Cache Capacity Allocation for BitTorrent-like Systems to Minimize Inter-ISP Traffic

Valentino Pacifici*, Frank Lehrieder†, György Dán*
* School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
†University of Würzburg, Institute of Computer Science, Würzburg, Germany

***Abstract*—Many Internet service providers (ISPs) have deployed peer-to-peer (P2P) caches in their networks in order to decrease costly inter-ISP traffic. A P2P cache stores parts of the most popular contents locally, and if possible serves the requests of local peers to decrease the inter-ISP traffic. Traditionally, P2P cache resource management focuses on managing the storage resource of the cache so as to maximize the inter-ISP traffic savings. In this paper we show that when there are many overlays competing for the upload bandwidth of a P2P cache then in order to maximize the inter-ISP traffic savings the cache's upload bandwidth should be actively allocated among the overlays. We formulate the problem of P2P cache bandwidth allocation as a Markov decision process, and describe two approximations to the optimal cache bandwidth allocation policy. Based on the insights obtained from the approximate policies we propose SRP, a priority-based allocation policy for BitTorrent-like P2P systems. We use extensive simulations to evaluate the performance of the proposed policies, and show that cache bandwidth allocation can improve the inter-ISP traffic savings by up to 30 to 60 percent. We validate the results via BitTorrent experiments on Planet-lab.**

I. Introduction

The number of peer-to-peer applications has increased significantly in recent years, and so has the amount of Internet traffic generated by peer-to-peer (P2P) applications. P2P traffic accounts for up to 70% of the total network traffic, depending on geographical location [1], and is a significant source of inter-ISP traffic. Inter-ISP traffic can be a source of revenue for tier-1 ISPs, but it is a source of transit traffic costs for ISPs at the lower levels of the ISP hierarchy, e.g., for tier-2 and tier-3 ISPs. Some ISPs have attempted to limit their costs due to P2P applications by throttling P2P traffic [2]. Nevertheless, the users of P2P applications constitute a significant share of the ISPs' customer base, and hence a solution that negatively affects the performance of P2P applications can result in a decrease of an ISP's revenues on the long term.

Recent research efforts have tried to decrease the amount of inter-ISP P2P traffic by introducing locality-awareness in the neighbor-selection policies of popular P2P applications, like BitTorrent [3]–[6]. Locality information can be provided by the ISPs [3]–[5] or can be obtained via measurements [6], and is used to prioritize nearby peers to distant ones when exchanging data. Through exchanging data primarily with nearby peers a P2P application can improve the locality of its traffic, and hence, can decrease inter-ISP traffic. Nevertheless, locality-aware neighbor selection can deteriorate the performance and the robustness of a P2P application [7].

To address the problem of increased inter-ISP traffic, many ISPs have deployed P2P caches [8], [9]. P2P caches, similar to web proxy caches, decrease the amount of inter-ISP traffic by storing the most popular contents in the ISP's own network, so that they do not have to be downloaded from peers in other ISPs' networks. According to measurement studies 30 to 80 percent of P2P traffic is cacheable [10], [11]. Nevertheless, the actual efficiency of a cache depends on two main factors. First, the amount of storage, which determines the share of the contents that can be kept in cache. Second, the available bandwidth of the cache, which determines the rate at which data can be served by the cache, if the data are in storage.

The goal of cache storage management is to maximize the probability that data are found in the cache when requested. The algorithms for cache storage management, called cache eviction policies, in the case of P2P caches differ significantly from those in the case of web proxy caching. Web objects are typically small, and consequently eviction policies can replace entire contents at once [12]. Objects in P2P systems are nevertheless typically too big to be replaced at once, so that eviction policies for P2P caches have to allow partial caching of contents [10], [11]. By allowing partial caching, P2P eviction policies can achieve within 10 to 20 percent of the optimal offline eviction policy [10], [11].

The impact of the cache bandwidth and its management has received little attention, even though cache bandwidth can be costly, as caches are often priced based on their bandwidth [8], [9]. In the case of web proxy caching bandwidth management is not necessary, because the incoming inter-ISP traffic saving equals the amount of data served from the cache. In the case of a P2P cache the inter-ISP traffic saving is, however, not only determined by how much data the cache serves but also by the characteristics of the overlay to which the data is served [13].

The fundamental question we address in this paper is whether given a limited amount of P2P cache bandwidth, the bandwidth can be actively managed such as to minimize the amount of inter-ISP traffic. We make three important contributions to answer this question. First, we provide a mathematical formulation of the cache bandwidth allocation problem, and show the existence of a stationary optimal policy. Second, we propose allocation policies to approximate the optimal policy and based on the insights gained from these policies we propose a simple priority-based policy for cache

bandwidth allocation. Third, through simulations and through experiments on Planet-lab we show that by actively allocating the upload bandwidth between different overlays the inter-ISP traffic savings due to P2P caches can be improved significantly.

The rest of the paper is organized as follows. In Section II we review the related work. In Section III we describe the system model and formulate the problem of cache bandwidth allocation. In Section IV we show the existence of an optimal cache bandwidth allocation policy, and describe policies to approximate the optimal policy. In Section V we show simulation results to quantify the potential of the proposed bandwidth allocation policies, and validate the simulations via experiments. Section VI concludes the paper.

II. Related work

The solutions for ISP-friendly P2P application design proposed in the literature fall into three main categories: peer-driven, ISP-driven and caching [14]. Peer-driven solutions adapt the neighbor selection strategy of the peers by relying on measurements of latency [15], on AS topology map information [5] or on third-party infrastructures like CDNs [6]. Motivated by the difficulty of inferring the ISPs' interests based on measurements [3], [4] investigated the use of ISP-provided information to influence peer selection. All these works make P2P systems more ISP-friendly by influencing the overlay construction, and are complementary to P2P caching.

Caching of P2P contents has been the subject of several works. Most works focused on the achievable cache hit ratios [16], [17], and on the efficiency of various cache eviction policies [10], [11]. Our work is orthogonal to the works on cache eviction policies, as we assume the existence of a cache eviction policy, and we consider the impact of allocating the cache's upload bandwidth between competing overlays on the amount of inter-ISP traffic generated by the overlays.

Cache upload bandwidth management for P2P video streaming systems was considered in [18], [19] in order to decrease the ISPs' incoming transit traffic. In the case of streaming the download rate of peers is determined by the video rate, and the received rate does not influence the peers' behavior. This makes the problem of cache bandwidth allocation for streaming systems significantly different from the problem considered in this paper. We do not only consider the impact of the cache upload rate on the instantaneous inter-ISP traffic, but also its impact on the system dynamic.

Closest to our work is [20] where the authors studied the impact of different bandwidth reservation schemes between two overlays via simulations. They concluded that the impact of cache bandwidth allocation was minor, which can be attributed to the inefficiency of the cache bandwidth utilization under the considered schemes. Compared to [20] in this paper we give a mathematical formulation of the problem of cache bandwidth allocation, use analytical models of the swarm dynamics and the inter-ISP traffic to give insight into the characteristics of an optimal allocation policy, and use simulations and experiments to demonstrate the inter-ISP traffic savings achievable through cache bandwidth allocation.

Our work relies on the analytical models of the system dynamics of BitTorrent-like systems in [13], [21]–[25]. These works used a Markovian model of the system dynamics of BitTorrent-like systems to model the service capacity and the scalability [21], [22], to evaluate the impact of peer upload rate allocation between two classes of peers [23], to assist the dimensioning of server assisted hybrid P2P content distribution [25], and to evaluate the impact of caches on the swarm dynamics and on the amount of inter-ISP traffic for a single overlay [13]. Our work differs significantly from these works, as we consider multiple overlays and use the fluid model of the system dynamics to get insight into the characteristics of an optimal P2P cache bandwidth allocation policy.

III. System Model and Problem Formulation

In the following we describe our model of a multi-swarm file-sharing system, and then we formulate the problem of cache bandwidth allocation between different swarms.

A. System Model

We consider a set $\mathcal{I} = \{1, \ldots, I\}$ of ISPs, and a set of swarms $\mathcal{S} = \{1, \ldots, S\}$, whose peers are spread over the ISPs. Peers are either leechers, which download and upload simultaneously, or seeds, which upload only. Leechers arrive to swarm s according to a Poisson process with intensity λ_s, the arrival rate of leechers in ISP i is $\lambda_{i,s}$. The Poisson process can be a reasonable approximation of the arrival process over short periods of time [26], even if the arrival rate of peers varies over the lifetime of a swarm. We model the leechers' impatience by the abort rate θ. A leecher departs at this rate before downloading the entire content. Seeds depart from the swarm at rate γ, so that a seed stays on average $1/\gamma$ time in the swarm. The upload rate of peers is denoted by μ and their download rate by c. We focus on the case when $\mu < c$. For simplicity we consider that all files have the same size, and thus, μ and c can be normalized by the file size. Finally, we assume that leechers can use a share η of their upload rate due to partial content availability. This model of swarm dynamics was used in [13], [21], [22], [24], [25].

We denote by $X_{i,s}(t)$ the number of leechers in ISP i in swarm s at time t, and by $Y_{i,s}(t)$ the number of seeds in ISP i in swarm s at time t. $X_{i,s}(t)$ and $Y_{i,s}(t)$ take values in the countably infinite state space $\mathbb{N}_0$. As a shorthand we introduce $Z_{i,s}(t) = (X_{i,s}(t), Y_{i,s}(t))$ and $Z_s(t) = (Z_{i,s}(t))_{i\in\mathcal{I}}$. Finally, we denote the state of the swarms by $Z(t) = (Z_s(t))_{s\in\mathcal{S}}$.

Seeds and leechers in ISP i can upload and download data to and from peers in any ISP $j \in \mathcal{I}$. We define the publicly available upload rate $u^P_{i,s}(t)$ as the available upload rate located in ISP i that can be used by leechers of swarm s in any ISP. This quantity tantamounts the upload rate of the leechers and the seeds $u^P_{i,s}(t) = \mu(\eta X_{i,s}(t) + Y_{i,s}(t))$. A leecher cannot download from itself, therefore the publicly available upload rate in ISP i to a local leecher of swarm s is $u^{PL}_{i,s}(t) = \max[0, \mu(\eta(X_{i,s}(t) - 1) + Y_{i,s}(t)]$.

B. P2P Cache Capacity Allocation Policies

The ISPs, as they are located in the lower layers of the ISP hierarchy, are interested in decreasing the inter-ISP traffic generated by the peers. In order to decrease its inter-ISP traffic, ISP $i \in \mathcal{I}$ maintains a cache with upload *bandwidth capacity* $K_i < \infty$, which acts as an ISP managed super peer [8]. The abstraction of a P2P cache as a source of upload bandwidth is motivated by that P2P caches are often priced by their maximum upload rates. Since every ISP's goal is to decrease its own incoming inter-ISP traffic, it is reasonable to assume that the cache operated by ISP i only serves leechers in ISP i.

ISP i can implement an *active* cache bandwidth allocation policy to control the amount of cache bandwidth $\kappa_{i,s}(t)$ available to leechers in ISP i belonging to swarm s. We denote the cache bandwidth allocation of ISP i at time t by the vector $\kappa_i(t) = (\kappa_{i,1}(t), \ldots, \kappa_{i,S}(t))$, and the set of feasible cache bandwidth allocations of ISP i by $\mathcal{K}_i = \{\kappa_i | \sum_{s\in\mathcal{S}} \kappa_{i,s} \leq K_i\} \subseteq [0, K_i]^{|\mathcal{S}|}$. We also make the reasonable assumption that $\kappa_{i,s}(t) > 0$ for a swarm s only if the corresponding file is at least partially cached at ISP i at time t.

Given the set $\mathcal{K}_i$ of feasible cache bandwidth allocations for ISP i, a cache bandwidth allocation *policy* π defines $\kappa_i(t)$ as a function of the system's history up to time t, i.e., $(Z(u))_{u<t}$, and past cache allocations $(\kappa_i(u))_{u<t}$. We denote the set of all cache bandwidth allocation policies by Π.

C. Caching and System Dynamics

Consider a policy π implemented by ISP i. We model the evolution of the swarms' state by an $I \times S \times 2$ dimensional continuous-time Markov jump process $\mathcal{Z}^\pi = \{Z(t), t \geq 0\}$, which is a collection of S coupled $I \times 2$ dimensional continuous-time Markov jump processes $\mathcal{Z}_s^\pi = \{Z_s(t), t \geq 0\}$.

Consider now a swarm $s \in \mathcal{S}$ under policy π, and denote the transition intensity from state z_s to state z_s' by $q^\pi_{z_s,z_s'}$. Denote by e_i the I dimensional vector whose i^{th} component is 1. The transition intensities from state $z_s = (x_s, y_s)$ are $q^\pi_{z_s,(x_s+e_i,y_s)} = \lambda_{i,s}$ (leecher arrival), $q^\pi_{z_s,(x_s-e_i,y_s)} = \theta x_{i,s}$ (leecher abort), and $q^\pi_{z_s,(x_s,y_s-e_i)} = \gamma y_{i,s}$ (seed departure). The transition intensity to state $(x_s - e_i, y_s + e_i)$, called the download completion rate, is a function of the maximum download rate of the leechers, and the available upload rate to leechers in ISP i.

1) The case of no cache: Without a cache ($K_i = 0$) the leechers in ISP i would get a share $x_{i,s}/\sum_i x_{i,s}$ of the total upload rate $u_s^P = \sum_i u_{i,s}^P$ [21], [22], [24], [25]. The download completion rate in this case can be expressed as

$$q^\pi_{(x_s,y_s),(x_s-e_i,y_s+e_i)} = \min(cx_{i,s}, u_s^P x_{i,s}/\textstyle\sum_i x_{i,s}). \quad (1)$$

We refer to the process defined this way as the *uncontrolled* stochastic process, and we denote it by $\mathcal{Z}$.

2) The case of cache: Consider that the instantaneous cache bandwidth allocated to swarm s is $\kappa_{i,s}$. The cache bandwidth increases the available upload rate, so that the download completion rate becomes

$$q^\pi_{(x_s,y_s),(x_s-e_i,y_s+e_i)} = \min(cx_{i,s}, u_s^P x_{i,s}/\textstyle\sum_i x_{i,s} + \kappa_{i,s}). \quad (2)$$

Since the cache bandwidth allocation can influence the transition intensities of the stochastic process, we refer to $\mathcal{Z}^\pi$ as the *controlled* stochastic process.

D. The Optimal Cache Capacity Allocation Problem

Let us denote by $I_{i,s}(Z_s(t), \kappa_{i,s}(t))$ the rate of the incoming inter-ISP traffic in ISP i due to swarm s as a function of the cache bandwidth $\kappa_{i,s}(t)$ allocated to swarm s by ISP i and the swarms' state $Z_s(t)$. $I_{i,s}(Z_s(t), \kappa_{i,s}(t))$ also depends on $\kappa_{j,s}(t)$ of ISPs $j \neq i$, but as we focus on the bandwidth allocation problem of ISP i, for simplicity we assume that $\kappa_{j,s}(t) = \kappa_{j,s}$ constant.

We can express the expected amount of incoming inter-ISP traffic under policy $\pi \in \Pi$ from time $t = 0$ until time T as

$$C_i^\pi(z, T) = E_z^\pi \left[\int_0^T \sum_{s\in\mathcal{S}} I_{i,s}(Z_s(t), \kappa_{i,s}(t)) dt \right],$$

where E_z^π denotes the expectation under policy π with initial state $Z(0) = z$.

Given the set Π of feasible cache bandwidth allocation policies, we define the cache bandwidth allocation problem for ISP i as finding the cache bandwidth allocation policy $\pi^* \in \Pi$ that minimizes the average incoming inter-ISP traffic rate $C_i^\pi(z)$ due to P2P content distribution, that is

$$\inf_{\pi\in\Pi} C_i^\pi(z) = \inf_\pi \limsup_{T\to\infty} \frac{1}{T} C_i^\pi(z, T). \quad (3)$$

We address three important questions related to cache bandwidth allocation in the following. First, is there an optimal policy π^* that solves (3). Second, what are the properties of the optimal allocation policy. Third, can an ISP benefit significantly from actively allocating its cache bandwidth.

IV. Adaptive Cache Bandwidth Allocation Policies

In this section we first discuss a baseline for bandwidth sharing. We then show the existence of an optimal stationary policy for the cache bandwidth allocation problem, and describe two approximations to the optimal policy. Throughout the section we assume that the inter-ISP traffic functions $I_{i,s}(z_s, \kappa_{i,s})$ are known, and are continuous convex non-increasing functions of $\kappa_{i,s}$. The assumptions of continuity, convexity and non-increasingness are rather natural. In Section V we describe a simple model of inter-ISP traffic, and use simulations and experiments to support the assumptions.

A. Demand-driven Capacity Sharing (DDS)

As a baseline for comparison, consider that ISP i does *not* actively allocate its cache bandwidth K_i. The cache in ISP i maintains a drop-tail queue to store the requests received from the leechers in ISP i, and serves the requests according to a first-in-first-out (FIFO) policy at the available upload bandwidth K_i. Consider that the $x_{i,s}$ leechers of swarm s in ISP i request data from the cache in ISP i at rate $\alpha_{i,s}$, and denote by $\sigma_{i,s}$ the mean service time of these requests. Then the offered load of swarm s to the cache is $\rho_{i,s} = \alpha_{i,s}\sigma_{i,s}$.

Clearly, if $\rho_{i,s} \geq 1$ then the FIFO queue is in a blocking state with probability $p_i^b > 0$.

If the requests from leechers in every swarm arrive according to a Poisson process, then the aggregate arrival process is Poisson. Since the arrival process is Poisson, an arbitrary request is blocked (i.e., dropped) with probability $p_{i,s}^b = p_i^b$ despite the possibly heterogeneous mean service times due to the PASTA property [27]. The effective (i.e., not blocked) load for swarm s can be expressed as $(1-p_i^b)\rho_{i,s}$, and consequently the share of cache bandwidth used to serve requests for swarm s can be estimated as

$$\frac{\kappa_{i,s}}{\sum_{s\in\mathcal{S}}\kappa_{i,s}} = \frac{(1-p_i^b)\rho_s}{\sum_{s\in\mathcal{S}}(1-p_i^b)\rho_s} = \frac{\rho_s}{\sum_{s\in\mathcal{S}}\rho_s}. \tag{4}$$

In general, if the arrival process of requests is not Poisson then (4) does not hold. Nevertheless, as under the assumption of a Poisson request arrival process the cache bandwidth is shared among the swarms proportional to the offered load (demand) of the swarms, we refer to this policy as the *demand-driven sharing* (*DDS*) policy.

B. Optimal Cache Capacity Allocation as a MDP

Although the primary goal of ISP i when allocating cache bandwidth to swarm s is to decrease the inter-ISP traffic, it inherently affects the upload rate available to the leechers, and hence, it can affect the evolution of the process $\mathcal{Z}_s^\pi$. Consequently, the optimal cache bandwidth allocation problem is a continuous-time Markov decision process (MDP) with the optimality criterion defined in (3).

The first two fundamental questions that we are to answer are (i) whether there is an optimal cache bandwidth allocation policy π^* that solves (3), and (ii) whether there is an optimal policy whose choices only depend on the *current* system state $Z(t)$. Such a policy is called *stationary*. In general, an optimal stationary policy might not exist for a MDP when the action space or the state space is infinite. The following theorem shows that for the cache bandwidth allocation problem there exists an optimal stationary policy.

Theorem 1: There exists an optimal stationary policy π^* that minimizes the average traffic $C_i^\pi(z)$ of ISP i.

The proof of the Theorem is in the Appendix. A consequence of Theorem 1 is that the optimal bandwidth allocation policy π^* is such that the allocation $\kappa_i(t)$ is only a function of the system state $Z(t)$, hence it is constant between the state transitions of $\mathcal{Z}^{\pi*}$.

The optimal policy π^* can be found using the policy iteration algorithm [28], but it requires the solution of the steady state probabilities of the controlled Markov processes $\mathcal{Z}^\pi$. This can be prohibitive even for a moderate number of ISPs and swarms. We therefore consider two approximations in the following.

C. One-step Look Ahead Allocation Policy (OLA)

The one-step look ahead (OLA) policy π^{OLA} is a simple approximation of the optimal stationary cache bandwidth allocation policy π^*.

Consider the controlled Markov process $\mathcal{Z}^{\pi^{OLA}}$, and let us denote the n^{th} transition epoch of the process by t_n. Then according to the OLA policy the cache bandwidth allocation $\kappa_i(t)$ of ISP i for $t_n < t \leq t_{n+1}$ is such that it minimizes the incoming inter-ISP traffic rate given the state $Z(t_n) = z$ of the process $\mathcal{Z}^{\pi^{OLA}}$

$$\kappa_i(t) = \arg\min_{\kappa_i\in\mathcal{K}_i} \sum_{s\in\mathcal{S}} I_{i,s}(z_s, \kappa_{i,s}). \tag{5}$$

By following the OLA policy the ISP minimizes the incoming inter-ISP traffic in every state of the process $\mathcal{Z}^{\pi^{OLA}}$. The OLA policy *adapts* to the system state, but unlike the optimal policy π^*, it does not consider the impact of cache bandwidth allocation on the evolution of the number of peers.

Recall that, by assumption, $I_{i,s}(z_s, \kappa_{i,s})$ are continuous convex non-increasing functions of $\kappa_{i,s}$ for every state z_s. In order to obtain the optimal solution to (5) consider the Lagrangian

$$L(z, \kappa_i, \zeta) = \sum_{s\in\mathcal{S}} I_{i,s}(z_s, \kappa_{i,s}) - \zeta\Big(\sum_{s\in\mathcal{S}} \kappa_{i,s} - K_i\Big), \tag{6}$$

where $\zeta \leq 0$ is the Lagrange multiplier. Then

$$\frac{\partial L(z, \kappa_i, \zeta)}{\partial \kappa_{i,s}} = \frac{\partial I_{i,s}(z_s, \kappa_{i,s})}{\partial \kappa_{i,s}} - \zeta \tag{7}$$

and

$$\frac{\partial L(z, \kappa_i, \zeta)}{\partial \zeta} = K_i - \sum_{s\in\mathcal{S}} \kappa_{i,s}. \tag{8}$$

Hence, a minimum of L over $\mathcal{K}_i$ is characterized by

$$\begin{aligned} \kappa_{i,s} > 0 \quad &\Rightarrow \quad \frac{\partial_+ I_{i,s}(z_s, \kappa_{i,s})}{\partial \kappa_{i,s}} \geq \zeta \geq \frac{\partial_- I_{i,s}(z_s, \kappa_{i,s})}{\partial \kappa_{i,s}} \\ \kappa_{i,s} = 0 \quad &\Rightarrow \quad \frac{\partial_- I_{i,s}(z_s, \kappa_{i,s})}{\partial \kappa_{i,s}} \geq \zeta, \end{aligned}$$

where ∂_+ and ∂_- denote the right and the left derivative of a semi-differentiable function. Since $\mathcal{K}_i$ is compact and convex, such a minimum exists and can be found using a projected subgradient method [29].

An important insight from the OLA policy is the following. If $I_{i,s}(z_s, \kappa_{i,s})$ are continuously differentiable then at optimality every swarm with non-zero cache bandwidth allocation provides equal marginal traffic saving. If $I_{i,s}(z_s, \kappa_{i,s})$ are not continuously differentiable, then for swarms with non-zero cache bandwidth allocation the intersection of the subdifferentials is non-empty.

D. Steady-state Optimal Allocation Policy (SSO)

The opposite of the *OLA* policy is to focus on the long-term evolution of the controlled Markov process $\mathcal{Z}^\pi$, that is, on the incoming inter-ISP traffic in steady-state and to consider time-independent cache bandwidth allocation policies $\overline{\pi} = \kappa_i$.

Let us denote the expected number of leechers and seeds in steady-state as a function of the cache bandwidth allocation policy $\overline{\pi}$ by $\bar{x}_{i,s}^{\overline{\pi}}$ and by $\bar{y}_{i,s}^{\overline{\pi}}$, respectively. They were shown to be a function of the cache upload rate $\kappa_{i,s}$ allocated to

swarm s [13]. As long as the total available upload rate is less than or equal to the total download rate of the leechers

$$\bar{x}_{i,s}^{\overline{\pi}} = \frac{\lambda_{i,s}}{\nu\left(1+\frac{\theta}{\nu}\right)} - \frac{\kappa_{i,s}}{\mu\eta\left(1+\frac{\theta}{\nu}\right)} - \Delta_i(\mathbf{x},\mathbf{y},\kappa) \quad (9)$$

$$\bar{y}_{i,s}^{\overline{\pi}} = \frac{\lambda_{i,s}}{\gamma\left(1+\frac{\theta}{\nu}\right)} + \frac{\kappa_{i,s}\theta}{\mu\eta\gamma\left(1+\frac{\theta}{\nu}\right)} + \frac{\theta}{\gamma}\Delta_i(\mathbf{x},\mathbf{y},\kappa), \quad (10)$$

where $\frac{1}{\nu} = \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma}) \geq 0$ [13], [22] and

$$\Delta_i(\mathbf{x},\mathbf{y},\kappa) = \frac{\sum_{j\in\mathcal{I}}(\lambda_{i,s}\kappa_{j,s} - \kappa_{i,s}\lambda_{j,s})}{\eta\gamma\left(1+\frac{\theta}{\nu}\right)\left(\sum_{j\in\mathcal{I}}(\lambda_{j,s}-\kappa_{j,s})\right)}. \quad (11)$$

Otherwise, when the total upload rate exceeds the total download rate, increasing the cache bandwidth allocated to the swarm does not affect the number of leechers and seeds in steady-state and their number is [13], [22]

$$\bar{x}_{i,s}^{\overline{\pi}} = \frac{\lambda_{i,s}}{c(1+\frac{\theta}{c})} \qquad \bar{y}_{i,s}^{\overline{\pi}} = \frac{\lambda_{i,s}}{\gamma(1+\frac{\theta}{c})}. \quad (12)$$

It is easy to verify that $\frac{\partial \bar{x}_{i,s}}{\partial \kappa_{i,s}} \leq 0$ and that $\frac{\partial^2 \bar{x}_{i,s}}{\partial \kappa_{i,s}^2} \geq 0$ for $\kappa_{i,s} \geq 0$, that is, the number of leechers in swarm s in ISP i in steady-state is a convex non-increasing function of the cache bandwidth allocated to swarm s in ISP i.

Given the functions $\bar{x}_{i,s}^{\pi}$ and $\bar{y}_{i,s}^{\pi}$ the steady-state optimal (*SSO*) bandwidth allocation policy can be formulated as

$$\overline{\pi}^* = \arg\min_{\kappa_i\in\mathcal{K}_i} \sum_{s\in\mathcal{S}} \bar{I}_{i,s}(\kappa_{i,s}), \quad (13)$$

where $\bar{I}_{i,s}(\kappa_{i,s})$ is the incoming inter-ISP traffic rate for the number of leechers and seeds in steady-state.

Since by assumption $I_{i,s}(z_s,\kappa_{i,s})$ is convex non-increasing in $\kappa_{i,s}$ for every state z_s, the steady-state optimal policy $\overline{\pi}^*$ can be found in a similar way as the OLA policy. The difference is that $\bar{I}_{i,s}(\kappa_{i,s})$ is a function of $\kappa_{i,s}$, $\bar{x}_{i,s}^{\pi}$ and $\bar{y}_{i,s}^{\pi}$, and the latter are themselves functions of $\kappa_{i,s}$. Note that the steady-state optimal policy $\overline{\pi}^*$ is not equivalent to the optimal policy π^* of the MDP, as the cache bandwidth allocated to a swarm s in ISP i would be nonzero even when $x_{i,s}(t) = 0$, which happens with nonzero probability.

V. Performance Evaluation and Insights

In the following we use simulations and experiments to compare the two approximate cache bandwidth allocation policies to DDS, and to provide insight into the characteristics of an optimal cache bandwidth allocation policy.

A. Transit Traffic Model

In Section IV we assumed that the incoming inter-ISP traffic function $I_{i,s}(z_s,\kappa_{i,s})$ is known. In the following we describe an approximate model of the incoming inter-ISP traffic, which we use to implement the *OLA* and the *SSO* policies. As the model is for a single swarm, we omit the subscript s for clarity.

The model is based on two assumptions. First, leechers compete with each other for the available upload rate as long as they would be able to download at a higher rate. Second, given a single byte downloaded in ISP i, the distribution of its sources is proportional to the amount of upload rate exposed to the leechers that are located in ISP i.

The leechers in ISP i demand data at a total rate of cx_i. As the cache appears as an arbitrary peer to the leechers in ISP i, the demand is directed to the upload rate κ_i of ISP i's cache and to the publicly available upload rate $u_i^{PL} + \sum_{j\neq i} u_j^P$ of all ISPs. The leechers demand from the cache's upload rate with a probability proportional to its value, i.e, with probability $\kappa_i/(u_i^{PL} + \sum_{j\neq i} u_j^P + \kappa_i)$. The rest they demand from the publicly available upload rate, so the rate D_i^d that leechers in ISP i demand from the publicly available upload rate can be expressed as

$$D_i^d = cx_i\left(1 - \frac{\kappa_i}{u_i^{PL} + \sum_{j\neq i} u_j^P + \kappa_i}\right). \quad (14)$$

If the system is limited by the download rate of the leechers, then the leechers receive the demanded rate. If the system is limited by the available upload rate, then the rate at which the leechers receive is proportional to the total publicly available upload rate divided by the total demanded rate

$$D_i^r = D_i^d min\left(1, \frac{\sum_j u_j^P}{\sum_j D_j^d}\right). \quad (15)$$

The rate that the leechers receive can originate from any ISP. Using the assumption that for a single byte downloaded in ISP i, the distribution of its sources is proportional to the amount of upload rate exposed to leechers in ISP i we get the following estimate of the incoming inter-ISP traffic of ISP i

$$I_i(z_s,\kappa_i) = D_i^r\left(\frac{\sum_{j\neq i} u_j^P}{u_i^{PL} + \sum_{j\neq i} u_j^P}\right). \quad (16)$$

$I_i(z_s,\kappa_i)$ defined by (14) to (16) is a continuous convex non-increasing function of the cache bandwidth κ_i allocated by ISP i. In the following we use this model of the incoming inter-ISP traffic to implement the *OLA* and *SSO* policies defined in the previous section.

B. Simulation Results

We used the P2P simulation and prototyping tool-kit ProtoPeer and the corresponding library for BitTorrent [30], [31] for the simulations. The simulations are flow-level: data transmissions are flows and the bandwidth for each flow is calculated according to the max-min-fair-share principle [32], an approximation of the bandwidth sharing behavior of TCP.

We simulate 12 to 15 BitTorrent swarms, each sharing a file of 150MB. The number of swarms is large enough to show the impact of the policies. At the same time, it keeps the runtime of the simulations at a reasonable level of a few hours per simulation run. The peers have an access bandwidth of 1Mbit/s upstream and 16Mbit/s downstream. The peers join swarm s in ISP i according to a Poisson process at a rate of $\lambda_{i,s}$. After completing the download they remain in the swarm for an exponentially distributed seeding time with average $1/\gamma = 10$ minutes. We simulate $I = 2$ ISPs. ISP 1 is the tagged ISP and ISP 2 is the aggregation of all other ISPs in the network.

Scenario	Number of swarms (S)	Identical swarms (s)	$\frac{\lambda_s}{\lambda}$	$\frac{\lambda_{2,s}}{\lambda_{1,s}}$
unif.,1:10	12	1,..,12	1/12	10
zipf,1:10	12		$\propto \frac{1}{s}$	10
unif.,1:1+1:10	12	1,..,10	1/12	10
		11,12	1/12	1
het.,2:2+1:10	15	1,..,4	1/8	10
		5,..,15	1/22	1

TABLE I
RELATIVE PEER ARRIVAL RATES IN THE SIMULATED SCENARIOS.

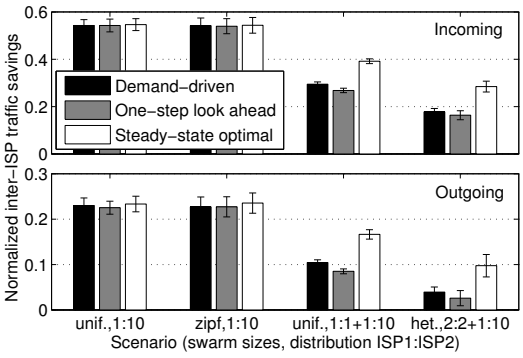

Fig. 1. Incoming and outgoing inter-ISP traffic savings for the four scenarios and three policies for $K_1 = 30$Mbit/s. Simulation results.

Such aggregation of the ISPs was shown to provide accurate results in [13]. We simulate the cache of ISP 1 as a BitTorrent seed with upload bandwidth K_1. The cache joins all swarms, but uploads only to leechers in ISP 1. The different cache bandwidth allocation policies are implemented in the peer.

Every simulation run corresponds to 6.5 hours of simulated time, and we use the results following a warm-up period of 1.5 hours. For every configuration we show the average of 5 simulation runs together with the 95%-confidence intervals.

1) Cache bandwidth allocation matters: We consider four scenarios to investigate under what conditions active cache bandwidth allocation can be beneficial. For simplicity, we denote the total arrival rate by $\lambda = \sum_i \sum_s \lambda_{i,s}$. We use the same total arrival rate $\lambda = 30$/min for all four scenarios, but the four scenarios differ in terms of the arrival rates $\lambda_{i,s}$ of the peers between swarms and between ISPs. Table I shows the relative arrival rates for the four scenarios. The scenarios are constructed so that they allow us to isolate the factors that influence the efficiency of cache bandwidth allocation policies.

As an example, in scenario *unif.,1:1+1:10* all $S = 12$ swarms have the same arrival rate $\lambda_s = \lambda/12$. The arrival rates for swarms 1 to 10 are asymmetric ($\lambda_{2,s} = 10\lambda_{1,s}$), while for swarms 11 and 12 they are symmetric ($\lambda_{2,s} = \lambda_{1,s}$). The use of Zipf's law for the arrival intensities in scenario *zipf,1:10* is motivated by recent measurements that show that the distribution of the number of concurrent peers over swarms exhibits Zipf like characteristics over a wide range of swarm sizes [33], [34]. Symmetric and asymmetric swarms are motivated by measurements that show the difference in terms of the spatial distribution of peers between contents of regional and of global interest (e.g., the popularity of movies depending on the language [34]).

Fig. 1 shows the normalized incoming and outgoing inter-ISP traffic saving of ISP 1 for the four scenarios for the *DDS*, *OLA* and *SSO* allocation policies. We calculate the normalized inter-ISP traffic saving as the decrease of the average inter-ISP traffic due to installing a cache divided by the average inter-ISP traffic without a cache ($K_1 = 0$), that is, $(C_i|_{K_1=0} - C_i^{\pi})/C_i|_{K_1=0}$. The upload bandwidth of the cache in ISP 1 is $K_1 = 30$Mbit/s.

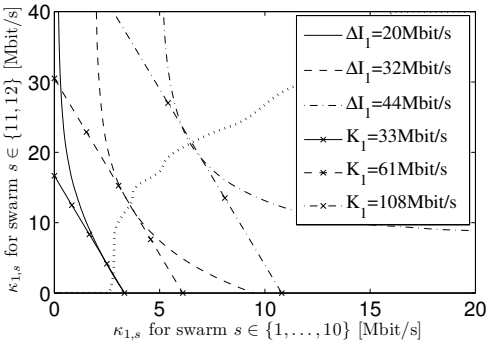

Fig. 2. Indifference map of ISP 1 for the *unif.,1:1+1:10* scenario based on simulation results. The dotted line shows the *SSO* cache bandwidth allocation for different values of cache bandwidth K_1.

For the *unif.,1:10* and the *zipf,1:10* scenarios, in which the ratio $\lambda_{2,s}/\lambda_{1,s} = 10$ is the same for all swarms, the difference between the results for the different cache bandwidth allocation policies is within the confidence interval. However, for the scenarios *unif.,1:1+1:10* and *het.,2:2+1:10* the bandwidth allocation policies make a significant difference in terms of traffic savings, both in terms of incoming and outgoing inter-ISP traffic. These results indicate that cache bandwidth allocation affects the transit traffic savings when the distribution of the peers over the ISPs is different among swarms, as for the *unif.,1:1+1:10* and the *het.,2:2+1:10* scenarios.

Comparing the different policies in Fig. 1 for the *unif.,1:1+1:10* scenario reveals that the *OLA* and the *SSO* policies have opposite effects on the inter-ISP traffic saving due to caching. The *OLA* policy performs worse than *DDS*, but the *SSO* policy compared to *DDS* increases the incoming inter-ISP traffic savings by about 33 percent and the outgoing inter-ISP traffic savings by over 60 percent. For the *het.,2:2+1:10* scenario the savings increase by 60 and 248 percent, respectively. Considering that P2P cache eviction policies achieve within 10 to 20 percent of the hit rate of the optimal off-line eviction policy [10], [11], the 30 to 60 percent decrease of the incoming inter-ISP traffic achieved through cache bandwidth allocation is more than what could be achieved through improved cache eviction policies. Fig. 2 shows the indifference map of ISP 1 for the *unif.,1:1+1:10* scenario, and illustrates the *SSO* bandwidth allocation policy. The horizontal and the vertical axes show the cache bandwidth allocated to each of the 10 asymmetric ($\lambda_{2,s} = 10\lambda_{1,s}$) and to each of the 2 symmetric ($\lambda_{2,s} = \lambda_{1,s}$) swarms, respectively. The curves show combinations of bandwidth allocations that lead to a particular transit traffic saving ΔI_1 (i.e., was there no cache bandwidth constraint K_1, ISP 1 would be indifferent between allocations on the same indifference curve). The straight diagonal lines show different cache bandwidth constraints K_1. The *SSO* cache

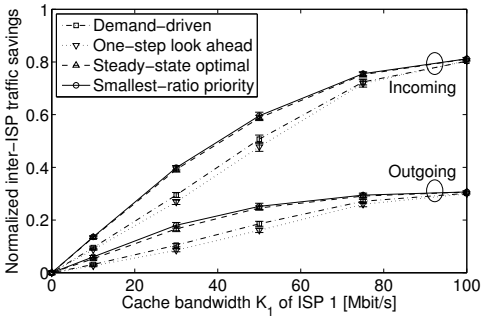

Fig. 3. Incoming and outgoing inter-ISP traffic saving for the *unif.,1:1+1:10* scenario vs. cache bandwidth in ISP 1. Simulation results.

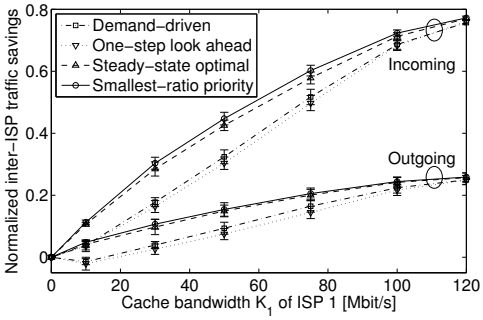

Fig. 4. Incoming and outgoing inter-ISP traffic savings for the *het.,2:2+1:10* scenario vs. cache bandwidth in ISP 1. Simulation results.

bandwidth allocation for K_1 is given by the coordinates of the point at which the cache bandwidth constraint line for K_1 is tangent to the indifference curve. The dotted line connects all such points: it shows the *SSO* cache bandwidth allocation for different K_1. We note that for $K_1 \leq 30$Mbit/s all cache bandwidth should be allocated to the 10 asymmetric swarms, above that, as K_1 increases so does the bandwidth that should be allocated to the 2 symmetric swarms. We also note that the shape of the indifference curves confirms that the inter-ISP traffic saving for a single swarm is a concave non-decreasing function of $\kappa_{i,s}$.

2) Smallest-ratio Priority Allocation: An intriguing question is whether, in general, allocating all bandwidth to asymmetric swarms is steady-state optimal for small cache bandwidths. To answer this question, consider a single swarm spread over two ISPs, $\mathcal{I} = \{1,2\}$, and denote the ratio of the arrival rates in the two ISPs by $r = \lambda_2/\lambda_1$.

Our focus will be on how the partial derivative $\frac{\partial \bar{I}_1(\kappa_1)}{\partial \kappa_1}$ of the steady-state inter-ISP traffic depends on r. For small κ_1 the incoming inter-ISP traffic $I_1(z,\kappa_1)$ of ISP 1 defined by (14) to (16) can be approximated by

$$I_1(z,\kappa_1) \approx \frac{x_1}{x_1+x_2} u_2^P. \tag{17}$$

We consider the case when the system is limited by the available upload rate, so we substitute (9) and (10) into (17) to obtain an approximation of the steady-state incoming inter-ISP traffic $\bar{I}_1(\kappa_1)$ of ISP 1 as a function of the cache bandwidth. Consider now the derivative at $\kappa_1 = 0$ and $\kappa_2 = 0$

$$\frac{\partial \bar{I}_1(\kappa_1)}{\partial \kappa_1}\Big|_{\substack{\kappa_1=0\\ \kappa_2=0}} = -\frac{r^2(\gamma+\nu)(\gamma-\mu) - r\mu(\theta-\gamma)}{(1+\frac{\theta}{\nu})\mu\eta\gamma^2(1+r)^2}. \tag{18}$$

Recall that $\gamma - \mu > 0$ is a necessary condition for the upload rate to be the limit, and it implies $\nu > 0$ [13], [22]. Hence for $\theta - \gamma \leq 0$, (18) is negative and decreases monotonically in r.

For $\theta - \gamma > 0$ we have to consider the mixed second order partial derivative at $\kappa_1 = 0$ and $\kappa_2 = 0$

$$\frac{\partial^2 \bar{I}_1(\kappa_1)}{\partial \kappa_1 \partial r}\Big|_{\substack{\kappa_1=0\\ \kappa_2=0}} = -\frac{2r(\gamma+\nu)(\gamma-\mu) + (r-1)\mu(\theta-\gamma)}{(1+\frac{\theta}{\nu})\mu\eta\gamma^2(1+r)^3}. \tag{19}$$

Since $\theta - \gamma > 0$, (19) is negative for $r \geq 1$. Consequently allocating cache bandwidth to swarms with a higher ratio r of arrival rates leads to a faster decrease of the steady-state inter-ISP traffic. At the same time, due to the term $(1+r)^3$ in the denominator $\lim_{r\to\infty} \frac{\partial^2 \bar{I}_1(\kappa_1)}{\partial \kappa_1 \partial r}|_{\kappa_1=0} = 0$, i.e, swarms with a high arrival ratio r provide approximately the same gain.

This approximation suggests that a priority-based policy that assigns the highest priority to the swarms with highest ratio $r = \lambda_2/\lambda_1$ would resemble the *SSO* allocation policy. We use this insight to define the *smallest-ratio priority* (*SRP*) cache bandwidth allocation policy. Under *SRP* the priority of a swarm is calculated based on the instantaneous ratio of the local leechers to the number of peers in the overlay outside of ISP i, $\hat{r}_{i,s} = \frac{x_{i,s}(t)}{\sum_{j\neq i} z_{j,s}(t)}$. The priority of swarms with $\hat{r}_{i,s} = 0$ and $\hat{r}_{i,s} = \infty$ is lowest, and the priorities of the remaining swarms are assigned in decreasing order of the ratios $\hat{r}_{i,s}$. We implemented the *SRP* policy in the simulator by assigning a priority level to every data flow and by modifying the bandwidth sharing algorithm. The bandwidth of flows with the same priority is calculated according to the original max-min-fair-share algorithm, while flows with a lower priority can only use the link bandwidth not used by flows with higher priority.

Fig. 3 shows the incoming and outgoing inter-ISP traffic savings normalized by the inter-ISP traffic without cache ($K_1 = 0$) for the *unif.,1:1+1:10* scenario as a function of the cache bandwidth K_1. The figure confirms that the observations made in Fig. 1 hold for a wide range of cache bandwidths K_1. Only above $K_1 \approx 75$Mbit/s, when the available upload bandwidth in ISP 1 exceeds the aggregate download bandwidth of the leechers within ISP 1, the marginal traffic saving diminishes and so does the difference between the policies. We note that the *SRP* policy performs slightly better than the *SSO* policy for all cache bandwidths. This is because the the *SSO* allocation can be far from optimal when the instantaneous number of peers in the system is far away from the steady-state average number of peers.

Fig. 4 shows the incoming and outgoing inter-ISP traffic saving normalized by the inter-ISP traffic without cache ($K_1 = 0$) for the *het.,2:2+1:10* scenario as a function of the cache bandwidth K_1. The figure allows us to draw similar conclusions as Fig. 3, except for the dip in the outgoing inter-ISP traffic saving for *DDS* at $K_1 = 10$Mbit/s. While surprising at first sight, the potential increase of the outgoing inter-ISP traffic due to caching for small, symmetric swarms (i.e., swarms 5 to 15) was pointed out in [13]. Since the *SRP* and the *SSO* policies allow little cache bandwidth to be used by the symmetric swarms for low K_1, they provide outgoing inter-ISP traffic savings even at $K_1 = 10$Mbit/s.

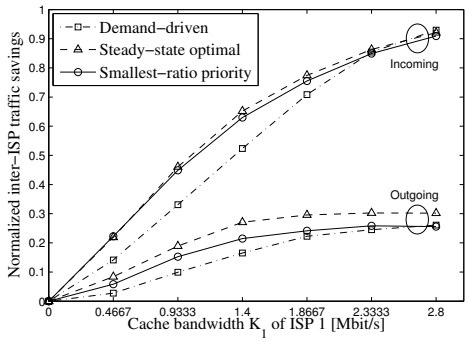

Fig. 5. Incoming and outgoing inter-ISP traffic saving for the *unif.,1:1+1:10* scenario vs. cache bandwidth in ISP 1. Experiment results.

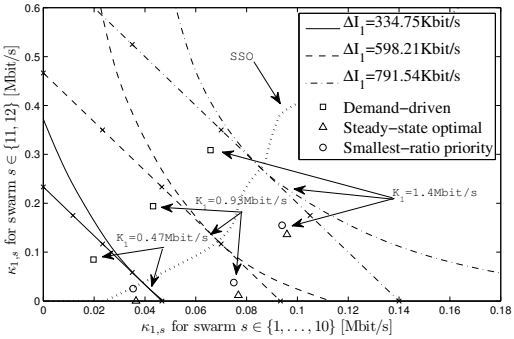

Fig. 6. Indifference map of ISP 1 for the *unif.,1:1+1:10* scenario based on the experiments, and the actual average cache upload rates for the *DDS*, *SSO* and *SRP* policies. Experiment results.

C. Experimental Validation on Planet-lab

As a proof of concept and to validate the simulation results we performed experiments involving approximately 500 Planet-lab nodes using BitTorrent. For the experiments we used the *unif.,1:1+1:10* scenario. We scaled down the file size, the upload rates and the download rates by a factor of 43 compared to the simulations in order to avoid interfering with other Planet-lab traffic: the file size was 3.5MB, and the upload and download bandwidths of the peers were 23kbit/s and 373kbit/s, respectively. Every experiment ran for 4 hours, and we use the results after an initial warm-up period of 1 hour.

For every swarm we assigned every Planet-lab node to one of the two ISPs, and measured the traffic exchanged between peers belonging to different ISPs. We used one peer per swarm as the cache of ISP 1; these 12 peers ran on a dedicated Linux computer. We implemented the cache bandwidth allocation policies using hierarchical token bucket (HTB) queues in Linux traffic control. We used one filter per swarm to redirect the upload traffic of the 12 peers to a HTB class that enforced the total cache upload bandwidth limit K_1. For the *SSO* and the *SRP* policies we attached to this class one subclass per swarm. By default each subclass had 500B/s of guaranteed bandwidth in order to keep the TCP connections alive. The actual priority and guaranteed bitrate was then set according to the cache bandwidth allocation policy. The excess bandwidth was distributed among the swarms as defined by the HTB queue. For *SRP* we updated the priorities every 10 seconds based on the average number of leechers and seeds over the preceding 30 seconds.

Fig. 5 shows the incoming and the outgoing inter-ISP traffic savings normalized by the inter-ISP traffic without cache ($K_1 = 0$) for the *unif.,1:1+1:10* scenario as a function of the cache bandwidth K_1. The experiments match the corresponding simulation results (cf. Fig. 3) and confirm the significant gain of cache bandwidth allocation observed in the simulations. The only difference is that the *SRP* policy performs slightly worse than in the simulations, which is due to the impact of the network layer implementation of bandwidth allocation and priorities on TCP congestion control. An application layer implementation of the policy could prevent this.

Fig. 6 shows the indifference map and the actual average cache upload rate received by the asymmetric (horizontal) and the symmetric (vertical) swarms under the three allocation policies. There is one marker per policy and total cache bandwidth K_1. The figure shows how the cache upload rate received by the swarms differs under the three policies depending on the cache bandwidth limit K_1. Under both *SRP* and *SSO* the cache uploads to the symmetric swarms at a significantly lower rate than under *DDS* except for very high K_1, which is the key to the higher inter-ISP traffic savings of both policies.

VI. Conclusion

Motivated by the large amount of inter-ISP P2P traffic, we investigated a new dimension of P2P cache resource management, the allocation of cache upload bandwidth between overlays. We formulated the problem of cache bandwidth allocation as a Markov decision process, and showed the existence of an optimal stationary allocation policy. Through comparing two fundamentally different approximations of the optimal allocation policy we demonstrated the importance of capturing the cache's impact on the swarm dynamics. We showed that cache bandwidth allocation can lead to significantly decreased inter-ISP traffic, and identified the heterogeneity of the swarm's distribution between ISPs as the primary factor that influences the potential of bandwidth allocation. Based on the insights obtained from the analysis of the policies we proposed a simple, priority-based cache bandwidth allocation policy that performed well both in simulations and in experiments with BitTorrent clients on Planet-lab.

VII. Acknowledgement

The authors thank Alexandre Proutiere for his comments on the drafts of this work.

References

[1] H. Schulze and K. Mochalski, "Internet Study 2008/2009," 2009. [Online]. Available: http://www.ipoque.com/resources/internet-studies

[2] P. Eckersley, F. von Lohmann, and S. Schoen, "Packet forgery by ISPs: A report on the Comcast affair," White paper, Nov. 2007.

[3] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can ISPs and P2P systems co-operate for improved performance?" *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 29–40, Jul. 2007.

[4] 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, Oct. 2008.

[5] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, "Improving traffic locality in BitTorrent via biased neighbor selection," in *Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS)*, Jul. 2006, pp. 66–75.

[6] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: a practical approach to reducing cross-ISP traffic in peer-to-peer systems," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 363–374, Oct. 2008.
[7] S. L. Blond, A. Legout, and W. Dabbous, "Pushing bittorrent locality to the limit," *Computer Networks*, vol. 55, no. 3, pp. 541–557, 2011.
[8] OverCache P2P, "http://www.oversi.com."
[9] PeerApp UltraBand, "http://www.peerapp.com."
[10] A. Wierzbicki, N. Leibowitz, M. Ripeanu, and R. Woźniak, "Cache replacement policies for P2P file sharing protocols," *Euro. Trans. on Telecomms.*, vol. 15, pp. 559–569, Nov. 2004.
[11] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1447–1460, Dec. 2008.
[12] N. Megiddo and D. Modha, "ARC: A self-tuning, low overhead replacement cache," in *Proc. of USENIX File & Storage Technologies Conference (FAST)*, March 2003.
[13] F. Lehrieder, G. Dán, T. Hoßfeld, S. Oechsner, and V. Singeorzan, "The impact of caching on BitTorrent-like peer-to-peer systems," in *Proc. IEEE Int'l Conf. Peer-to-Peer Computing (P2P)*, Aug. 2010.
[14] G. Dán, T. Hoßfeld, S. Oechsner, P. Cholda, R. Stankiewicz, I. Papafili, and G. Stamoulis, "Interaction patterns between P2P content distribution systems and ISPs," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 222–230, May 2011.
[15] S. Ren, E. Tan, T. Luo, L. Guo, S. Chen, and X. Zhang, "TopBT: A topology-aware and infrastructure-independent BitTorrent," in *Proc. IEEE INFOCOM*, Apr. 2011.
[16] N. Leibowitz, A. Bergman, R. Ben-Shaul, and A. Shavit, "Are file swapping networks cacheable? Characterizing P2P traffic," in *Proc. Int'l Workshop on Web Content Caching and Distribution (WCW)*, Aug. 2002.
[17] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet service providers fear peer-assisted content distribution?" in *Proc. ACM Internet Measurement Conf. (IMC)*, Oct. 2005, pp. 63–76.
[18] G. Dán, "Cooperative caching and relaying strategies for peer-to-peer content delivery," in *Proc. Int'l Workshop on Peer-to-Peer Systems (IPTPS)*, 2008.
[19] J. Dai, B. Li, F. Liu, B. Li, and H. Jin, "On the efficiency of collaborative caching in ISP-aware P2P networks," in *Proc. IEEE INFOCOM*, Apr. 2011.
[20] I. Papafili, G. D. Stamoulis, F. Lehrieder, B. Kleine, and S. Oechsner, "Cache capacity allocation to overlay swarms," in *International Workshop on Self-Organizing Systems*, Feb. 2011.
[21] X. Yang and G. de Veciana, "Service capacity of peer to peer networks," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 2242–2252.
[22] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proc. ACM SIGCOMM*, Aug. 2004, pp. 367–378.
[23] F. Clévenot-Perronnin, P. Nain, and K. W. Ross, "Multiclass P2P networks: Static resource allocation for service differentiation and bandwidth diversity," *Performance Evaluation*, vol. 62, pp. 32–49, Oct. 2005.
[24] Y. Tian, D. Wu, and K. W. Ng, "Modeling, analysis and improvement for BitTorrent-like file sharing networks," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–11.
[25] I. Rimac, A. Elwalid, and S. Borst, "On server dimensioning for hybrid P2P content distribution networks," in *Proc. IEEE Int'l Conf. Peer-to-Peer Computing (P2P)*, Sep. 2008, pp. 321–330.
[26] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurement, analysis, and modeling of BitTorrent-like systems," in *Proc. ACM Internet Measurement Conf. (IMC)*, Oct. 2005, pp. 35–48.
[27] R. W. Wolff, "Poisson arrivals see time averages," *Operations Research*, vol. 30, no. 2, pp. 223–231, 1982.
[28] X. Guo and O. Hernandez-Lerma, *Continuous-time Markov Decision Processes*, ser. Springer Series in Stochastic Modelling and Applied Probability. Springer, 2009.
[29] N. Z. Shor, *Minimization Methods for Non-differentiable Functions*, ser. Springer Series in Computational Mathematics. Springer, 1985.
[30] "Protopeer," http://protopeer.epfl.ch/index.html.
[31] W. Galuba, K. Aberer, Z. Despotovic, and W. Kellerer, "ProtoPeer: A P2P toolkit bridging the gap between simulation and live deployment," in *Proc. International Conference on Simulation Tools and Techniques*, Mar. 2009.
[32] D. Bertsekas and R. Gallagher, *Data Networks*. Prentice Hall, 1987.
[33] G. Dán and N. Carlsson, "Power-law revisited: A large scale measurement study of P2P content popularity," in *Proc. Int'l Workshop on Peer-to-Peer Systems (IPTPS)*, April 2010.
[34] T. Hoßfeld, F. Lehrieder, D. Hock, S. Oechsner, Z. Despotovic, W. Kellerer, and M. Michel, "Characterization of BitTorrent swarms and their distribution in the Internet," *Computer Networks*, vol. 55, no. 5, Apr. 2011.
[35] R. L. Tweedie, "Criteria for ergodicity, exponential ergodicity and strong ergodicity of markov processes," *J. Appl. Prob.*, vol. 18, pp. 122–130, 1981.

APPENDIX

Proof of Theorem 1: Recall that the controlled processes $\mathcal{Z}_s^\pi$ are coupled through the bandwidth allocation policy π. In the following we show that $\mathcal{Z}^\pi$ fulfills four criteria.

C1: The set $\mathcal{K}_i$ of cache bandwidth allocations is compact. This follows from $0 \le \kappa_{i,s}(t) \le K_i < \infty$.

C2: For every state $z = (x,y)$ the incoming inter-ISP traffic rate $\sum_s I_{i,s}(z_s, \kappa_{i,s})$ and the transition intensities $(q^\pi_{(x_s,y_s),(x_s-e_i,y_s+e_i)})_{s\in\mathcal{S}}$ are continuous functions of $\kappa_{i,s}$. The former holds by assumption, the latter follows from (2).

C3: The average inter-ISP traffic $C_i^\pi(z)$ is finite for every policy π and initial state z. In order to show this we show that $\mathcal{Z}_s^\pi$ satisfies the Foster-Lyapunov condition for every $s \in \mathcal{S}$, then we give a bound on the inter-ISP traffic rate in every state of the system. Let us define the Lyapunov function $w(z_s) = \sum_i (x_{i,s} + y_{i,s}) + 1$. Also, let us define the sequence $(t_n)_{n\ge 0}$ of time instants, which consists of the transition epochs of the process and of the instants when $\kappa_{i,s}(t)$ changes according to the policy π. Finally, we define the generalized average drift

$$AW(z_s) = E[w(Z_s(t_{n+1})) - w(Z_s(t_n))|Z_s(t_n) = z_s]. \qquad (20)$$

Consider now the Foster-Lyapunov average drift condition [35]

$$|AW(z_s)| < \infty \quad \forall z_s, \text{ and } \; AW(z_s) < -\varepsilon \quad z_s \notin C, \qquad (21)$$

where $\varepsilon > 0$ and $C \subset \mathbb{N}_0^{|\mathcal{I}|\times 2}$ is finite. For $\lambda_s < \infty$ the uncontrolled process $\mathcal{Z}_s$ satisfies (21): $|AW(z_s)| \le 1$ due to the random-walk structure of the process, and $AW(z_s) = (\lambda_s - \theta x_s - \gamma y_s)/(-q_{z_s,z_s}) < -\varepsilon$ for x_s or y_s sufficiently big. Consider now the mean drift $AW^\pi(z_s)$ of the controlled process. Again, $|AW^\pi(z_s)| \le 1$. Furthermore we have

$$AW^\pi(z_s) \le AW(z_s)\frac{-q_{z_s,z_s}}{-q_{z_s,z_s} - K_i} < -\varepsilon\frac{-q_{z_s,z_s}}{-q_{z_s,z_s} - K_i} < 0.$$

Consequently, the controlled process $\mathcal{Z}_s^\pi$ also satisfies the Foster-Lyapunov average drift condition. Since the process is aperiodic and irreducible, the drift condition guarantees ergodicity [35]. Furthermore, for $\tilde{M} = c > 0$ it holds that $I_{i,s}(z_s, \kappa_{i,s}) \le \tilde{M} w(z_s)$. This together with the ergodicity of all $\mathcal{Z}_s^\pi$ implies that $C_i^\pi(z)$ is finite.

C4: Define $H(z) = C_i^\pi(z) - C_i^\pi(a)$, where a is an arbitrarily chosen state. Then $\sum_{z'} H(z') q^\pi_{z,z'}$ is continuous in $\kappa_{i,s}$ for every state z. This follows from the finiteness of $C_i^\pi(z)$ and from *C2*.

For a continuous-time MDP with countably infinite state space and non-negative cost the following result is known (see, e.g., Theorem 5.9 in [28]).

Lemma 1: Under C1-C4 *there exists a stationary policy* π^* *that is average cost optimal.*

Since the cost function $C_i^\pi(z)$ defined in (3) is the average cost, this concludes the proof. ∎