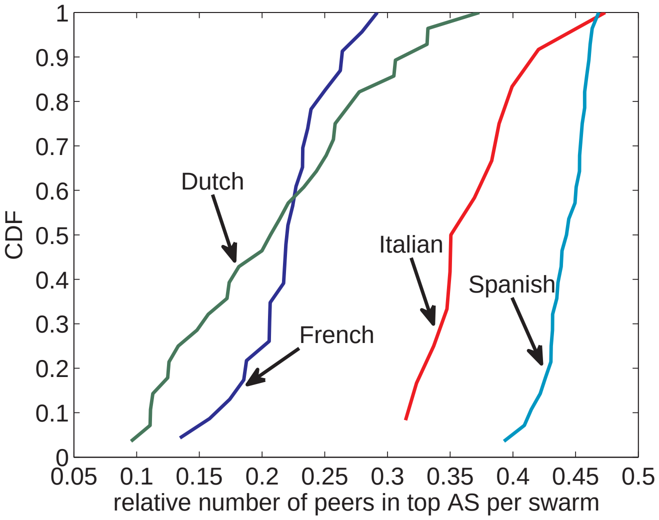

Figure 16: Relative number of peers in a swarm's top AS (`Reg.`).

Swarm D is an exception here. We have seen in Sec. 4.2 that the peer populaton within swarm D remains almost constant over time and doesn't show any periodic day-night pattern. Thus, the swarm distributes content that seems not to be preferred regionally. However, swarm D shows the highest skewness in terms of number of peers per AS compared to the other swarms. In particular, 30 % of the peers belong to the same AS with the AS number 30058. A closer look reveals that the company responsible for this AS offers its customers different ways to host their content, e.g. using dedicated, virtual or colocated servers.

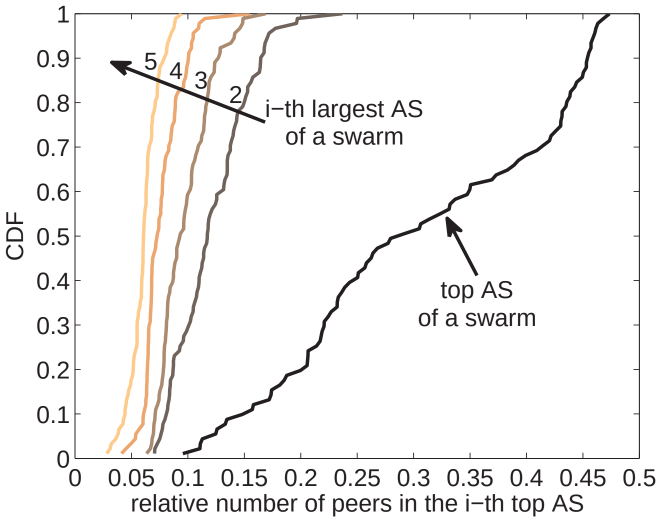

Figure 17: Relative number of peers in i-th top AS (`Reg`.).

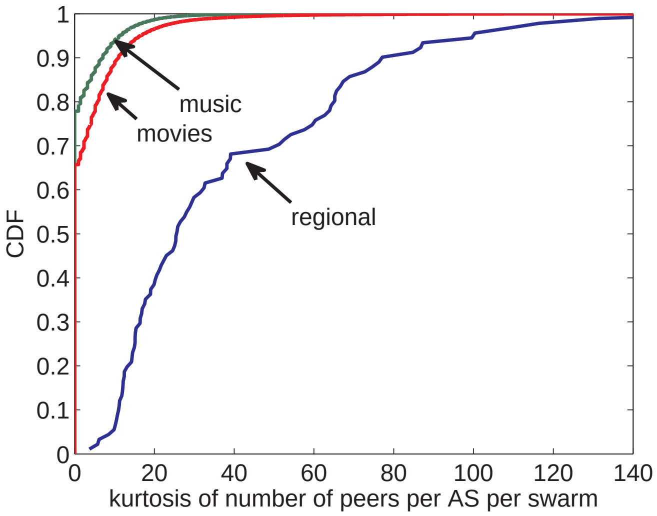

Figure 18: Kurtosis of number of peers per AS (`Reg`., `Mus`., `Mov`.).

The fact that users are interested in regional content leads to a high top AS fraction, which is the relative number peers in a swarm's top AS. This is especially true for Spanish content, see Figure 16. Here, the top AS of each swarm in the `Reg`. set is used for comparison, i.e., the AS containing most peers from a swarm. A CDF of the relative share of peers that are located in these AS's is plotted for swarms with Dutch, French, Italian and Spanish content.

While in all cases there are at least 10% of the total swarm population in the top AS, this share is between 40 and 48% for the Spanish content, implying a high degree of peer grouping. To judge whether this phenomenon only exists for a single AS, we evaluated also the second to

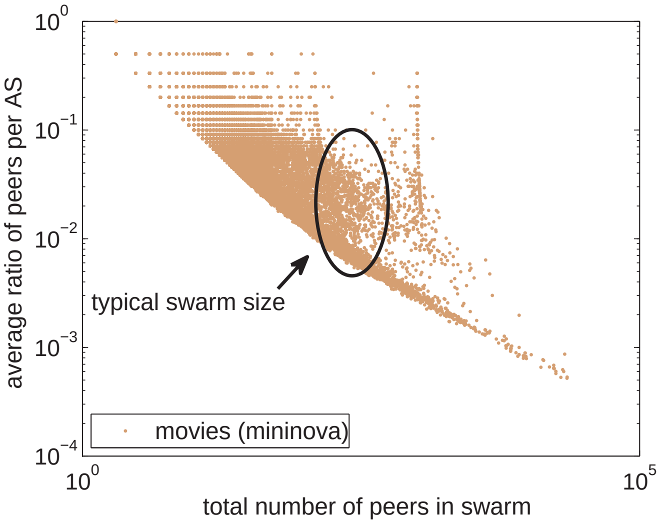

Figure 19: Scatter plot of the total number of peers in a swarm vs the average ratio of peers per AS for the `Mov.` data set.

fifth largest AS's of the swarms in the `Reg.` data set, cf. Figure 17. It appears that the top AS of a swarm contains significantly more peers than the other AS's, although these are still holding around 5% of the total swarm population.

We affirm this result by comparing the kurtosis, i.e., the fourth moment of a distribution that indicates statistical peaks, of the number of peers per AS for the swarms in the `Reg.`, the `Mus.` and `Mov.` sets. The results are shown in form of a CDF in Figure 18.

The regional swarms show a much higher kurtosis than the two larger and more general sets. This leads us to the conclusion that the concentration of a larger fraction of the swarm in the same AS is much more common in regional swarms. Therefore, at least the concentrated parts of these swarms may profit more from locality-aware mechanisms. This means that the regional interest in a shared file can play a significant role in the suitability of the according swarm for locality promotion, something previously underestimated. In particular, the high kurtosis values for a certain fraction of swarms providing music or movie files in Figure 18 indicates that this phenomenon of regional interests with many peers in the top AS can be observed for any kind of content.

We want to highlight here that – independent of the locality mechanism under study – these relevant swarm characteristics have to be considered to show the effectiveness of any algorithm. In [18], we show that a scenario with a heavily skewed peer distribution, Biased Unchoking [19] and Biased Neighbor Selection [8] have opposite effects in the peers in the largest AS. Biased Unchoking can be utilized better in AS's with many peers and therefore lowers the download times. Biased Neighbor Selection lets peers in a large AS mainly compete against each other, while peers from other AS's also have neighbors in the large AS. As a result, the upload capacity of the large AS is used by both local and remote peers, while the local peers do not utilize the rest of the swarm. Consequently, the download times increase.

4.6 Reliable Monitoring of Trackers

Monitoring of BitTorrent trackers can be efficiently utilized in a distributed way to get a snapshot of the swarm. Application layer traffic optimization mechanisms may utilize the monitored data, which comprises the swarm size, the type of content, the language of the offered content, or even the AS topology of the peers currently participating in the swarm. However, the problem remains that in that case the monitored data depends on the actual tracker software.

In our measurements, we found for example one particular swarm (`Ele.`) for which we discovered only 10% of the peers. The tracker returned a swarm size of 400,000 peers, however, we only observed 30,000 IP addresses. We used 219 PlanetLab nodes and requested the tracker every 10 seconds from each machine over 24 hours. Thus, we received more than one million tracker responses with 50 IPs. In that case, we should observe at least around 375,000 different IPs according to Eq. (6). It has to be remarked again, that in Sec. 3.3 we have shown that for 97.5% of all movies (`Mov.`) and more than 98.5% of all music files (`Mus.`), all IP addresses in the swarm were captured.

There are two possible reasons for this observation regarding the swarm `Ele.` (1) The tracker always returns the same IP addresses. This could be the case when locality-awareness mechanisms are implemented by the tracker. However, this is not the case here; the nodes in PlanetLab are distributed world-wide. Thus, it seems reasonable that the random generator or the function which returns a random subset of all peers is wrongly implemented. (2) The tracker returns wrong information about the number of seeders and leechers in the swarm. In both cases, the question arises how an ALTO mechanism can reliably monitor swarms for badly implemented trackers.

5 Modeling AS Topology of BitTorrent Swarms

As we have seen from the measurement results presented in Sec. 4, one key aspect for modelling BitTorrent swarms is the skewed peer distribution. In this section, we present a simple model which returns the probability $P(k)$ that a peer belongs to the k-th largest AS within a swarm consisting of n different AS's. In particular, we investigate whether the peer distribution among the different AS's follows a power-law, which means

$$P(k) = a/k^b + c\,. \tag{9}$$

Therefore, we consider all swarms $\mathfrak{I}_n$ consisting of exactly n different AS's from `Mus.` and the `Mov.` data set, respectively. For each swarm $i \in \mathfrak{I}_n$, we measure the ratio $\widetilde{P}_i(k)$ of peers belonging to the k-th largest AS in swarm i for $k = 1, 2, \cdots, n$. Then, we compute the average ratio $\widetilde{P}(k)$ over all swarms, yielding at

$$\widetilde{P}(k) = \frac{1}{|\mathfrak{I}_n|} \sum_{i \in \mathfrak{I}_n} \widetilde{P}_i(k)\,. \tag{10}$$

Figure 20 shows the measured ratio $\widetilde{P}_i(k)$ of peers belonging to the k-th largest AS within a swarm consisting of $n = 40$ different AS's. All swarms consisting of exactly n different AS's are considered from the `Mus.` data set. The oberved ratio $\widetilde{P}_i(k)$ is then compared with the

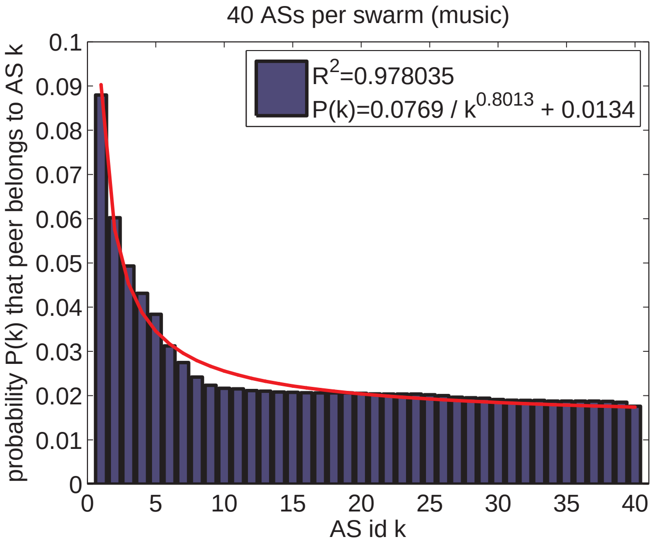

Figure 20: Comparison of the measured ratio $\widetilde{P}(k)$ and the theoretical probability $P(k)$ that a peer belongs to the k-th largest AS within a swarm consisting of $n = 40$ different AS's. All swarms consisting of exactly n different AS's are considered from the `Mus.` data set.

power-law model function as defined in Eq. (9). The parameters a, b, c of the model function are retrieved by means of non-linear regression. We used the optimization toolbox of Matlab to find an optimal fitting function for the given measurement data. Optimal in this case means to find the unknown parameters a, b, c in Eq. (9), such that the mean squarred error is minimized. As a result, we obtain $P(k) = 0.0769/k^{0.8013} + 0.0134$ which is plotted as solid, red curve. Figure 20 indicates that the power-law describes quite well the peer distribution among AS's.

The goodness-of-fit for the model function $P(k)$ is expressed by means of the coefficient of determination R^2. A value close to one means a perfect match between the model function and the measured data. For the measurements given in Figure 20 and the obtained model function, the coefficient of determination is $R^2 = 0.978035$ indicating the good match in a statistical way. In our case, the coefficient of determination can be computed as follows

$$R^2 = 1 - \frac{\sum_{k=1}^{n} \left(\widetilde{P}(k) - P(k) \right)^2}{\sum_{k=1}^{n} \left(\widetilde{P}(k) - 1/n \right)^2} . \tag{11}$$

Analogously, Figure 21 compares the measured ratio $\widetilde{P}_i(k)$ and the fitted model function $P(k) = 0.1445/k^{1.1632} + 0.0128$ for swarms from the `Mov.` data set consisting of $n = 40$ different AS's. Again, the power-law can be observed and the coefficient of determination of $R^2 = 0.993338$ indicates a nearly perfect match between the measurement data and the model function.

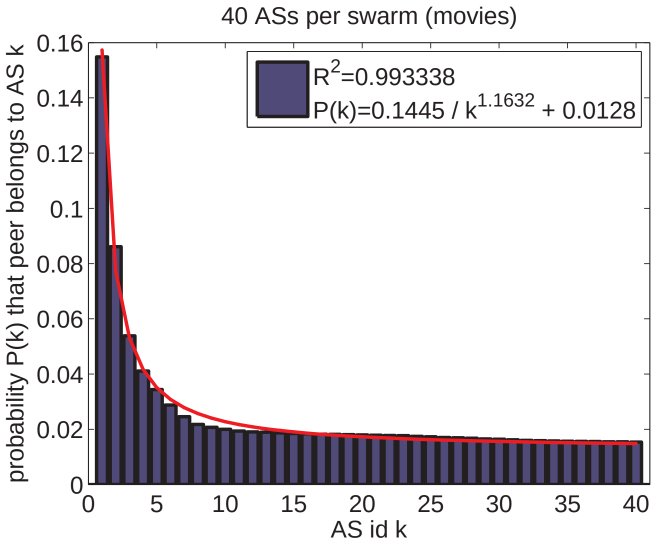

Figure 21: Comparison of the measured ratio $\widetilde{P}(k)$ and the theoretical probability $P(k)$ that a peer belongs to the k-th largest AS within a swarm consisting of $n = 40$ different AS's. All swarms consisting of exactly n different AS's are considered from the `Mov.` data set.

In the following, we have computed the optimal parameters of the power-law function as defined in Eq. (9) for all swarms consisting of exactly n different AS's. Again, the coefficient of determination R^2 is used to measure the goodness-of-fit. Figure 22 and Figure 23 show a scatter plot of the number n of different AS's in a swarm vs. R^2 for the `Mus.` and the `Mov.` data set, respectively. The maximum number of observed AS's is 1,744 for movie swarms and 809 for music swarms. As we can see from both figures, the match between the measurement data and the pwer-law model function is very well and the coefficient of determination is above 0.9.

In order to provide a model for the AS topology of BitTorrent swarms, the number of AS's per swarm is required in addition to the parameters of the power-law model. The number of different AS's was discussed in 4.3 and a further analysis shows that it can be modeled with a log-normal distribution. Using the measurement data, the maximum likelihood estimates of the paramters for the log-normal model distribution were calculated. In particular, we obtain $\mu = 1.2161$ and $\sigma = 1.1009$ for the `Mus.` data set, resulting in a coefficient of determination of $R^2 = 0.99$. For the `Mov.` data set, the parameters of the log-normal distribution are $\mu = 1.5113$ and $\sigma = 1.2636$ leading to $R^2 = 0.99$, again.

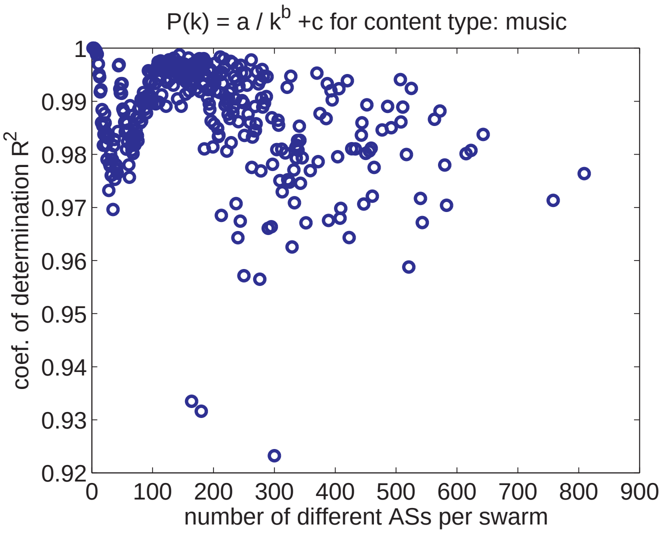

Figure 22: Scatter plot of the number n of different AS's in a swarm vs. the coefficient of determination R^2 as goodness-of-fit measure between the measurement data and the power-law model according to Eq. (11) for the `Mus.` data set.

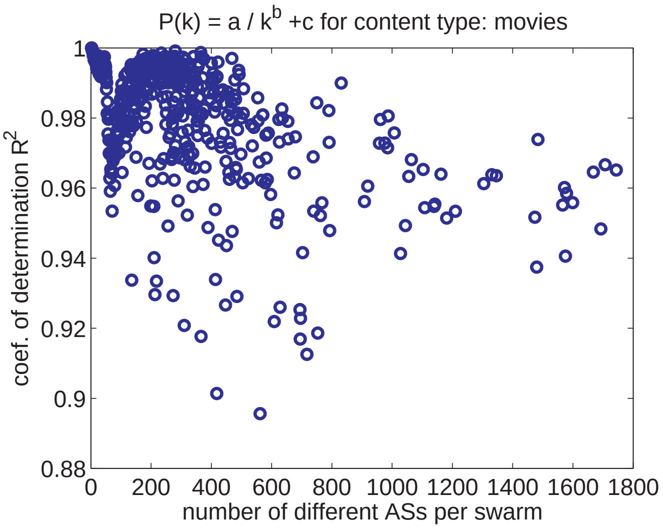

Figure 23: Scatter plot of the number n of different AS's in a swarm vs. the coefficient of determination R^2 as goodness-of-fit measure between the measurement data and the power-law model according to Eq. (11) for the `Mov.` data set.

6 Summary

From the results presented above, we make the following main observations for modeling BitTorrent swarms and its relevance for traffic optimization mechanisms.

Considering the swarm statistics according to the offered content (i.e., TV shows, movies and music) shows that the larger the offered content is in terms of volume, the larger the average and maximum number of peers is in such a swarm, as already shown in less detail in [10]. Additionally, our results show that the distribution of peers among the swarms follows the Pareto principle for the different measurement sets (1), (4) and (5) which contain random files. This means that 80% of all peers belong roughly to the top 20% swarms for all media types. The Pareto principle cannot be observed for measurement set (2), (3), and (6), since we only consider popular or recently published contents there. These recently published torrents are highly popular. This is reasonable, since users are typically interested in new contents, recently broadcasted movies etc. In consequence, traffic optimization should concentrate on the relatively low number of swarms with larger content and high popularity, since the potential gains are much higher than for small swarms. Not only does a larger content lead to more traffic, but also the possibilities for locality promotion are more numerous in larger swarms, where there are more peers in one AS in general.

Also, especially for regional content we observe a day-night behavior of the swarm size, since mostly users of a certain region (within a similar time zone) are interested in that content, e.g., movies in French are mostly downloaded by users from France. In general, we found for 5 % of the investigated swarms a clear statistical indication for day-night behavior. Therefore, traffic optimization schemes need to take into account that their efficiency may vary over time. Also, a one-time observation of a swarm may not suffice to characterize it for its suitability for locality promotion, even if it is no longer in its flash-crowd phase.

Both regional swarms with location-dependent content and large swarms show some top AS's with many peers when considering the AS topology of a swarm, i.e., the AS affiliations of peers within a swarm. However, also the observed number of different AS's of a swarm increases significantly with the swarm size. As a result, more than 90% of the observed AS's contain less than 10 peers in a random swarm. The average number of peers per AS is below 2 peers for 99% of the swarms in some measurement sets. However, the variation of the number of peers per AS can be quite large, e.g., there are many AS's with a single peer, but some AS's with several peers inside.

For modeling the AS topology of BitTorrent swarms, we showed that the number of different AS's within a swarm follows a log-normal distribution. Further, the probability that a peer belongs to the k-th top AS follows a power-law model. Thus, the peer distribution among AS's within a swarm is heavily skewed. This is generally neglected in the evaluation of traffic optimization schemes, where a more even distribution of peers in the internet topology is assumed. The resulting error in judging the effectiveness of a locality promotion mechanism is even more pronounced for swarms containing regional content, where the skewness in the peer distribution is higher. Even if traffic optimization is actively done for only the AS's with large shares of swarms, as proposed in [10], the effect on the whole swarm must be considered. Since these AS's contain significant fractions of the total swarm, applying locality awareness here may very well affect the rest of the peers and the traffic distribution in AS's with a smaller share of peers.

Acknowlegements

The authors would like to thank Benjamin Kleine and Valentin Burger for their help in performing the measurements. This work has been funded partially in the framework of the EU ICT Project SmoothIT (FP7-2007-ICT-216259) and by the Federal Ministry of Educations and Research of the Federal Republic of Germany (Förderkennzeichen 01 BK 0800, G-Lab). The authors alone are responsible for the content of the paper.

References

[1] T. Hoßfeld et al., “An economic traffic management approach to enable the triplewin for users, ISPs, and overlay providers,” in *FIA Prague Book*, 2009.

[2] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, “P4P: Provider Portal for Applications,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 351–362, 2008.

[3] V. Aggarwal, A. Feldmann, and C. Scheideler, “Can isps and p2p users cooperate for improved performance?,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 29–40, 2007.

[4] “IPOQUE. Internet Study 2007: Data about P2P, VoIP, Skype, file hosters like RapidShare and streaming services like YouTube,” 2007.

[5] B. Cohen, “Incentives Build Robustness in BitTorrent,” in *1st Workshop on the Economics of Peer-to-Peer Systems*, (Berkeley, USA), 2003.

[6] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. A. Felber, A. A. Hamra, and L. Garces-Erice, “Dissecting bittorrent: Five months in a torrent’s lifetime,” in *Proceedings of Passive and Active Measurements (PAM)*, (Antibes Juan-les-Pins, France), 2004.

[7] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, “”the bittorrent p2p file-sharing system: Measurements and analysis”,” in *IPTPS’05*, (Ithaca, NY, USA), 2005.

[8] R. Bindal et al., “Improving traffic locality in bittorrent via biased neighbor selection,” in *ICDCS ’06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, (Lisboa, Portugal), 2006.

[9] S. L. Blond, A. Legout, and W. Dabbous, “Pushing BitTorrent Locality to the Limit,” Technical Report inria-00343822, INRIA, Sophia Antipolis, France, 2008.

[10] H. Wang, J. Liu, and X. 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.

[11] PlanetLab, “An open platform for developing, deploying, and accessing planetary-scale services,” 2009.

[12] G-Lab Project, “National platform for future internet studies,” 2009.

[13] S. Khirman, “Torrent as localisation - raw data, `http://www.khirman.com/blog/as_raw_data`,” 2009.

[14] J. Kobza, S. Jacobson, and D. Vaughan, “A survey of the coupon collector’s problem with random sample sizes,” *Methodol. Comput. Appl. Probab.*, vol. 9, no. 4, pp. 573–584, 2007.

[15] A. Binzenhöfer and T. Hoßfeld, “Warum Panini Fußballalben auch Informatikern Spaß machen,” in *Fußball eine Wissenschaft für sich* (H.-G. Weigand, ed.), Verlag Königshausen & Neumann, 2006.

[16] D. Qiu and R. Srikant, “Modeling and performance analysis of bittorrent-like peer-to-peer networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 367–378, 2004.

[17] W. A. Sethares and T. W. Staley, “Periodicity transforms,” *IEEE Transactions on Signal Processing*, Nov 1999.

[18] F. Lehrieder, S. Oechsner, T. Hoßfeld, Z. Despotovic, W. Kellerer, and M. Michel, “Can p2p-users profit from locality-awareness?,” *Currently under submission*, 2009.

[19] S. Oechsner, F. Lehrieder, T. Hoßfeld, F. Metzger, K. Pussep, and D. Staehle, “Pushing the performance of biased neighbor selection through biased unchoking,” in *9th International Conference on Peer-to-Peer Computing*, (Seattle, USA), sep 2009.