2010

Reducing Cross-ISP Traffic of P2P Systems: The End or The Beginning of P2P Traffic Control

Peng Yang
University of Nebraska-Lincoln, pyang@cse.unl.edu

Lisong Xu
University of Nebraska-Lincoln, xu@cse.unl.edu

Yang, Peng and Xu, Lisong, "Reducing Cross-ISP Traffic of P2P Systems: The End or The Beginning of P2P Traffic Control" (2010). *CSE Technical reports*. 135.
http://digitalcommons.unl.edu/csetechreports/135

Reducing Cross-ISP Traffic of P2P Systems: The End or The Beginning of P2P Traffic Control

Peng Yang, Lisong Xu
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115
Email: {pyang, xu}@cse.unl.edu

***Abstract*—As Peer-to-Peer (P2P) systems are widely deployed in the Internet, P2P traffic control becomes a challenge for Internet Service Providers (ISPs) and P2P system vendors. Some recent works consider the interaction between ISPs and P2P systems and propose ISP-friendly P2P traffic control mechanisms for reducing cross-ISP traffic. In this paper, we consider another fundamental problem: the interaction among multiple coexisting P2P systems. Specifically, we propose an ISP-friendly inter-overlay coordination framework (COOD) for controlling P2P traffic, which consists of three important components: network traffic optimization, overlay service differentiation, and ISP policy enforcement. Our packet-level simulation result shows that, compared to current P2P traffic control mechanisms, COOD can provide better overall performance to multiple coexisting P2P systems, achieve service differentiation among different P2P systems, and implement flexible mechanisms to effectively control cross-ISP P2P traffic.**

I. Introduction

Peer-to-peer (P2P) technology has been emerging as one of the most popular and promising inventions in the past few years. Numerous P2P systems are now running in the Internet and greatly changing the paradigm of how the Internet is used. However, such a dominating technology is facing increasing obstructions from both economic aspect and performance aspect.

It is commonly accepted that the traditional P2P technologies like Gnutella [1] and BitTorrent [2] etc. place great pressure on the networks of Internet Service Providers (ISPs) by consuming too much bandwidth and producing a large amount of cross-ISP traffic, and thus some ISPs have started to limit or even throttle P2P traffic. Such a hostile attitude held by ISPs is harmful to P2P technology in the long run. Therefore, some recent works [3], [4], [5], [6], [7] and [8] consider the interaction between ISPs and P2P systems, and propose ISP-friendly P2P traffic control mechanisms for greatly reducing cross-ISP P2P traffic.

On the other hand, another fundamental problem is that P2P traffic is not optimized especially when there are multiple coexisting P2P overlays in the Internet. In this case, different P2P overlays inefficiently compete for the Internet resources, which in turn results in the degradation of the overall performance of all P2P overlays. Therefore, in this paper, we consider not only the interaction between ISPs and P2P overlays, but also the interaction among multiple coexisting P2P overlays.

Specifically, we study the P2P traffic control in a heterogenous network with multiple coexisting P2P overlays. The network heterogeneity includes both administrative heterogeneity and resource heterogeneity. The administrative heterogeneity means the network is divided into multiple administration domains which are managed by different ISPs. The resource heterogeneity means that the network hosts may have different upload capacities and the ISP domains may be interconnected by backbone links with different capacities. In such a heterogenous network, both traditional P2P technology and recent ISP-friendly P2P traffic control may fail to offer optimal performance to multiple coexisting P2P overlays.

The failure is due to the fact that both traditional P2P technology and ISP-friendly P2P traffic control are either resource-oblivous or lack a systematical way to coordinate the traffic of multiple coexisting P2P overlays. For example, traditional P2P technology usually forms a random mesh overlay among peers. On the contrary, many ISP-friendly P2P traffic control mechanisms blindly cut off cross-ISP traffic by localizing P2P traffic within the boundary of each ISP. Their overlay structures are not built according to resource metrics such as resource usage but are mainly determined by some factors that are irrelevant to them. Even if some of them are resource-aware, the lack of multi-overlay coordination may cause performance loss when there are multiple coexisting overlays. Specifically, the following problems may happen:

(1) P2P traffic of multiple coexisting overlays is not well coordinated. Different P2P overlays may compete for the bottleneck bandwidth resource though there are other spare bandwidth resources, which can be utilized by some overlays and could potentially boost the overall performance.

(2) P2P traffic of multiple coexisting overlays is not differentiated. Different P2P systems may require different Qualities of Service (QoS). As a good example, a streaming P2P overlay may have much more rigid bandwidth requirement than a file sharing P2P overlay which only has elastic bandwidth requirement.

In this paper we propose an ISP-friendly inter-overlay coordination framework (COOD), which is able to provide better overall performance to multiple coexisting overlays in a heterogenous network and offer an efficient mechanism to control cross-ISP P2P traffic. In COOD, P2P traffic control is divided into three logically related parts: (1) network traffic

Department of Computer Science & Engineering, University of Nebraska-Lincoln,
Technical Report, TR-UNL-CSE-2010-0003

optimization; (2) overlay service differentiation and (3) ISP policy enforcement. The first two parts provide optimal overall performance to multiple coexisting P2P overlays and achieve service differentiation among them. The third part ensures that the optimized P2P traffic generated by the first two parts is ISP-friendly. Particularly, in COOD network traffic optimization, we model P2P traffic control as an optimization problem where the overall performance of multiple coexisting P2P overlays is maximized. COOD overlay service differentiation further shapes the competing traffic of multiple coexisting P2P overlays on backbone links according to weighted max-min fairness [9] so that overlays with the same priority receive fair share of bandwidth; overlays with different priorities receive bandwidth proportional to their priorities. Finally, COOD ISP policy enforcement allows ISPs to specify usage policy (i.e. maximum utilization) on cross-ISP backbone links so that their cost can be controlled under popular charging models such as the 95th-percentile charging model [5].

All three parts of COOD framework are resource aware, which requires cooperation from ISPs to obtain bandwidth usage information. In this sense, COOD framework is an interface between ISPs and P2P overlays just like P4P [5]. However, the unique contribution of COOD framework is that it focuses on providing optimal overall performance to multiple coexisting overlays, a topic not addressed in recent ISP-friendly P2P traffic control proposals. Unlike many of those proposals, COOD is performance-oriented. It has no intention to localize P2P traffic if highly localized P2P traffic is harmful to P2P performance. Instead, it always tries to optimize P2P performance while maintaining cross-ISP P2P traffic under control. COOD is also easy to deploy. It transforms the theoretical foundations (like convex optimization) into mechanisms (like peer selection) that fit the current P2P paradigm. The architecture of COOD is designed to be similar to that of P4P, so that it can be easily developed using the P4P framework which has been implemented and tested in several commercial networks.

The rest of the paper is organized as follows: Section II presents the network and overlay model and describes the problem considered in the paper; Section III describes the details of COOD framework; Section IV evaluates the performance of COOD framework using extensive packet-level simulations; Section V summarizes the related work; Section VI concludes the paper.

II. Model and Problem Description

Before we present our model and problem description, we summarize the important notation used throughout of the paper in Table I for your reference.

TABLE I: Notation used in this paper

Notation	Description
s_i	autonomous system i in a network
r_i	network point of presence (POP) i in a network
l_i	backbone link i in a network
o_k	overlay k in a network
z_j^k	peer j in overlay o_k
n_i^k	the number of peers in overlay o_k and AS s_i
t_{ij}^k	the P2P traffic of overlay o_k from POP r_i to POP r_j
T_{ij}^k	the P2P traffic of overlay o_k from AS s_i to AS s_j
t^k	the total P2P traffic of overlay o_k
$F^k(t^k)$	the utility function of overlay o_k
$I(t_{ij}^k, l)$	the indicator function to tell whether traffic t_{ij}^k goes through backbone link l
u_i^k	the total upload bandwidth offered by the peers in overlay o_k and POP r_i
U_i^k	the total upload bandwidth offered by the peers in overlay o_k and AS s_i
c_l	the bandwidth capacity of backbone link l
a_l	the available bandwidth of backbone link l
b_l	the background traffic on backbone link l
t_l	the total traffic on backbone link l
p_l	the p-metric of backbone link l
p_l'	the modified p-metric of backbone link l
p_i^k	the p-metric of overlay o_k in POP r_i or AS s_i
$p_i'^k$	the modified p-metric of overlay o_k in POP r_i or AS s_i
$C(z_j^k)$	the data structure maintained for peer z_j^k in the tracker of overlay o_k, which stores the IP addresses of its neighbors
$N(o_k, l)$	the data structure maintained for each backbone link l in the tracker of overlay o_k, which stores the connection identifiers of o_k that use l

[1]We model a physical network as a set of autonomous systems (ASes), with each AS denoted by $s_i, i = 1, 2, \ldots$. Within an AS there are multiple network points of presence (POPs) which provide hosts the access to the AS and through which the access to the whole network. We denote a POP by $r_i, i = 1, 2, \ldots$. POPs are connected by backbone links. We denote a backbone link by $l_i, i = 1, 2, \ldots$.

We denote an overlay by $o_k, k = 1, 2, \ldots$ and denote a peer in overlay o_k by $z_j^k, j = 1, 2, \ldots$. There is often a central element in an overlay called *tracker* whose purpose is to enable peers to find each other. Upon a new peer joining an overlay, the tracker of that overlay sends the new peer a list of peers so that the new peer can communicate with them. Overall, such a behavior constructs a mesh-like overlay structure with P2P traffic flowing from peers in one POP to peers in another POP.

Let t_{ij}^k be the P2P traffic from POP r_i to r_j by overlay o_k. The main topic of this paper is to control $t_{ij}^k, \forall o_k, \forall r_i, \forall r_j$ so that

- *Goal 1*: the overall performance of multiple coexisting overlays is maximized;
- *Goal 2*: the traffic of different overlays with different priorities is treated differently in terms of allocated bandwidth on backbone links;
- *Goal 3*: traffic on expensive backbone links is controlled within the budget of ISPs.

Goals 1 and 2 deal with the performance and fairness of

[1] In this paper, we use terms “AS” and “ISP” interchangeably. However, such slackness does not undermine what we propose in the paper.

the traffic control, respectively. Similar problems are studied in [10], [11], [12] and [13] etc. for multiple coexisting TCP flows using a constrained utilization maximization framework, which maximizes the overall performance, i.e. total utility, according to a certain fairness criterion (depending on the utility functions). However, applying the same method to P2P traffic control is difficult, since P2P traffic is usually an aggregation of tens of thousands of TCP (UDP) flows, no single flow can determine the performance of an entire overlay. In addition, P2P traffic is affected by the peer dynamics such as the number of peers in different POPs, peer arrivals and departures, which add another complexity to directly applying the constrained utility maximization framework. Thus in this paper, we decompose performance and fairness into two separated problems, i.e. goals 1 and 2, both of which are easy to implement in the current P2P paradigm.

For the performance problem (i.e. goal 1), our work is still based on a constrained utility maximization framework. However we map this rather theoretical framework into a peer selection mechanism, which will be shown in the subsequent section. Also we do not use this framework to handle fairness, which implies that we do not impose any particular form on the utility functions. Multiple overlays with heterogenous utility functions can be easily incorporated into this framework. Even overlays without well-defined utility functions can benefit from the derived peer selection mechanism, which is particular helpful since most real-world P2P applications do not have well-defined utility functions.

For the constrained utility maximization framework considered in this paper, We suppose that the performance of overlay o_k can be represented by a utility function $F^k(t^k)$, which is an increasing, twice-differentiable and strictly concave function of t^k, which is the total traffic generated by overlay o_k:

$$t^k = \sum_{\forall r_i} \sum_{\forall r_j} t_{ij}^k \tag{1}$$

Thus maximizing the overall performance of multiple coexisting overlay is equivalent to the following constrained utility maximization problem P.

$$\max \sum_{\forall o_k} F^k(t^k) \tag{2}$$

s.t.

$$\sum_{\forall o_k} \sum_{\forall r_i} \sum_{\forall r_j} t_{ij}^k \times I(t_{ij}^k, l) \leq c_l - b_l, \forall l \tag{3}$$

$$\sum_{\forall r_j} t_{ij}^k \leq u_i^k, \forall o_k, \forall r_i \tag{4}$$

where $I(t_{ij}^k, l)$ is an indicator function which equals 1 if traffic t_{ij}^k goes through backbone link l and equals 0 otherwise. c_l is the bandwidth capacity of backbone link l, b_l is the background traffic and u_i^k is the total upload bandwidth offered by the peers in overlay o_k and POP r_i. Inequality (3) means that P2P traffic cannot exceed the available bandwidth of backbone links; Inequality (4) indicates that P2P traffic of any overlay generated by peers in any POP cannot exceed the upload bandwidth offered by the peers in that overlay and POP. We assume that peers have enough download bandwidth so that there is no download bandwidth constraint in problem P.

For the fairness problem (i.e. goal 2), we consider how to achieve service differentiation among different P2P overlays in terms of the bandwidth allocated to each overlay on backbone links. It should be noted that by service differentiation, we do not mean $t_{ij}^k > t_{ij}^m$ if overlay o_k has a larger priority than overlay o_m, since they are just aggregated P2P traffic, which is affected by the peer dynamics. The service differentiation in this paper is implemented at the connection level. Specifically, the competing connections of two overlays with the same priority should get the same bandwidth share. While the competing connections of two overlays with different priorities should get different bandwidth share according to their priorities.

Goal 3 handles the ISP cost of P2P traffic, which is not addressed by the first two goals. The philosophy here is that while bandwidth resource should be utilized fully and efficiently to maximize performance, it should be noted that certain bandwidth resource, such as the bandwidth of inter-ISP backbone links, is used at the cost of ISP benefit [14]. Thus, it is extremely desirable to control the P2P traffic on backbone links with high costs. In this paper, we consider limiting the P2P traffic on backbone links by letting ISPs specify the usage policy of backbone links. The usage policy is specified by the maximum link utilization for a backbone link. For example, an ISP may specify the maximum link utilization for link l to be $u\%$. Thus, it is required that

$$\frac{\sum_{\forall o_k} \sum_{\forall r_i} \sum_{\forall r_j} t_{ij}^k I(t_{ij}^k, l) + b_l}{c_l} \leq u\% \tag{5}$$

By limiting the maximum link utilization, the cost of an ISP can be controlled.

III. An ISP-friendly inter-overlay coordination framework (COOD)

Like P4P, COOD requires cooperation from both ISPs and P2P service vendors. It also adopts a similar architecture as that of P4P with servers running by ISPs to collect various types of network information and feed them back to trackers. Trackers then use the information to control the traffic generated by overlays. The architecture similarity allows easy incorporation of COOD into the P4P framework.

COOD has three components: (1) COOD network traffic optimization; (2) COOD overlay service differentiation and (3) COOD ISP policy enforcement. Each component is orthogonal to other components and can be enabled or disabled individually. All three components rely on a common infrastructure, which we call COOD network view. This view can be seen as a graph G = (V, E), where V is a set of nodes and E is a set of edges. A node in V can be a router or a POP. An edge in E is a backbone link connecting two nodes in the view. The complete COOD network view is distributed among different COOD servers running by different ISPs so that each ISP can maintain the view of its own network. The COOD network view stores bandwidth usage information of backbone links that cannot be measured directly and accurately by P2P overlays. By utilizing such information, the resource-oblivous behaviors of P2P overlays can be avoided. The result is a synergic coexistence of multiple overlays and ISPs.

A. COOD network traffic optimization

COOD network traffic optimization maximizes the overall performance of multiple coexisting overlays. It is based on the constrained utility maximization problem P proposed in section II. It transforms problem P into a peer selection mechanism based on a distributed algorithm.

1) A distributed algorithm for problem P: To derive a fully distributed algorithm to solve problem P, let's consider its Lagrangian. We begin by decomposing problem P into subproblems. First we transform the objective function (2) into standard minimization form,

$$\min - \sum_{\forall o_k} F^k(t^k) \tag{6}$$

Associating Lagrangian multipliers $p_l, \forall l$ with the constraints defined in inequality (3) and associating Lagrangian multipliers $p_i^k, \forall o_k, \forall r_i$ with the constraints defined in inequality (4), we modify the objective function in (6) into

$$\begin{aligned} \min\ L(t) = \quad & - \sum_{\forall o_k} F^k(t^k) \\ & + \sum_{\forall l} p_l (\sum_{\forall o_k} \sum_{\forall r_i} \sum_{\forall r_j} t_{ij}^k \times I(t_{ij}^k, l) \\ & - c_l + b_l) \\ & + \sum_{\forall o_k} \sum_{\forall r_i} p_i^k (\sum_{\forall r_j} t_{ij}^k - u_i^k) \end{aligned} \tag{7}$$

Then we obtain the Lagrangian dual as follows:

$$\max_{p_l, p_i^k} L(t) \tag{8}$$

Here the Lagrangian multipliers, as shown later, represent bandwidth usage.

We observe that the Lagrangian dual can be decomposed into multiple subproblems, each of which can be independently solved by an overlay:

$$\begin{aligned} \max_{t_{ij}^k} f(t^k) \quad = \quad & F^k(t^k) \\ - \quad & \sum_{\forall l} p_l^* \sum_{\forall r_i} \sum_{\forall r_j} t_{ij}^k \times I(t_{ij}^k, l) \\ - \quad & \sum_{\forall r_i} (p_i^k)^* \sum_{\forall r_j} t_{ij}^k \end{aligned} \tag{9}$$

where $p_l^*, \forall l$ and $(p_i^k)^*, \forall o_k, \forall r_i$ are optimal values of the Lagrangian multipliers that solve the dual problem in (8). According to duality theory, the solution to (9) is the solution to problem P.

TABLE II: Subgradient algorithm

```
1. Choose initial Lagrangian multiplier values $p_l(0) = 0, \forall l$ and
   $p_i^k(0) = 0, \forall o_k, \forall r_i$.
2. Repeat the following iteration until convergence, start with $\mu = 0$:
   2.1. Solve the subproblem in (9) for each overlay using the incremental
        approach with $p_l(\mu), \forall l$, $p_i^k(\mu), \forall o_k, \forall r_i$ and derive
        $t_{ij}^k(\mu), \forall o_k, \forall r_i, \forall r_j$.
   2.2. Update Lagrangian multipliers according to equations (11) and (12)
        and derive $p_l(\mu+1), \forall l$ and $p_i^k(\mu+1), \forall o_k, \forall r_i$.
   2.3. $\mu = \mu + 1$
```

The above nice decomposition is easy to solve distributedly by a subgradient algorithm which involves solving the subproblem in (9) and updating the Lagrangian multipliers repeatedly until the algorithm converges.

There are several efficient algorithms capable of solving the subproblem in (9). Here we adopt one from [15] and [16]. The algorithm maximizes the objective function in (9) using an incremental approach. Beginning with $t_{ij}^k = 0, \forall o_k, \forall r_i, \forall r_j$, we find one t_{ij}^k with the largest positive marginal utility, which is defined as

$$\frac{df(t^k)}{dt_{ij}^k} = F'^k(t^k) - \sum_{\forall l} p_l \times I(t_{ij}^k, l) - p_i^k \tag{10}$$

and increase this t_{ij}^k. As $F^k(t^k)$ is a strictly concave function, $F'^k(t^k)$ decreases as we increase t_{ij}^k. We increase t_{ij}^k until its marginal utility is no longer the largest. Then we increase another t_{ij}^k with the largest marginal utility. This process repeats until all the marginal utilities become 0.

With the optimal $t_{ij}^k(\mu)$ computed by the above method at iteration μ, the Lagrangian multipliers are updated as follows

$$\begin{aligned} p_l(\mu+1) \quad = \quad & [p_l(\mu) \\ + \quad & \theta_l (\sum_{\forall o_k} \sum_{\forall r_i} \sum_{\forall r_j} t_{ij}^k(\mu) I(t_{ij}^k, l) + b_l(\mu) - c_l)]^+ \end{aligned} \tag{11}$$

$$p_i^k(\mu+1) = [p_i^k(\mu) + \theta_i^k (\sum_{\forall r_j} t_{ij}^k(\mu) - u_i^k)]^+ \tag{12}$$

where $\theta_l > 0$, $\theta_i^k > 0$ are step sizes and $[\bullet]^+$ is nonnegative orthant projection. The Lagrangian multipliers defined in (11) and (12) actually measure the bandwidth usage of backbone links and the upload bandwidth usage of hosts. For p_l, it is updated according to the differences between $\sum_{\forall o_k} \sum_{\forall r_i} \sum_{\forall r_j} t_{ij}^k(\mu) I(t_{ij}^k, l) + b_l(\mu)$ and c_l. Note that the former is just the total traffic on backbone link l at iteration μ, which is denoted by $t_l(\mu)$ in the subsequent discussion. Thus when $t_l(\mu)$ is larger than c_l, or in other words l is congested, p_l increases. Otherwise it decreases. Similarly p_i^k increases if the traffic $\sum_{\forall r_j} t_{ij}^k(\mu)$, which is originated in POP r_i, is larger than the total upload bandwidth offered by the peers in the POP, or in other words the total upload bandwidth becomes a bottleneck.

The subgradient algorithm is summarized in Table II.

2) COOD peer selection: Based on the subgradient algorithm in Table II we propose the following peer selection mechanism. In the subsequent discussion we call a peer a *requesting peer* if it is requesting its tracker for a set of peers and call the AS and the POP where it resides in a *requesting AS* and a *requesting POP*, respectively.

The idea of COOD peer selection is to mimic the behavior of the subgradient algorithm and select a set of best peers for a requesting peer, which leads to the maximum increase in the aggregated utility function in (2) (i.e. the overall performance).

The subgradient algorithm in Table II has two components: updating Lagrangian multipliers and solving the subproblem in (9). In the following discussion, we'll talk about their adaptations in COOD peer selection.

The adaptation of updating Lagrangian multipliers is straightforward. The Langrangian multipliers in (11) measure the bandwidth usage of backbone links. Thus they are maintained in COOD network view and are updated by COOD servers periodically according to equation (11). The Langrangian multipliers in (12) are about upload bandwidth usage of peers in a POP. Such information can be measured by peers, and then transmitted to its tracker in their periodical communication. Trackers then maintain and update them according to equation (12). In COOD peer selection, we call the Lagrangian multipliers performance metrics or *p-metrics* for short, since they measure bandwidth usage and have great impact on the performance of overlays.

The adaptation of solving the subproblem in (9) is more tricky. The adaptation is based on the following observation. When solving the subproblem in (9), we always increase P2P traffic t_{ij}^k with the largest marginal utility. Since $F'^k(t^k)$ is fixed for all t_{ij}^k, this is equivalent to increase t_{ij}^k with the smallest $\sum_{\forall l} p_l \times I(t_{ij}^k, l) + p_i^k$, i.e the sum of p-metrics along its path according to equation (10). By returning a set of peers to a requesting peer, we are increasing the P2P traffic from the returned peers to the requesting peer if there is data flow among them. Thus we naturally want to increase the traffic with the smallest sum of p-metrics along its path just like what the subgradient algorithm does. Actually this is the best choice that we can make, since it leads to the maximum increase in the overall performance of multiple coexisting overlays defined in (2).

Based on the above idea, when a tracker returns a set of peers to a requesting peer, it selects peers within a POP with the smallest sum of p-metrics to the requesting POP. However there are several concerns of this approach.

One concern is about the robustness of an overlay structure. The above approach may cause an overlay structure to lose a certain degree of randomness, since all peers returned to a requesting peer are from a single POP. Randomness is very important to the robustness of an overlay. Thus, we make the following modification to the above approach. Instead of finding the best POP, a tracker finds the best AS. The best AS is the one with the smallest sum of p-metrics from its gateway router to a requesting POP. Accordingly, the p-metric defined in (12) should be modified to represent the upload bandwidth usage in an AS instead of a POP:

$$p_i^k(\mu+1) = [p_i^k(\mu) + \theta_i^k(\sum_{\forall s_j} T_{ij}^k(\mu) - U_i^k)]^+ \tag{13}$$

where $\theta_i^k > 0$ is the step size, $T_{ij}^k(\mu)$ is the traffic of overlay o_k from AS s_i to AS s_j at iteration μ and U_i^k is the upload bandwidth offered by the peers in overlay o_k and AS s_i. The tracker randomly returns a portion of the returned peers in the best AS to the requesting peer and then randomly selects the rest of the peers from other ASes and returns them to the requesting peer too. Because an AS usually has much more peers to choose from than a POP and the peers to be returned are not from a single AS, the randomness of an overlay structure increases. The percentage used to select the peers from the best AS is called the *dominating percentage*. A higher dominating percentage means that more peers are selected from the best AS. However, a dominating percentage that is too high may be harmful to the robustness of an overlay, since most of the peers returned by a tracker are from a single AS. In COOD peer selection, dominating percentage is a configurable parameter that is specified by a tracker.

Another concern is about p-metrics. The p-metrics defined in (11) and (12) are only effective when they are nonzero. However, there are cases in which they are 0s. This happens when the amount of traffic is not larger than the corresponding bandwidth capacity in (11) and (12). Thus, it is possible that there are multiple ASes whose sum of p-metrics along their paths to a requesting POP is 0. From the subgradient algorithm, those ASes are equally good. We choose one from them with the smallest sum of modified p-metrics. Inspired by [14], we propose the following modified p-metrics

$$p'_l(\mu) = \frac{1}{c_l - t_l(\mu)} \tag{14}$$

where c_l is the bandwidth capacity of backbone link l and t_l is the traffic on the link and

$$p'^k_i(\mu) = \frac{1}{(U_i^k - \sum_{\forall s_j} T_{ij}^k(\mu))/n_i^k(\mu)} \tag{15}$$

where U_i^k is the total upload bandwidth offered by the peers in overlay o_k and AS s_i; $T_{ij}^k(\mu)$ is the traffic of overlay o_k from AS s_i to s_j at iteration μ and $n_i^k(\mu)$ is the number of peers in overlay o_k and AS s_i at iteration μ. The modified p-metrics measure bandwidth surplus (bandwidth capacity minus bandwidth usage), and larger bandwidth surplus corresponds to smaller modified p-metrics.

We summarize the peer selection mechanism in Table III.

B. COOD overlay service differentiation

COOD overlay service differentiation provides different qualities of service in terms of bandwidth allocated to each overlay on backbone links. This is different from the works [15], [16], and [17] etc, which focus on service differentiation in terms of upload bandwidth. The goal of COOD overlay service differentiation is to ensure that competing overlays with the same priority share bandwidth on backbone links

TABLE III: Peer selection mechanism

```
Upon a requesting peer in POP $r_i$ asking for a set of peers, the tracker
does the following:
1. Evaluate the sum of p-metrics along the path to POP $r_i$ for each
   candidate AS and choose one with the smallest sum of p-metrics.
2. If there are multiple ASes with 0 sum of p-metrics then
      Evaluate the sum of modified p-metrics for those ASes and
      choose one AS with the smallest sum of modified p-metrics.
      Label the chosen AS as the target AS.
   else
      Label the chosen AS in step 1 as the target AS
3. Among $n$ peers to be returned, choose $m \leq n$ peers randomly in the
   target AS. Choose $n - m$ peers randomly from other ASes.
4. Return the chosen peers to the requesting peer.
```

fairly; while competing overlays with different priorities share bandwidth according to their priorities.

As mentioned in Section II, COOD overlay service differentiation is implemented at the connection level. Connections of different overlays share backbone links and their bandwidth allocation is limited by the bandwidth capacity of those backbone links. Such an allocation problem can be solved by a constrained utility maximization problem which has been studied as a general resource allocation method to maximize the social welfare in [11] and [12] and has subsequently been modified in [18] and [19] to allow the (weighted) max-min fair allocation. In this paper, the bandwidth of connections in different overlays is allocated according to the weighted max-min fairness. We begin by first presenting the max-min fair bandwidth allocation and extend it to the weighted max-min fair allcation. Simply put, the max-min fairness is that no user can increase its allocation without decreasing the already smaller or equal allocation of another user. Our allocation mechanism is based on the water filling algorithm. Due to the particular characteristic of the max-min fairness, such a mechanism is efficient and is able to avoid transient behaviors of the dynamic system proposed in [19] before convergence. We call this resource sharing mechanism *static demand adaptation* (as opposed to the dynamic system) or *SDA* for short.

SDA uses the water filling algorithm to solve the bandwidth allocation problem. It has been proven in [20] that the water filling algorithm when applied to network bandwidth allocation always yields the max-min fair allocation. In COOD overlay service differentiation, bandwidth allocation is made by COOD servers and allocation results (i.e. the bandwidth allocated to a conection) are pushed to trackers which in turn push the allocation results to peers. To allocate the bandwidth of a backbone link, the information a COOD server needs to know is the number of connections that are bottlenecked by this backbone link. By saying that a connection is bottlenecked by a backbone link, we mean that the p-metric of this link is the maximum among the p-metrics on the path of the connection. If there are more than one link on the path with the largest p-metric, any one of them can be identified as the bottleneck link for the connection. Thus, for any connection, it is bottlenecked by one and only one link. Any two connections bottlenecked by the same bottleneck link get the same bandwidth allocation. Thus, it is sufficient for a COOD server to know the number of connections bottlenecked by a backbone link and allocate equal share of bandwidth to each connection (we'll extend this idea for the weighted max-min fairness and the service differentiation later).

From the above description, SDA relies on the number of connections traversing a particular backbone link. Such information is maintained by each tracker and submitted to COOD servers. Specifically, a tracker maintains two sets of data structures, one for each peer, the other for each backbone link. The data structure for peer z_j^k is a set of IPs, which is denoted by $C(z_j^k)$, to which the peer has connections. The data structure for backbone link l, which is denoted by $N(o_k, l), \forall o_k$, is a list of triples $< o_k, ip_1, ip_2 >$ that uniquely identifies a connection between two peers with IP addresses ip_1 and ip_2 in overlay o_k that uses backbone link l. Both data structures are updated periodically through the communication between a tracker and its peers and upon a peer leaving an overlay. From data structure $N(o_k, l), \forall o_k$, a COOD server can easily figure out how many connections use backbone link l by counting the number of triples in $N(o_k, l), \forall o_k$. Table IV presents the details of how to maintain these two data structures.

The idea of SDA is described in Table V. The bandwidth allocation of SDA can be described as a process that fills up bottleneck links one by one. In the first round, SDA chooses a backbone, say l_1, with the largest p-metric in COOD network view. If the p-metric equals 0, SDA terminates, since no backbone link is a bottleneck for any connection. Otherwise let

$$N(l_1) = \bigcup_{\forall o_k} N(o_k, l_1) \tag{16}$$

where $N(l_1)$ contains all the triples from $N(o_k, l_1), \forall o_k$. For any triple $< o_k, ip_1, ip_2 > \in N(l_1)$, allocate to the corresponding connection an equal share of the bandwidth of link l_1, which is

$$\frac{a_{l_1}}{|N(l_1)|} \tag{17}$$

where a_{l_1} is the available bandwidth of link l_1 and $|N(l_1)|$ is the cardinality of set $N(l_1)$. Thus in the first round, the bandwidth allocation of all connections through the link with the largest p-metric is determined. Similarly in the second round, the bandwidth of the link, say l_2, with the second largest p-metric is allocated to any connection through it. Note that a connection which is bottlenecked by link l_1 may also go through link l_2. So the bandwidth of those connections bottlenecked by link l_1 and through link l_2 should be subtracted from the available bandwidth of link l_2. The remaining bandwidth is equally allocated to any connection that is bottlenecked by link l_2. Similar action also applies to round three, round four etc., provided that in each round the bandwidth of the connections bottlenecked in previous rounds

TABLE IV: SDA information collection

```
$C(z_j^k)$:        the current set of IP addresses to which peer $z_j^k$
                   has connections
$C'(z_j^k)$:       the set of IP address to which $z_j^k$
                   has connections (maintained in the tracker before update)
$L(r_n, r_m)$:     the set of backbone links on the path from
                   POP $r_n$ to $r_m$.
$N(o_k, l)$:       the set of triples $< o_k, ip_1, ip_2 >$ for backbone link $l$

upon receiving the information from peer $z_j^k$:

$ip_1$ = the IP address of peer $z_j^k$
Map $ip_1$ to its POP $r_m$;
For each IP address $ip_2$ in $C(z_j^k)$
  Map $ip_2$ to its POP $r_n$;
  Obtain $L(r_n, r_m)$ from COOD network view;
  For every link $l$ in $L(r_n, r_m)$
    If $< o_k, ip_1, ip_2 > \notin N(o_k, l)$
      Add $< o_k, ip_1, ip_2 >$ into $N(o_k, l)$;
    End
  End
End
For each IP address $ip_2$ in $C'(z_j^k) - C(z_j^k)$
  Map $ip_2$ to its POP $r_n$;
  Obtain $L(r_n, r_m)$ from COOD network view;
  For every link $l$ in $L(r_n, r_m)$
    Delete $< o_k, ip_1, ip_2 >$ from $N(o_k, l)$;
  End
End

upon peer $z_j^k$ with IP address $ip_1$ leaving the overlay:

Delete $< o_k, ip_1, any >$ from $N(o_k, l), \forall l$
```

TABLE V: SDA bandwidth allocation

```
$L$: a set of backbone links in COOD network view
   with nonzero p-metrics
$a_l$: the available bandwidth of backbone link $l$
$b(< o_k, ip_1, ip_2 >)$: bandwidth allocation to connection
                          $< o_k, ip_1, ip_2 >$
$D$: the set of connections which are already bottlenecked;
   (initialized to empty)
$N(l)$: the set of triples $< o_k, ip_1, ip_2 >$ maintained for link $l$,
   which is the union of $N(o_k, l), \forall o_k$

If $L$ is emtpy
  Terminate;
End
Sort the backbone links in $L$ in decreasing order in terms of their p-metrics;
For each backbone link $l \in L$ in decreasing order
  $bottlenecked = 0$;
  For each triple $< o_k, ip_1, ip_2 > \in N(l)$
    If $< o_k, ip_1, ip_2 > \in D$
      $a_l = a_l - b(< o_k, ip_1, ip_2 >)$;
      $bottlenecked = bottlenecked + 1$;
    End
  End
  For each triple $< o_k, ip_1, ip_2 > \in N(l)$
    If $< o_k, ip_1, ip_2 > \notin D$
      $b(< o_k, ip_1, ip_2 >) = a_l / (|N(l)| - bottlenecked)$;
      Add $< o_k, ip_1, ip_2 >$ in $D$;
    End
  End
End
```

is subtracted from the available bandwidth of the link in this round if they use the link.

In the above description of SDA, we allocate the same share of bandwidth to the connections bottlenecked by the same backbone link. Now we extend it to the weighted max-min fair allocation [9] to allow service differentiation, where each overlay is associated with a weight. We denote the weight of overlay o_k by w_k. In SDA, the weighted max-min fairness can be achieved by allocating bandwidth in the following way. Suppose for backbone link l, the set of connections bottlenecked by it is denoted by D_l. SDA allocates bandwidth in a way such that

$$\frac{c_i}{w_i} = \frac{c_j}{w_j}, \forall c_i, c_j \in D_l \tag{18}$$

where c_i, c_j are two connections in D_l (which are abused to denote their bandwidth allocations) and w_i, w_j are the weights of their corresponding overlays, respectively. In other words, for any two connections which are bottlenecked by the same backbone link, the ratio between the allocation and the weight should be the same. The condition in (18) can be satisfied by dividing the remain bandwidth a_l of link l into slices. The bandwidth s_l of each slice can be expressed by

$$s_l = \frac{a_l}{\sum_{c_i \in D_l} w_i} \tag{19}$$

Then the bandwidth allocated to connection c_i with weight w_i can be simply determined by

$$c_i = w_i \times s_l \tag{20}$$

C. COOD ISP policy enforcement

ISP policy is incorporated in COOD by the notion of virtual capacity. An ISP maintains the traffic on a backbone within the virtual capacity. Virtual capacity is specified by a target utilization value $u_l\%$ for backbone link l. Suppose the capacity of l is c_l, its virtual capacity is $u_l\% \times c_l$. Based on the type and the bandwidth usage of a backbone link, an ISP can adjust its virtual capacity flexibly. For example, if a backbone link connects to a transit network running by another ISP, an ISP may set the virtual capacity to a lower value so that the cost of the traffic on the link can be controlled under popular charging models such as 95th-percentile charging model.

Virtual bandwidth capacity is used at two places in COOD framework. Firstly it is used to compute the p-metric and the modified p-metric for a backbone link, in which real capacity is replaced by virtual capacity. In this way, a backbone link with lower virtual capacity tends to generate a larger p-metric or modified p-metric and is considered more congested by COOD network traffic optimization. As a result, less P2P traffic is routed through this link by COOD network traffic optimization. Virtual capacity is also used in SDA. Suppose the non-P2P traffic accounts for $n\%$ of the total bandwidth c_l. SDA can directly allocate $(u\% - n\%) \times c_l$ bandwidth among all competing overlays. Thus, the total traffic on link l is at most $n\% \times c(t) + (u - n)\% \times c_l = u\% \times c_l$, and the ISP policy on this link is enforced.

IV. Evaluation

We evaluate COOD using NS2 simulator and the BitTorrent simulation package developed in [21] at the packet level. Unlike numerical and flow-level simulations, our simulation package runs fully at the packet-level, in which packets are sent by two-way TCP (Agent/TCP/FullTcp/Newreno) connections and go through complete protocol stack modeled by NS2. In this way, our simulations are able to capture most complexities involved in real-world networks.

We compare COOD framework with two other technologies: traditional P2P technology and ISP-friendly P2P technology. In traditional P2P technology, an overlay is constructed randomly. A peer communicates with other peers selected by the tracker randomly from all available peers. In ISP-friendly P2P technology, a tracker selects most of peers in the requesting AS of a requesting peer and randomly selects the rest of peers in other ASes. It then returns the selected peers to the requesting peer so that they can communicate with each other. The percentage of the peers selected from the requesting AS is called the *locality percentage*. A larger locality percentage means more P2P traffic is localized within the boundary of each AS.

We divide our simulation into 3 sets for the three components of COOD framework. The following simulation parameters and assumptions are common for all 3 sets of simulations:

(1) There are 3 types of upload bandwidth which a host may offer: 0.3Mbps, 1.0Mbps and 5Mbps. The upload bandwidth distribution of an AS is denoted by a vector $[\phi_1, \phi_2, \phi_3]$, where $\phi_i, i = 1, 2, 3$ is the percentage of hosts having type i upload bandwidth. For example $[0.3, 0.3, 0.4]$ means within an AS, 30% hosts have upload bandwidth of 0.3Mbps, 30% hosts have upload bandwidth of 1.0Mbps, and 40% hosts have upload bandwidth of 5Mbps.

(2) The network topology used throughout all simulations is shown in Figure 1 which also shows the bandwidth capacity of the inter-AS backbone links and the upload bandwidth distribution of all the ASes (shown beside the corresponding ASes). We assume that the bandwidth capacities of the backbone links within each AS are sufficiently large and their impact can be ignored. Thus we construct a star topology for each AS with its hosts directly connecting to the router of the AS. An inter-AS backbone link between two ASes is connecting to the routers of those two ASes. Also, In order to run simulations in a relatively short time, we set the bandwidth capacities of backbone links to relatively small values of 10Mbps, 50Mbps, and 100Mbps. Note that increasing the bandwidth capacities of backbone links changes the absolute simulation results, but not the relative performance of different P2P technologies.

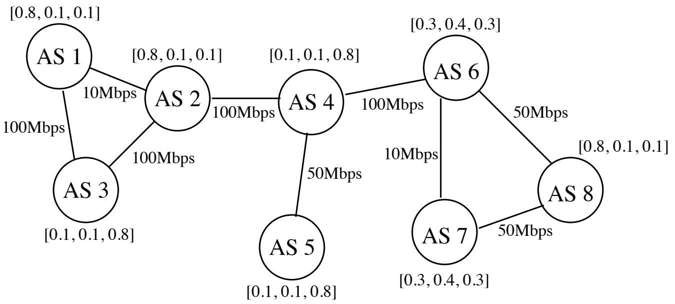

Fig. 1: Network topology

(3) For an overlay, its initial seed distribution is a vector of numbers representing the number of initial seeds in different ASes. For example, initial seed distribution [1, 1, 2] means initially there is 1 seed in AS 1, 1 seed in AS 2 and 2 seeds in AS 3. Besides the initial seed distribution, an overlay also has a peer distribution, which is a vector of numbers indicating the number of peers in different ASes. For example, peer distribution [100,100,100] means there are 100 peers in AS 1, 100 peers in AS 2 and 100 peers in AS 3. Unlike initial seeds, peers join overlays during a certain time interval.

(4) Each overlay has a 20MB file to share. Each file is divided into chunks of 256 bytes.

(5) Both locality percentage in ISP-friendly P2P technology and dominating percentage in COOD network traffic optimization are set to 0.7.

A. COOD network traffic optimization

In this set of simulations we evaluate COOD network traffic optimization. We suppose that two overlays are running in the network shown in Figure 1. The initial seeds distribution for both overlays is $[2, 2, 2, 2, 2, 2, 2, 2]$, i.e. there are 2 seeds in each AS initially. The peer distribution for both overlays is $[48, 48, 48, 23, 23, 13, 23, 8]$. Peers join the overlays during a time interval of 100 seconds.

The overall average downloading time for each overlay is shown in Figure 2. As can be seen that COOD network traffic optimization produces much better performance than traditional and ISP-friendly P2P technologies. Note that COOD not only reduces the overall downloading time of both overlays, but also reduces the individual downloading time of each overlay.

Figure 3 shows the average upload bandwidth utilization in each AS, from which we can see that COOD network traffic optimization makes better use of the upload bandwidth offered by AS 3 and AS 4. These 2 ASes contain hosts with superior upload bandwidth, which boosts the performance of COOD network traffic optimization (especially for the peers in AS 1 and AS 2, since they establish a large number of connections to the peers in AS 3 and AS 4 to take advantage of the

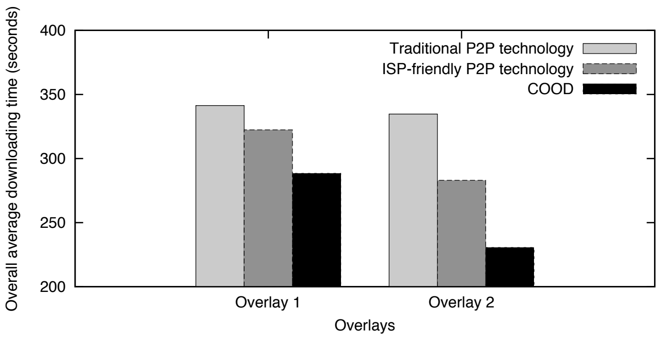

Fig. 2: COOD produces shorter overall average downloading times for both overlay 1 and overlay 2 than traditional and ISP-friendly P2P technologies.

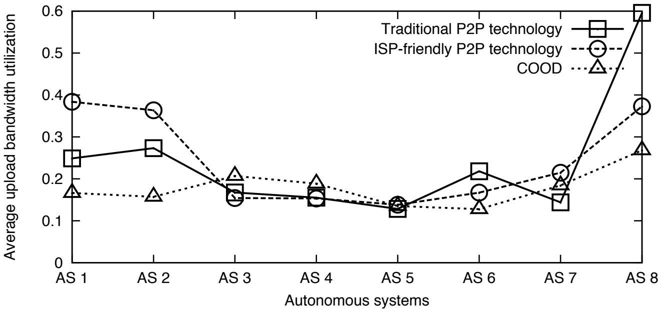

Fig. 3: COOD makes better use of the superior upload bandwidth in AS 3 and AS 4 than traditional and ISP-friendly P2P technologies.

superior upload bandwidth). Although AS 5 also contains hosts with superior upload bandwidth, COOD traffic optimization produces roughly the same utilization as that of traditional and ISP-friendly P2P technologies. The reason is that all other ASes except AS 4 are at least 2 hops away from AS 5. Given the fact that an AS is evaluated by the sum of p-metrics (or modified p-metrics) to a requesting POP, the traffic from AS 5 to other ASes (except AS 2) is deprecated because it has a larger AS-hops and tends to have a larger sum of p-metrics.

The performance boost of COOD is at the cost of higher inter-AS P2P traffic as shown in Figure 4 (we disable COOD ISP policy enforcement in this set of simulations so that the P2P traffic is not limited on the Inter-AS backbone links). The figure shows the 95th-percentile of the traffic on the inter-AS backbone links. The label "$i - j$" on x axis means the traffic on the inter-AS backbone link from AS i to AS j. It can be seen that ISP-friendly P2P technology achieves the minimum inter-AS P2P traffic; COOD network traffic optimization and traditional P2P technology produce comparable inter-AS P2P traffic.

However, the P2P traffic produced by COOD has shorter AS-hops (P2P traffic localized within an AS has 0 AS-hop, P2P traffic between two directly connected ASes has 1 AS-hops and so on ...) than that of traditional P2P technology as shown in Figure 5. In this figure we normalize the traffic with

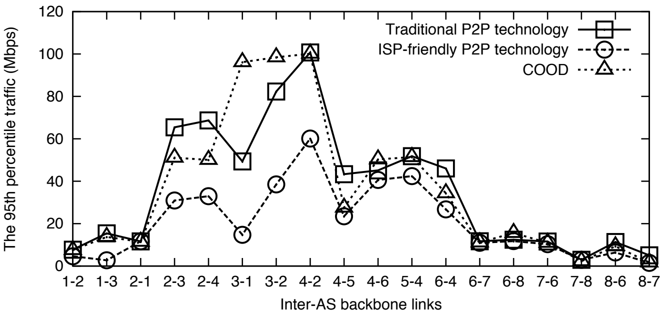

Fig. 4: COOD produces comparable inter-AS P2P traffic as that of traditional P2P technology. However, the inter-AS P2P traffic produced by COOD has shorter AS hops as shown in Figure 5.

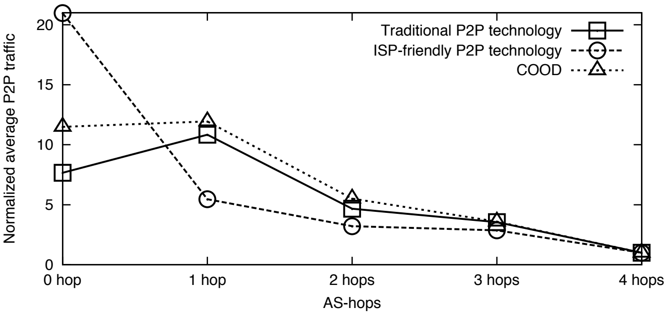

Fig. 5: The inter-AS P2P traffic produced by COOD has larger concentration at 0 and 1 AS-hop than that of traditional P2P technology. ISP-friendly technology has the largest concentration at 0 AS-hop, which causes performance loss of P2P overlays.

4 AS-hops (the largest AS-hops in Figure 1) to 1.0 in order to show the concentration of the traffic with different AS-hops. It can be seen that COOD has larger concentration at 0 AS-hop and 1 AS-hop than that of traditional P2P technology which spreads the inter-AS P2P traffic over a wide range of AS-hops. P2P traffic with short AS-hops is less expensive, since it traverses fewer ISPs. Also, it usually has shorter delay, which is important to those delay-sensitive P2P applications. The figure also shows the tradeoff between localizing P2P traffic and improving P2P performance. ISP-friendly P2P technology has the largest concentration at 0 AS-hop; however such highly localized traffic hinders it from utilizing the superior upload bandwidth offered by the peers in AS 3 and AS 4, which causes performance loss of P2P overlays. More simulation results about controlling the cross-ISP P2P traffic are shown in Section IV-C.

B. *COOD overlay service differentiation*

In this set of simulations, we evaluate COOD overlay service differentiation for two different scenarios.

In the first simulation scenario, the network topology is the same as that of Section IV-A except that the bandwidth

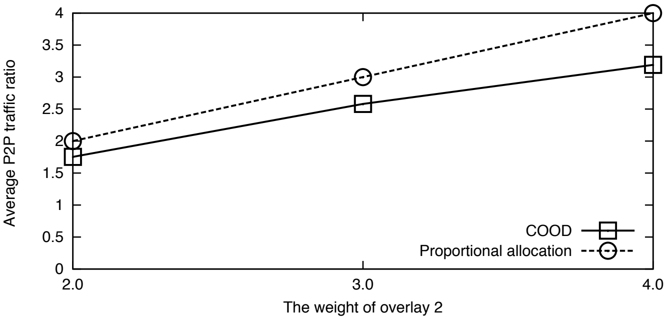

Fig. 6: COOD achieves service differentiation for 2 overlays with different weights. The average P2P traffic of each overlay on the backbone link from AS 4 to AS 2 is roughly proportional to their weights.

capacity of the inter-AS backbone link between AS 4 and AS 2 is modified to 10Mbps to represent a bottleneck link. There are 2 overlays with the same initial seed distribution and peer distribution, which are [0, 0, 0, 5, 0, 0, 0, 0] and [0, 50, 0, 0, 0, 0, 0, 0], respectively. All initial seeds have an upload capacity of 5Mbps. Peers join the overlays during a time interval of 100 seconds.

In this scenario, the peers and the initial seeds are distributed in AS 2 and AS 4, respectively. The initial seeds are in AS 4 and serve as content servers. The peers in AS 2 have to download new data from the servers through inter-AS backbone link from AS 4 to AS 2. Overlay 2 has a higher priority than overlay 1. We set the weight of overlay 1 to 1.0 and increase the weight of overlay 2 from 2.0 to 3.0 and 4.0. Since both overlays have the same initial seed distribution and peer distribution, according to the weighted max-min fairness, the bandwidth of the link from AS 4 to AS 2 allocated to an overlay should be roughly proportional to its weight.

Figure 6 shows the ratios between the average P2P traffic of overlay 2 and that of overlay 1 on the inter-AS backbone link from AS 4 to AS 2. The labels “2.0”, “3.0” and “4.0” on x axis correspond to the cases when the weight of overlay 2 is set to 2.0, 3.0 and 4.0, respectively. The imaginary reference corresponding to the proportional allocation is also shown in the figure. As can be seen from the figure, although the exact proportional allocation is not achieved, the ratio approximately increases as the weight of overlay 2 increases.

In the second simulation scenario, we consider a more complicated simulation scenario where we modify the bandwidth capacity of the inter-AS backbone link between AS 2 and AS 4 to 30Mbps. There are 4 overlays distributing in ASes 2, 4, 6 and 7. The initial seed distribution and peer distribution of each overlay is shown in Table VI. All initial seeds have an upload capacity of 5Mbps. Peers join the overlays during a time interval of 100 seconds.

In this scenario, all initial seeds are distributed in ASes 4 and 7; peers in ASes 2 and 6 have to download new data from the initial seeds. The bottleneck links are from AS 7 to AS 6 and from AS 4 to AS 2. The bandwidth allocation of those bottleneck links are determined by the weights of the four overlays which are 1.0, 2.0, 1.0 and 2.0, respectively.

TABLE VI: Initial seed and peer distributions

Overlay	Initial seed distribution	Peer distribution
1	[0, 0, 0, 0, 0, 0, 5, 0]	[0, 0, 0, 0, 0, 50, 0, 0]
2	[0, 0, 0, 0, 0, 0, 5, 0]	[0, 50, 0, 0, 0, 0, 0, 0]
3	[0, 0, 0, 5, 0, 0, 0, 0]	[0, 50, 0, 0, 0, 0, 0, 0]
4	[0, 0, 0, 5, 0, 0, 0, 0]	[0, 50, 0, 0, 0, 0, 0, 0]

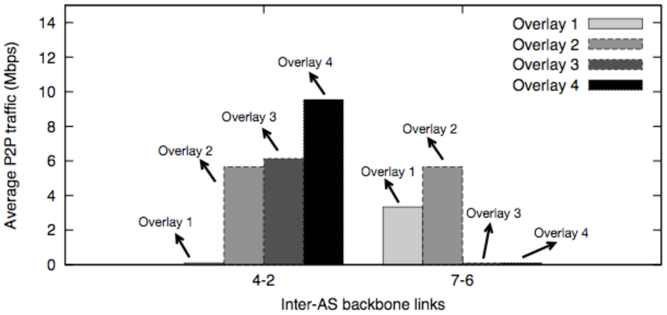

Fig. 7: COOD achieves service differentiation for 4 overlays under multiple bottleneck links according to the water-filling algorithm.

Figure 7 shows the average P2P traffic of each P2P overlay on the bottleneck inter-AS backbone links. It shows that the competing P2P traffics get the bandwidth allocation according to their priorities. In this scenario there are 2 bottleneck backbone links. The bottleneck link from AS 7 to AS 6 has the largest p-metric and is considered in the first round of SDA algorithm. The competing P2P traffics of overlay 1 and overlay 2, which are bottlenecked by this link, get the bandwidth share according to their priorities as shown in right part of Figure 7. The bottleneck link from AS 4 to AS 2 is considered in the second round of SDA algorithm and its bandwidth is allocated to the competing P2P traffics of overlay 3 and overlay 4 that are bottlenecked by it. Although the traffic of overlay 2 also traverses the link from AS 4 to AS 2, it does not compete for the bandwidth with the traffics of overlay 3 and overlay 4. Its bandwidth allocation is bottlenecked at the link from AS 7 to AS 6 and should be subtracted from the the available bandwidth of the link from AS 4 to AS 2. The remaining bandwidth is allocated to the P2P traffics of overlay 3 and overlay 4 according to their priorities as shown in the left part of Figure 7.

C. COOD ISP policy enforcement

In this set of simulations, we evaluate the ability of COOD to enforce ISP policy on the utilization of backbone links.

We consider the first simulation scenario used in Section IV-B again but with ISP policy set for expensive inter-ISP backbone links. In that scenario, peers in AS 2 are downloading data from the initial seeds in AS 4. This time, we set the weights of both overlays to 1.0. We further suppose that the network has 2 ISPs. AS 1,2 and 3 belong to ISP 1; All the other

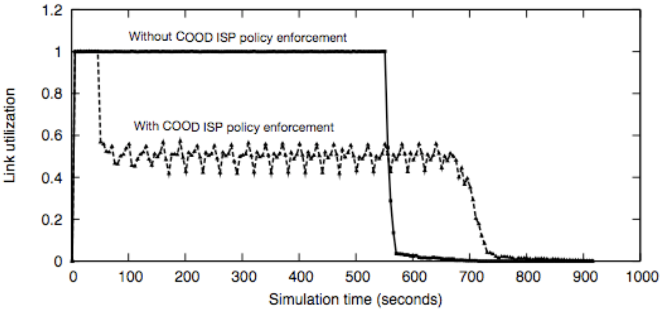

Fig. 8: COOD enforces the ISP policy set for the inter-AS backbone link from AS 4 to AS 2.

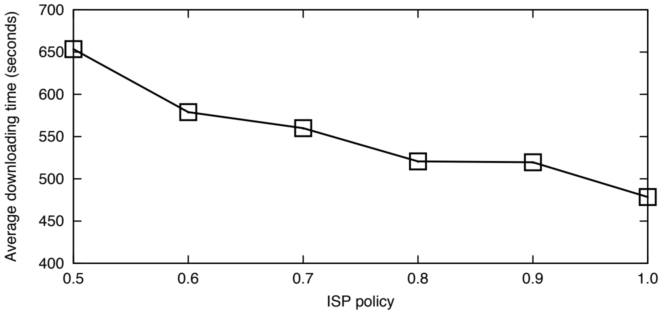

Fig. 9: Tradeoff between the ISP usage policy and the average downloading time. A tight ISP usage policy may punish the performance of P2P overlays.

ASes belong to ISP 2. ISP 2 has to pay for the traffic forwarded to ISP 1 and would like to limit the maximum utilization of the inter-AS backbone link from AS 4 to AS 2. The administrator of ISP 2 sets the maximum utilization of that backbone link to 0.5 through COOD ISP policy enforcement. We assume that no other limitations are set for any other inter-AS backbone links.

Figure 8 shows the utilization of the inter-AS backbone link from AS 4 to AS 2 throughout the simulation. It also shows the utilization of the link without COOD ISP policy enforcement for comparison. As can be seen, COOD ISP policy enforcement successfully maintains the utilization of the inter-AS backbone link around 0.5 throughout of the simulation. The average traffic on the backbone link when using COOD ISP policy enforcement is 4118321 bps; while it is 7812750 bps without COOD ISP policy enforcement. The reduction of the traffic on the inter-AS backbone link means ISP 2 will pay less for the traffic forwarded to ISP 1.

Next, we consider the tradeoff between ISP usage policy and the average downloading time in the above simulation scenario. We set the usage policy of the inter-AS backbone link from AS 4 to AS 2 to 0.5 initially and gradually release it from 0.5 to 1.0. As shown in Figure 9, when a tight usage policy is set, the average downloading time of the peers in AS 2 is long. As the policy is gradually released, the average downloading time is improved. In this scenario, the peers in AS 2 have to download new data from AS 4, thus the cross-ISP P2P traffic is crucial to the performance of the peers. Such a case is not uncommon. Often peers in an ISP rely heavily on cross-ISP P2P traffic from other ISPs either because they have to download new data from the peers in other ISPs or because the peers in the same ISP can not offer enough upload bandwidth to maintain certain quality of service. Thus, ISP policy on those cross-ISP backbone links should be set cautiously to avoid punishing P2P performance much. The guideline is that ISP policy is set so that the cost of an ISP is controlled under a limit. Within that limit, an ISP should allow as much cross-ISP P2P traffic as possible.

V. Related Works

To date few literatures study optimization P2P traffic of multiple coexisting overlays in a heterogeneous network. The problem of multiple coexisting overlay is that the interaction of different selfish overlays may cause them to step on each other and result in degradation of overall performance. Such selfish interaction is studied in [22] and [23] etc. In [22], multiple overlay routing is studied where each overlay selfishly minimizes its weighted average delay independently. The performance of such routing is examined and the interaction among multiple overlays is studied using game theory. It is showed that the interaction may result in sub-optimal result and hurt the overall performance. [23] studies selfish overlay routing. The selfishness of multiple coexisting overlays could cause race condition and result in oscillations in both route selection and network load. Our work is considerably different. In our proposal, the selfishness of each overlay is avoided by a framework, in which different overlays are well coordinated and share the bandwidth efficiently.

The interaction between ISPs and P2P overlays and the impact of P2P traffic on ISPs give rise to significant research efforts recently. Most of those works suggests that there are tussles between the two parties. [24] studies the interaction in a game theoretical framework in which P2P overlay and ISP (traffic engineering) act independently to realize their own objectives. The study shows that at equilibrium the mis-alignment between ISP objectives and P2P overlay objectives may increase the costs of ISPs or the costs of P2P overlays. The economical impact of P2P traffic is examined in [4] and [5] etc. Those works point out that random matching among peers, which is typical of current P2P paradigm, generates too much inter-ISP P2P traffic, which increases the costs for ISPs. Several ISP-friendly P2P traffic control mechanisms have been proposed recently to attack this problem. Most of them reduce cross-ISP P2P traffic by making peers communicate with other peers in its local domain. Notably the Ono project in [4] reduces cross-ISP P2P traffic by utilizing content distribution distribution networks to discover peers in the same local domain. The merit of the Ono project is that it does not require any involvement from ISPs or any kind of infrastructure. On the contrary, P4P in [5] takes another approach. Instead of avoiding any involvement from ISPs, it embraces the cooperation from ISPs. The P2P traffic is controlled according to the information fetched from ISPs. Since the fetched information is more accurate and relevant than any inferred information by overlays themselves, cross-ISP P2P traffic can be more effectively reduced. Compared with [24], our work adopts a cooperation model between ISPs and P2P overlays, thus the misalignment between these two parties is avoid. As to various ISP-friendly P2P traffic control mechanisms, the difference is that our proposal does not seek to localize P2P traffic. Instead, it gives ISPs the ability to specify usage policy of inter-ISP links and controls P2P traffic so that the policy is enforced.

P2P bandwidth allocation has been studied in [15], [16], [25] and [17] etc. Most of those works focus on upload bandwidth allocation. In [15], the problem is studied using a constrained utilization maximization framework. The purpose of bandwidth allocation, like our proposal, is to enable service differentiation among P2P overlays with different priorities. In [16] and [25] the same problem is visited again using auction games where upload bandwidth is auctioned by upstream peers to maximize their revenue, while downstream peers submit their bids to minimize their costs. [17] improves the allocation efficiency by utilizing the divide-and-conquer strategy, where upload bandwidth allocation is conducted for a set of peers instead of for every single peer. Such strategy greatly improves the scalability of upload bandwidth allocation. Compared with these works, we mainly consider bandwidth allocation on backbone links. Thus, those works and our work are complementary to each other. Also our bandwidth allocation model is not based on the constrained utilization maximization framework or the auction games but on an efficient water-filling algorithm capable of achieving the weighted max-min fairness.

VI. Conclusion

In this paper we proposed a P2P traffic control framework called COOD. COOD adopts a cooperation model between ISPs and P2P overlays. By obtaining network topological information and link usage information from ISPs, COOD provides P2P overlays with an information warehouse. This warehouse is used by P2P overlays to optimize their traffic and is also used by COOD itself to push bandwidth allocation results to P2P overlays. Using COOD network traffic optimization, the overall performance of multiple coexisting overlays is maximized. The presence of COOD overlay service differentiation ensures a fair bandwidth allocation to competing P2P overlays and allows service differentiation among P2P overlays with different priorities. COOD ISP policy enforcement gives ISPs the ability to control expensive cross-ISP P2P traffic. In sum, with COOD framework more effective cooperative traffic control can be achieved among P2P overlays and between P2P overlays and ISPs.

References

[1] "The gnutella protocol specification v0.4." [Online]. Available: www9.limewire.com/developer/gnutella_protocol_0.4.pdf

[2] "BitTorrent protocol specification v1.0." [Online]. Available: http://wiki.theory.org/BitTorrentSpecification

[3] 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 *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, Washington, DC, USA, 2006, pp. 66–66.

[4] D. R. Choffnes and F. E. Bustamante, "Taming the Torrent: a practical approach to reducing cross-ISP traffic in peer-to-peer systems," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 363–374, October 2008.

[5] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider portal for applications," in *Proceedings of ACM SIGCOMM*, Seattle, WA, August 2008.

[6] B. Ruan, W. Xiong, H. Chen, and D. Ye, "Improving locality of BitTorrent with ISP cooperation," in *2009 International Conference on Electronic Computer Technology*, Macau, February 2009, pp. 443–447.

[7] J. Wang, C. Huang, and J. Li, "On ISP-friendly rate allocation for peer-assisted VoD," in *Proceeding of the 16th ACM international conference on Multimedia*, Vancouver, British Columbia, Canada, October 2008, pp. 279–288.

[8] L. Sheng and H. Wen, "Nearby neighbor selection in P2P systems to localize traffic," in *Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services*, Venice/Mestre, Italy, May 2009, pp. 68–73.

[9] B. Radunovic and J.-Y. Le Boudec, "A unified framework for max-min and min-max fairness with applications," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1073–1083, October 2007.

[10] S. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 525–536, Aug. 2003.

[11] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[12] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[13] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 1063–6692, 2000.

[14] V. Reddy, Y. Kim, S. Shakkottai, and A. Reddy, "Designing ISP-friendly peer-to-peer networks designing ISP-friendly peer-to-peer networks using game-based control," in *The 1st Workshop on Internet Economics (WIE'09)*, September 2009.

[15] C. Wu and B. Li, "Diverse: application-layer service differentiation in peer-to-peer communications," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 222–234, Jan. 2007.

[16] ——, "Strategies of conflict in coexisting streaming overlays," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, May 2007.

[17] M. Wang, L. Xu, and B. Ramamurthy, "A flexible divide-and-conquer protocol for multi-view peer-to-peer live streaming," in *Proceedings of IEEE P2P*, Seattle, WA, Septmeber 2009.

[18] B. Wydrowski and M. Zukerman, "Maxnet: A congestion control architecture for maxmin fairness," *IEEE Communications Letters*, vol. 6, no. 11, pp. 512–514, Nov. 2002.

[19] B. Wydrowski, L. L. Andrew, and M. Zukerman, "Maxnet: A congestion control architecture for scalable networks," *IEEE Communications Letters*, vol. 7, no. 10, pp. 511 –513, October 2003.

[20] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[21] K. Eger, T. Hoßfeld, A. Binzenhöfer, and G. Kunzmann, "Efficient simulation of large-scale P2P networks: packet-level vs. flow-level simulations," in *2nd Workshop on the Use of P2P, GRID and Agents for the Development of Content Networks (UPGRADE-CN'07)*, Monterey Bay, USA, June 2007, pp. 9–16.

[22] W. Jiang, D.-M. Chiu, and J. C. S. Lui, "On the interaction of multiple overlay routing," *Performance Evaluation*, vol. 62, no. 1-4, pp. 229–246, 2005.

[23] R. Keralapura, C.-N. Chuah, N. Taft, and G. Iannaccone, "Can coexisting overlays inadvertently step on each other," in *Proceedings of IEEE ICNP*, Boston, MA, November 2005.

[24] Y. Liu, H. Zhang, W. Gong, and D. Towsley, "On the interaction between overlay routing and underlay routing," in *Proceedings of IEEE INFOCOM*, vol. 4, Miami, FL, March 2005, pp. 2543– 2553.

[25] C. Wu, B. Li, and Z. Li, "Dynamic bandwidth auctions in multi-overlay P2P streaming with network coding," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 6, pp. 806–820, June 2008.