# Cache Capacity Allocation to Overlay Swarms

Ioanna Papafili[1], George D. Stamoulis[1],
Frank Lehrieder[2], Benjamin Kleine[2], Simon Oechsner[2],

[1] Athens University of Economics and Business, Athens, Greece
[2] Julius-Maximilian University of Wuerzburg, Wuerzburg, Germany

**Abstract.** Peer-to-peer applications generate huge volumes of Internet traffic, thus leading to higher congestion, as well as higher costs for the ISPs particularly due to inter-domain traffic. The traditional traffic management approaches employed by ISPs (e.g. traffic shaping or even throttling) often lead to a deterioration of users' Quality-of-Experience. Previous works have verified that the insertion of ISP-owned Peers (IoPs) can deal effectively with this situation. The mechanism offers caching while exploiting the self-organizing structure of overlays. Thus, it leads to both improved performance for peers and reduced inter-domain traffic costs for ISPs. In this paper, we study how the available IoP bandwidth capacity should be allocated among the various swarms that it can possibly join. In particular, we identify a variety of factors that influence the effectiveness of Swarm Selection and Bandwidth Allocation, and we investigate their impact on the practically relevant example of BitTorrent, primarily by means of simulations.

**Keywords:** peer-to-peer, ISP-owned Peer (IoP), cost reduction, performance improvement, Swarm Selection, Bandwidth Allocation.

## 1 Introduction

Dissemination of large content files by means of peer-to-peer (P2P) applications is very popular among Internet users for almost 5 years. However, due to their popularity and the size of the files exchanged, P2P applications generate huge traffic volumes. Furthermore, they are highly self-organized and perform their own overlay traffic management, generally without taking into account the underlying network. Thus, it is possible that some information is brought to a peer from a remote one, while being locally available. Due to the resulting increase of inter-domain traffic caused by P2P applications, ISPs pay increased inter-connection charges. Thus, ISPs employ traffic engineering aiming mainly to reduce traffic volumes on their inter-connection links. Traffic optimization performed separately by ISPs and overlays leads the Internet eco-system to a suboptimal situation [1]. Thus, there is need for an incentive compatible mechanism to exploit the nice characteristics of self-organized overlay networks and co-assist ISP's traffic engineering, and lead the system to a *win-win* situation; e.g. leading to both reduced inter-domain traffic charges for ISPs and improved performance for overlay users. This is the objective of Economic Traffic Management (ETM), which is the subject of EU-funded project SmoothIT [2].

An ETM mechanism that appears to achieve a win-win situation is the insertion of ISP-owned Peer(s), proposed and initially analyzed in [3] for the case of a *single* swarm. The ISP-owned Peer (IoP) is an entity equipped with abundant resources, e.g. bandwidth and storage, managed and controlled by the ISP. Note that the IoP runs the overlay protocol with minor modifications, e.g., due to its high amount of resources, it is allowed to unchoke more than the regular number of peers, and thus participates actively in the overlay, storing content and aiming at subsequently seeding this content. The underlying objective is both to increase the level of traffic locality within an ISP, and thus reduce traffic redundancy on its inter-domain links, and to improve performance experienced by the users of peer-to-peer applications (win-win). For the case of BitTorrent [4], which is henceforth assumed, by exploiting the self-organizing incentive mechanism 'tit-for-tat' and due to its high resources, the IoP will be preferred by regular peers to exchange data with. Furthermore, since the IoP is not managed by an actual user but the ISP, and deals with multiple swarms, certain important functions need to be determined. Namely: *i) how the IoP will discover new swarms to join, ii) how it will decide which one of the known swarms to join, and iii) how it will allocate its resources among the swarms it has joined.*

In this paper, we focus on the Swarm Selection and Bandwidth Allocation procedures on the example of the popular and widely-used BitTorrent protocol. We study the impact of these two procedures on the IoP's performance and investigate how different factors affect them. In fact, in practical cases with dynamically varying conditions the IoP would *adapt* its decisions to the changes of these factors in a self-organized way. We use simulative performance evaluation mainly. Thus, we have performed extensive simulations using the multi-swarm capable SmoothIT Simulator v3.0 [5], which is based on the ProtoPeer platform [6]. Note that the impact of cache insertion has been studied already in the literature (see Section 5) but only for single-swarm scenarios. We are not aware of any research that considers a similar multi-swarm approach and the relevant questions arising, as is done in this work. It should be noted that storing content that is illegally shared by users is problem for ISPs. However, this is generally true for any caching approach, of which there are several commercially offered and used in practice, such as products from Oversi [7] or PeerApp [8]. Thus, here we limit ourselves to technical analysis of the IoP approach.

The remainder of the paper is organized as follows: in Section 2, we describe the IoP insertion mechanism. In Section 3, we investigate Swarm Selection and in Section 4, we study the Bandwidth Allocation. In Section 5, we present a brief survey of related work, while in Section 6 we discuss the evaluation results presented in Sections 3 and 4 and conclude our work.

## 2 ISP-owned Peer insertion

Since we study our topics of interest using the practically relevant example of the BitTorrent protocol, we first briefly describe the most important mechanisms of this overlay. BitTorrent is a file-sharing application for the dissemination of large content files to a large number of peers. All peers sharing the same content file form one overlay, which is called a swarm. Peers download a file with meta-data called torrent;

the meta-data includes hash values of the content file and the address of the tracker. The tracker is a centralized entity responsible for neighbor discovering that keeps a complete list of peers that participate in a swarm. Peers request a list of neighbors from the tracker, and the tracker returns a *random* list of peers, both leechers and seeders. Leechers are peers that have only part of the content file, while seeders are peers that have the complete file. Then data chunks exchange among peers begins. The data exchange follows the rules of the BitTorrent protocol; the most significant ones are the choke algorithm and the rarest first replication.

While the insertion of an ISP-owned (and controlled) peer seems to be an indirect way of an ISP to manage its P2P traffic, it is based on the exploitation of the self-organization incentive mechanism of BitTorrent, namely the choke algorithm [4], to increase its effectiveness. The choke algorithm attempts to optimize peers' download rates by employing a variant of 'tit-for-tat'; peers reciprocate to each other by uploading to peers which upload to them with highest rate. Due to its high amount of resources, the IoP is expected to be unchoked with a higher probability by other peers, and therefore is able to concentrate all data chunks more quickly and then help regular peers acquire them. (Note that an ISP can also decide to restrict uploading from the IoP by non-local peers.) This will lead to a reduction of traffic in the inter-domain links, and of the associated charges of the ISP (win), under tariffs such as those based on incoming inter-domain traffic or on the total such traffic, which are often applied in practical cases. However, our target is also to maintain ("no lose" for the overlay) or improve ("win" for the overlay) the performance experienced by the users. Since the aim of the ISP is to affect as much P2P traffic as possible, the IoP should participate in a set of swarms where it can join and offer its available resources more effectively; thus, two important mechanisms have to be implemented *periodically* by the IoP: the ***Swarm Selection*** and the ***Bandwidth Allocation*** mechanisms.

Swarm Selection is the mechanism that selects those swarms that the IoP will join out of the set of all *known* swarms, so that the IoP has a high impact in terms of inter-domain traffic reduction and peers' performance improvement. The detection of new swarms and the discovery of IoP by new peers joining swarms supported by IoP are out of the scope of this paper. Here, we just assume that the IoP is updated on all new and existing swarms by an external entity also managed by the ISP; the same entity is responsible for helping new peers discover IoP; see [9]. Swarm Selection includes a rating of different swarms based on overlay criteria (e.g. content file size, the numbers of leechers and seeders), underlay criteria (e.g. access bandwidth of local peers in the swarm), or both, and finally the selection of a set of swarms with the highest rating.

Bandwidth Allocation is the mechanism responsible for the efficient distribution of the IoP bandwidth (i.e., data rate) among all swarms that the IoP currently joins according to Swarm Selection. The actual data rate that will be used by the peers of a swarm cannot be a priori decided by the IoP; on the contrary, it depends on the number of peers, their download rate, etc. Bandwidth Allocation implies resource dedication to a swarm for the time interval until the bandwidth allocation algorithm re-runs; beyond that bandwidth is provided to peers inside the same swarm based on the overlay protocol. We investigate three different types of Bandwidth Allocation in this work, *uniform* policy, *proportional* policy (according to the ratio of the number of local leechers to the total swarm size) or finally, *max-min* policy (again according to this ratio), which are presented and studied in Section 4.

## 3 Swarm Selection

In this section, we provide an analysis of the Swarm Selection mechanism, investigating the impact of the content file size, the number of local leechers and the number of local seeders, individually and in pairs, on IoP's effectiveness. For simplicity we consider a static case with two swarms evolving simultaneously (see below), and investigate the impact of the aforementioned factors in static IoP Swarm Selection; i.e. in deciding which swarm the IoP should join. Since we cannot a priori determine the dynamically-varying values of the number of leechers and the number of seeders in our simulations, we use instead the *static* parameters of mean inter-arrival times of leechers and the mean seeding time of seeders to prescribe the swarm composition and size in the simulation set-up; the parameters mentioned are those that are *observable on-line* and thus can be the basis for the periodic IoP decision making. Note that here we assume that the IoP unchokes remote peers with no restriction.

In order to verify monotonicity of the leechers' and seeders' number w.r.t. the aforementioned static parameters, we used the theoretical model proposed in [10] that extends the fluid model of [11] to include caches. Thus, we calculate average numbers of leechers and seeders in a swarm w.r.t. the file size, the leechers' arrival rate, or the seeding time in steady-state. For a simple scenario of two ASes and two swarms and using default values for the parameters of the model (as given in [10]), we have derived results for cases of different file sizes, mean inter-arrival times or seeding times. Numerical results (which are omitted, for brevity reasons) show that the average number of leechers is (as intuitively expected) increasing with the file size and decreasing with the mean inter-arrival time and the mean seeding time. These justify our selection of the two aforementioned static factors as criteria for Swarm Selection. Below, we first present the simulation setup for our evaluations and subsequently evaluation results derived from these simulations.

**Simulation setup.** For the evaluation of the Swarm Selection mechanism, we have used the topology of two symmetric ASes, inter-connected via transit inter-AS links to a hub AS, shown in Fig. 1. The tracker and original seeder are connected to the transit hub AS, which has no peers, while the IoP is always inserted in AS1. The topology is very simple but sufficient to serve our purposes; namely to evaluate the impact of the IoP insertion and its mechanisms on inter-AS traffic and peers' download times. Normal peers have an access bandwidth of 2048/128 kB/s down/up (homogeneous), the initial seeder has a capacity of 1280 kB/s (up and down) while the IoP has a capacity of 6400 kB/s (up and down), which is adequately large to reveal the impact of the Swarm Selection procedure.

In the overlay level, we assume two swarms that are specified by the three set-up parameters file size, mean inter-arrival time and mean seeding time. The latter two replace in the configuration of our simulation the number of leechers and the number of seeders, which can be monitored in practice. Peers arrive according to a Poisson process, and after they finish downloading they remain in the swarm and serve as seeders for a time duration that follows the exponential distribution. The tracker and the IoP never leave the swarm. To start the content distribution process properly, we assume the existence of an initial seeder at the beginning of the simulation. This is a peer that acts as the original source of the shared files and leaves the swarm after 1 hour, so that it has no impact on the steady-state of the swarm.

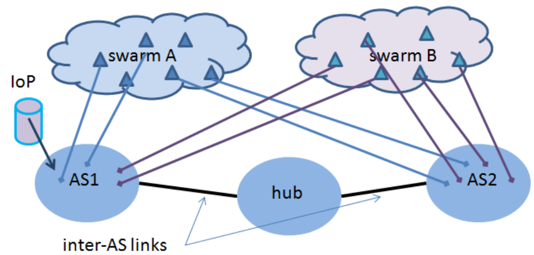

**Fig. 1**. Two-AS topology.

By default, both ASes have some peers that participate only in swarm A, some that participate only in swarm B and some that participate in both swarms. The default file size is 150MB, the leechers' mean inter-arrival time is meanIAT = 100.0s, and the mean seeding time meanST = 600.0s; this results in swarms of about 30 peers concurrently online in steady-state. Such values are common according to recent measurements [12]. To study the impact of the three factors, we tune parameters only for swarm A as reported in Table 1. For swarm B, we always employ these default values. The simulation duration was 3.5 hours, but the results of the first 1.5 hours were ignored as this was considered to be the warm-up phase.

| *Scenario* | *A* | *B* | *C* | *D* | *E* | *F* |
|---|---|---|---|---|---|---|
| ***Modified parameters*** | File Size: 50 MB | meanIAT: 300.0 s | meanST: 200.0 s | File Size: 50 MB, meanIAT: 300.0 s | File Size: 50 MB, meanST: 200.0 s | meanIAT: 300.0 s, meanST: 200.0 s |

**Table 1.** Evaluation scenarios for Swarm Selection.

**Evaluation results for Swarm Selection factors.** In Fig. 2, we present results regarding inter-AS inbound traffic of AS for scenarios A, B and C, where only one of the tunable parameters is modified. (All results hereafter are presented with their respective 95% confidence intervals for 10 simulation runs.) We observe that the impact on inbound inter-AS traffic of AS1 is more significant when the IoP has joined a swarm that either i) serves a content file of larger size, or ii) has lower mean inter-arrival times of leechers and, thus, more leechers (recall the discussion on monotonicity), or iii) has lower mean seeding time and, thus, more leechers and less seeders (again by monotonicity). Thus, IoP insertion has a larger impact on swarms with lower total available upload capacity, or equivalently, higher capacity needs.

Results regarding the average download times are presented in Fig. 3; note that we refer to swarms A and B as sA and sB. For scenario A, it can be seen that peers of both swarms experience large performance improvements when the IoP is inserted in swarm B, which serves the large content file. Although counter-intuitive, peers of the other swarm A experience a large performance improvement too. This is probably due to the capacity surplus from multi-swarm peers. That is, peers of swarm B are now served also by the IoP, resulting in less inter-action with multi-swarm peers; thus a higher capacity of multi-swarm peers is now available to peers of swarm A. For scenarios B and C, we observe that the IoP achieves larger performance improvements for the swarm it joins in all scenarios, although also peers of the other swarm benefit.

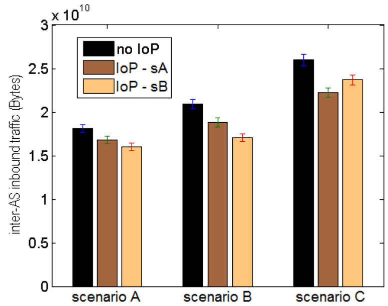


**Fig. 2.** Incoming inter-AS traffic for AS1.

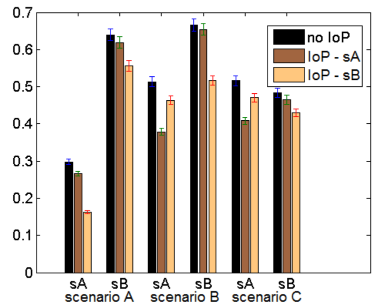


**Fig. 3.** Average download times for AS1.

In Fig. 4, the outgoing and incoming inter-AS traffic for AS1 is depicted for scenarios D, E and F, where two tunable parameters are modified in each scenario. We observe that the reduction of the incoming traffic is larger when the IoP serves either i) the swarm with the larger file and higher arrival rate of leechers, or ii) the swarm with the smaller file but the lower seeding time, or iii) the swarm with the higher arrival rate and larger seeding time. The outgoing traffic behaves similarly in the various scenarios. Generally, we can conclude that the impact of the IoP insertion on inter-AS traffic is more significant when it joins a swarm with a larger number of leechers. Also, the effect of the IoP is highly dependent on the arrival rate of leechers; the seeding time is somewhat less important, and the content file size even less.

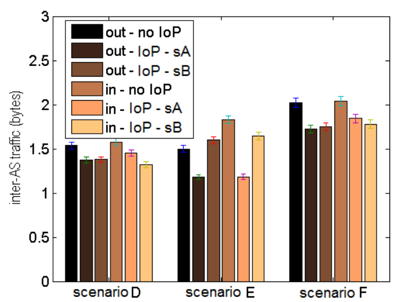


**Fig. 4.** Out- and incoming traffic for AS1.

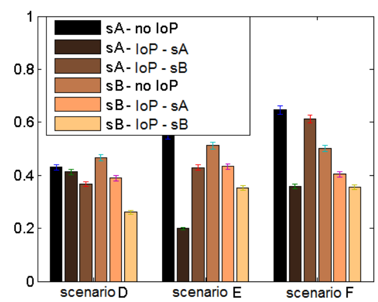


**Fig. 5.** Average download times for AS1.

The average download times for scenarios D, E, and F are shown in Fig. 5. We observe larger performance improvements for the peers of the swarm that the IoP joins in each of these three scenarios; nevertheless, the peers of the other swarm also benefit by the IoP. Note that in scenarios D and E, largest improvement is experienced respectively by peers of: i) the swarm with larger file size and higher arrival rates, or ii) the swarm with smaller file size but also smaller seeding time, when the IoP participates also in them. This happens because the capacity demand is higher in these swarms. Furthermore, scenario F reveals that when the IoP participates in the swarm with the lower seeding time, the percentage of improvement is higher than when it joins the swarm with higher arrival rate. Therefore, we conclude that the effect of IoP on the download times is more affected by the seeding time, somewhat less by the arrival rate and finally even less by the content file size.

## 4 Bandwidth Allocation

In comparison to the last section, we now discuss the possibility to increase the effectiveness of IoPs in scenarios where its bandwidth has to be distributed on multiple swarms. Thus, we assume that the IoP has already committed itself to a number of swarms using a swarm selection strategy based on our previous results. More specifically, we consider scenarios where the IoP has joined two swarms. This allows us to clearly see which of the swarms benefits the most from the allocated cache capacity. Since here it is our primary goal to understand which bandwidth allocation strategy works best, we choose this simple setup instead of employing more realistic scenarios with many swarms. However, we expect our conclusions to apply to such scenarios too. Finally, we assume that the IoP does not unchoke remote peers, in order to maximize its positive effect on the performance of local peers.

For this evaluation, we introduce different Bandwidth Allocation Algorithms (BWAAs), namely *UNIFORM, MIN, MAX, PROP* and *INV-PROP*. The task of the BWAA is to distribute the IoP's available bandwidth in the specific given scenario. In this context, preferable means the highest possible reduction of inter-AS traffic and download time. Here, *UNIFORM* is the default, straightforward strategy (which distributes the available IoP-bandwidth equally among the swarms), and therefore serves as a benchmark strategy for the new approaches to be compared against.

Except for the *UNIFORM*-algorithm, all distribution concepts are based on a parameter R. This parameter represents the ratio of local leechers to external peers. *MIN and MAX* assign all of the available bandwidth to either the swarm with the smallest or the one with the largest R-value. *PROP* and *INV-PROP* on the contrary use this value to give a proportional amount and respectively an inversely proportional amount of the upload capacity to each of the swarms.

Thus, we cover here a range of different policies, where the two most extreme ones, *MIN* and *MAX*, come close to being Swarm Selection policies, since they in effect let the IoP participate only in a single swarm. In a sense, these two strategies therefore introduce a fourth swarm selection criterion to the three parameters discussed previously, namely R. However, due to the fact that they affect the same mechanism as the other strategies in this section, they are compared to these instead of treated separately. In addition, it is not feasible in practice to allocate the full bandwidth of a cache to one swarm, since a saturation effect may set in that wastes a part of this bandwidth. Therefore, we treat the *MIN* and *MAX* policies as theoretical boundaries regarding the Bandwidth Allocation.

**Simulation setup.** Our simulations are again based on a two AS scenario, in which the first AS, AS1 represents the point of view of a single ISP and the second AS, AS2, embodies the rest of the world. The analysis involves two swarms that coexist simultaneously, but do not affect each other (i.e., they have no regulars peers in common), in contrast to the previous experiments. The only connecting instance is embodied by the IoP which enters both swarms. The IoP is equipped with high upload capacity (768 kB/s) which has to be distributed between the two overlays. Compared to the upload capacity of the considered swarms, this capacity is large enough to have an impact on the traffic and the download times of the peers. Still, it does not add too much capacity to the swarm to marginalize the P2P protocol and its effects.

We consider overlay setup similar to the setup described in Section 3 only using meanIAT = 10.0 s. and meanST = 300.0 s; this results in swarms consisting roughly of 120 peers in the steady state. The only difference here is the peer distribution across the two ASs. Again overlay values are common according to recent measurements [12].

In every simulation, our design uses different peer distributions per swarm. Swarm A distributes its peers equally across the two ASs. Swarm B varies the amount which is assigned to the system of the ISP. Therefore, either 5%, 10%, 15% or 30% of the peers in swarm B are located in AS1, which represents the focus of our interest.

In the remainder of this section, we first take one of the possible peer distributions for swarm B and analyze this setup in detail. Subsequently, we take a look at the impact of varying this peer distribution. All results are presented with their according 95% confidence intervals obtained from 10 runs.

**Evaluation results for Bandwidth Allocation strategies.** First, we focus our discussion on the scenario where AS1 contains 50% of the swarm A peers (as always) and 5% of the swarm B peers, before taking a look at the general picture. In this case, the IoP has the opportunity to distribute its available capacity on two swarms, where one has about 60 local peers and the other has only 10% of that size.

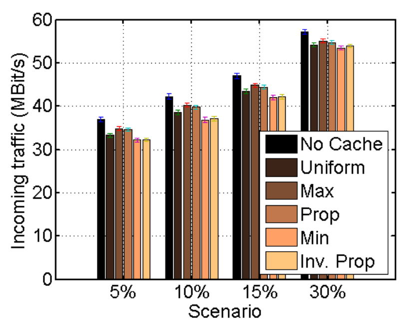


**Fig. 8.** Inbound traffic for AS1.

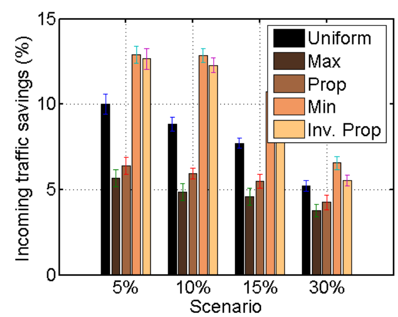


**Fig. 9.** Incoming traffic savings for AS1.

The impact of the BWAAs onto the inter-AS traffic can be seen in Fig. 8. On the x-axis, the four scenarios are displayed, representing the amount of peers that is entering AS1. The graph illustrates the total amount of incoming traffic for AS1.The outgoing traffic is neglected here, as in Section 3, but shows a similar behavior. Fig. 8 also differentiates among the utilized BWAAs. They are emphasized by the use of a color scale where *black* represents the only setup where no IoP was used.

We observe that the introduction of an IoP always reduces the amount of inter-AS traffic regardless of which BWAA was chosen. This effect is not surprising, since we induce additional local upload capacity into the system and particularly in AS1. Among the different algorithms, however, we also note that the amount of inter-AS-traffic is varying. Since the main goal of this section is to optimize the bandwidth-distribution of an existing IoP, we focus our evaluation on the differences between the algorithms. Therefore, we take all results and compare them with the scenario where no IoP is used. In this way, we get an insight into the possible traffic-savings and can easily compare the efficiency of the BWAAs.

This kind of evaluation leads to results as they are illustrated in Fig. 9. The graph shows the results of the incoming inter-AS traffic savings in AS1. The first observation we make is that the efficiency of a IoP is highly dependent on the chosen BWAA. This can be seen in the incoming direction, where the *MAX*-algorithm only saves about 5% of the incoming inter-AS traffic, but this saving can be more than doubled by using the *MIN*- or *INV-PROP*-algorithm. We also note that the savings of the *MAX*-algorithm are even worse than those of a standard uniform distribution. In this (5%) scenario, we achieved the best results by utilizing the *MIN*-algorithm.

This finding implies that an IoP is more efficient in smaller ASs, for the following reasons. If the IoP utilizes the *MAX*-algorithm, it assigns its whole bandwidth to the swarm with the highest ratio of local leechers to external peers. Since 50% of the peers in swarm A are located in the AS 1 against only 5% of swarm B, swarm A will always have a higher number of local leechers, as well as a lower number of external peers. Therefore, it is preferable for the IoP to assign the whole available upload capacity to swarm A. Here, the bandwidth has to be divided between much more peers, than it would have to in swarm B. Subsequently, each one of the peers in swarm A receives only a small share of bandwidth. This leads to a longer download-time and finally to increased inter-AS traffic since the peer spends more time downloading from external sources.

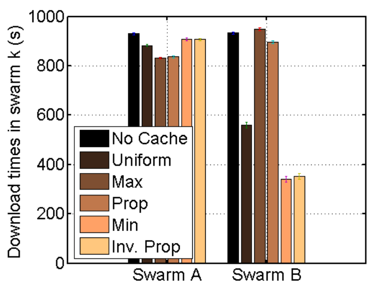


**Fig. 10.** Download times per swarm for the 5% scenario.

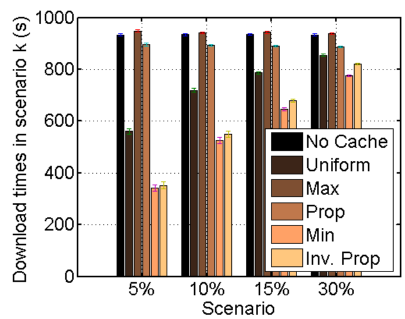


**Fig. 11.** Download times per scenario.

This observation is confirmed as we analyze the results for the peers' download times, which are illustrated in Fig. 10. The graph depicts the mean download times for each of the two swarms. As we expected, the *MAX*-algorithm leads to a reduction of the download time in the swarm A. However, this decrease is small if we compare it to the possible reductions that can be achieved in swarm B. *MAX* is able to reduce the mean download time in swarm A by about 100 seconds. In contrast, we are able to save over half of the download time (500 seconds) in swarm B, if we use the *MIN*-algorithm. The *UNIFORM* distribution is able to reduce the download times in both swarms but again, we observe that, due to the lower number of peers, the bandwidth is used more efficiently in swarm B. However, it should be noted that in any case, the number of peers experiencing the shorter download times is less in swarm B, since there are less leechers of this swarm in AS1. Combining the results for the traffic performance and the download times, we conclude that in this (5%) scenario, the IoP achieves the highest effectiveness if it uses the *MIN*- or the *INV-PROP*-algorithm.

Next, we take a look at the whole range of our considered scenarios, comparing the results for multiple configurations. As we can see in Fig. 8 and 9 changing the peer distribution of swarm B on the one hand has a large impact on the total inter-AS traffic. On the other hand it also affects the efficiency of each of the BWAAs.

The first observation can be made by analyzing the development of the total traffic exchange over the different scenarios, which is illustrated in Fig. 8. We observe that the total amount of incoming traffic is increasing, as the proportion of peers in the AS1 rises from 5% to 30%. This finding can be traced back to the fact that we allocate a larger portion of swarm B into AS1. This variation leads to more peers in AS1 that have the possibility to up- and download data externally. Therefore, the total amount of inter-AS traffic is rising.

However, the four peer distribution scenarios also affect the individual BWAA efficiency, yet showing the same effect we could already observe for the 5% scenario. Fig. 9 illustrates inter-AS traffic savings compared to the respective scenario, where no IoP is used. In this graph, we observe that inter-AS savings are dropping as the amount of peers in AS1 increases. This finding confirms the assumption of a decrease of the IoP's efficiency in larger ASs. As we compare the values of the first and last scenarios, we note that the savings of the *MIN*-algorithm, which prefers swarm B, are almost cut in half if the amount of local peers in this swarm is increased to 30%.

We also denote a decreased saving regarding the algorithms that prefer swarm A (*MAX* and *PROP*). This dropping appears surprisingly, as we do not modify the swarm affected by these algorithms. However, this observation can be explained by the raised amount of total inter-AS traffic. In every scenario, the algorithms save the same absolute amount of exchanged traffic. Since, however, the total traffic is now increased, the proportional saving is lowered. Thus, for all scenarios, our evaluation shows that the highest reduction of inter-AS traffic can be achieved by the *MIN*-algorithm although the possible savings decrease, as the numbers of local peers in the two swarms converge.

If we now take a look at the download times for all setups, the statistics support our previous conclusions for the 5%-scenario. Fig. 11 shows the download time statistics for the four different scenarios. As we raise the proportion of peers located in AS1, we observe increasing values for those algorithms, which mainly affect swarm B. In the 5%-scenario, the *MIN*-algorithm was able to significantly reduce the download time by about 700 sec, whereas in the 30%-scenario, this reduction is lowered to about 200 sec. If we compare the algorithms among each other, we can conclude that again, the *MIN*-algorithm delivers the best performance. The *INV-PROP*-algorithm.is also a good choice, and may be preferable if more than two swarms are in place.

## 5 Related work

The problem of suboptimal behavior for both ISPs and overlay due to randomized overlay decisions and classic traffic engineering, respectively, has been addressed in other research works in the past. There is extensive literature of approaches that study similar scenarios and have same objectives, namely the reduction of inter-connection costs, in terms of inter-domain traffic, and users' performance improvement. In this section, we discuss some of the most representative ones.

In [13], a centralized entity called the Oracle is introduced. The Oracle provides a service to help peers choose local neighbours. When a peer has several possible sources to download content from, it queries the Oracle, sending a list of IP addresses. The Oracle ranks the received addresses according to ISP metrics, taking locality information into account, and returns the list back to the application.

Another centralized approach (namely, P4P) is proposed in [14]. P4P stands for "Proactive network Provider Participation for Applications". P4P's main objective is to allow cooperation between ISPs and overlay providers in order to boost content distribution and optimize utilization of ISP network resources. The P4P framework consists of an ISP-controlled entity, called iTracker, which provides locality information to overlay application trackers (appTrackers). The information is used by the appTrackers to perform a biased initial peer selection before replying to a request.

An approach of different flavor, called Ono has been proposed in [15]. Ono is a fully distributed solution that exploits existing network information to boost self-organising of the overlay network. It has been implemented as a plug-in for the Azureus BitTorrent client and works by executing periodical DNS lookups to CDN names in order to build and maintain proximity maps. A peer compares its proximity maps to other peer's and calculates the overlap. The Ono approach is purely overlay-based, i.e. no cooperation from the ISP or CDN is required.

The aforementioned non-transparent approaches share similar key objectives to our work. However, regarding their function, they propose either alternative neighbour selection criteria, based on some proximity metric provided either by the ISP, as in [13] and [14], or by a public infrastructure such as CDNs [15]. The IoP, on the other hand, does not make use of any network information; it transparently participates in the overlay, offers "for free" additional resources to other peers, and exploits a native self-organizing incentive mechanism.

A solution that is more closely related to the IoP is proposed in [16]. An ISP offers additional *free* resources to selected users that act in a most underlay-friendly way and thus become able to bias localization of the overlay traffic. The purpose of this action is twofold. First, the operator chooses those peers who upload mostly to local peers, thus, increasing the amount of bandwidth available to local peers. Hence, less data is to be downloaded from distant domains. Moreover, this approach offers an incentive for the peers to get in line with the ISP and behave socially and locality-aware. This solution has several similarities to the IoP, such as the extra capacity made available by the ISP to the network and the fact that it also leads to win-win, since it improves the users' performance too.

## 6 Conclusions

The insertion of ISP-owned Peers was previously known to be a promising solution for the ISP to avoid the increasing inter-domain traffic charges due to sub-optimally routed overlay traffic and the performance deterioration of peer-to-peer users due to other ISP traffic management practices. In this paper, we investigated by means of simulations the impact of different overlay factors that may affect IoP's effectiveness, and studied policies for the efficient allocation of the IoP's abundant resources to the swarms that it joins. Our main findings are that the inter-arrival rate of peers and their seeding times are more influential inputs of a Swarm Selection policy, w.r.t. the inter-domain traffic and the download time of the peers, than the size of the shared file. As for the Bandwidth Allocation policy employed once the IoP has joined a set of swarms, our results suggest that it is useful to utilize more of the IoP's upload capacity in swarms with a low share of local leechers, since this leads to higher savings of inter-domain traffic. Future work should focus on combining all IoP influential factors in a single index, which is better suited to the case of many swarms.

**Acknowledgments.** This work has been accomplished in the framework of the EU ICT Project SmoothIT (FP7-2007-ICT-216259). The authors would like to thank all SmoothIT partners for useful discussions on the subject of the paper, as well as the anonymous referees for their constructive comments.

## References

1. Y. Liu, H. Zhang, W. Gong, D. Towsley, "On the Interaction Between Overlay and Underlay Routing," Proc. IEEE INFOCOM 2005
2. The SmoothIT Project: http://www.smoothit.org
3. I. Papafili, S. Soursos, G. D. Stamoulis, 'Improvement of BitTorrent Performance and Inter-domain Traffic by Inserting ISP-owned Peers,' 6th International Workshop on Internet Charging and QoS Technologies ICQT'09, Aachen, Germany, May 2009
4. B. Cohen, Incentives build robustness in BitTorrent, In Proceedings of the First Workshop on the Economics of Peer-to-Peer Systems, Berkeley, CA, USA, June 2003
5. SmoothITSimulator v3.0, http://protopeer.epfl.ch/wiki/BitTorrent
6. ProtoPeer, http://protopeer.epfl.ch/index.html
7. Oversi Networks Ltd. Company website, http://www.oversi.com/
8. PeerApp Company website, http://www.peerapp.com/Default.aspx
9. The SmoothIT project: Deliverable 2.3 – ETM Models and Components and Theoretical Foundations (Final), October 2009
10. F. Lehrieder, G. Dán, T. Hoßfeld, S. Oechsner, V. Singeorzan, 'The Impact of Caching on BitTorrent-like Peer-to-peer Systems,' IEEE International Conference on Peer-to-Peer Computing P2P 2010, Delft, the Netherlands, August 2010
11. D. Qiu, R. Srikant, "Modeling and Performance Analysis of BitTorrent-like peer-to-peer networks", Proc. ACM SIGCOMM Conference on Applications, 2004
12. T. Hoßfeld, D. Hock, S. Oechsner, F. Lehrieder, Z. Despotovic, W. Kellerer, M. Michel, "Measurement of BitTorrent Swarms and their AS Topologies," Computer Networks, in press
13. V. Aggarwal, A. Feldmann, C. Scheideler: Can ISPs and P2P users cooperate for improved performance?, SIGCOMM Computer Communication Revue, Vol. 37, No. 3, 2007
14. H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, A. Silberschatz: P4P: Provider Portal for Applications, ACM SIGCOMM, Seattle, WA, USA, August 2008
15. D. Choffnes, F. Bustamante: Taming the Torrent: A practical approach to reducing cross-ISP traffic in P2P systems, ACM SIGCOMM, Seattle, WA, USA, August 2008
16. K. Pussep, S. Kuleshov, C. Gross, S. Soursos, "An Incentive-based Approach to Traffic Management for Peer-to-Peer Overlays," 3rd Workshop on Economic Traffic Management (ETM 2010), September 2010, Amsterdam, The Netherlands