Socially-optimal ISP-aware P2P Content Distribution via a Primal-Dual Approach

Jian Zhao, Chuan Wu
The University of Hong Kong
{jzhao,cwu}@cs.hku.hk

***Abstract*—Peer-to-peer (P2P) technology is popularly exploited to enable large-scale content distribution (*e.g.*, live and on-demand video streaming) with low server costs. Most P2P protocols in place are network agnostic, where peers download content chunks from each other regardless of their ISP belonging, leading to significant amounts of inter-ISP traffic. It has been a daunting challenge how to design protocols that optimize the P2P content distribution topology, such that inter-ISP traffic is minimized while the dissemination performance is maximized, not to mention one that motivates peer's voluntary ISP-aware peer selection. In this paper, we formulate a social welfare maximization framework for dynamical construction of the P2P content distribution topology taking into consideration both peers' gain due to chunk downloading and inter-ISP traffic that is incurred. Given its nature of an assignment problem, we resort to a primal-dual framework to design the solution algorithm, which can be practically implemented as a set of distributed, interleaving auctions, where peers bid for bandwidth to download chunks at other peers considering the potential inter-ISP traffic as a cost factor. Such an auction-based mechanism encourages peers to download from neighbors with low network costs in between, in order to succeed in bandwidth acquisition for chunk downloading. We analyze and prove the social optimality achieved by the distributed auctions and verify the performance of our proposal using realistic emulation experiments with real P2P traffic.**

I. Introduction

Forecasts from Cisco's Visual Networking Index [1] reveal that P2P traffic is expected to grow to more than 7 Petabytes per month by 2014 – more than the double of the amount of P2P traffic in 2009. Most of the existing P2P protocols (for file sharing, live and on-demand media streaming, etc.) are network agnostic, *i.e.*, peers pick peers to download from as long as the latter cache the content in need, regardless of the ISP belonging of each other. This has resulted in significant amounts of inter-ISP traffic that cost many ISPs dearly, leading to their P2P traffic filtering, which in turn significantly deteriorates the performance of P2P applications.

There have been many efforts on minimizing inter-ISP traffic by connecting peers to nearby neighbors in the same AS or ISP. Aggarwal *et al.* [2] and Xie *et al.* [3] advocate collaboration between P2P applications and ISPs, where ISPs provide information of the underlying network (*e.g.*, bandwidth, distance) for a P2P application to make localized peer selection. Picconi *et al.* [4] propose a two-tier adaptive overlay structure, with highly clustered primary overlays built among nearby peers and a number of secondary links interconnecting the clusters. The secondary links are unchoked when necessary to enable global stream propagation. Peer selection is carried out at a coarse level in the above work, without guarantee of the optimality of the resulting topologies in content distribution performance and inter-ISP traffic reduction. Wang *et al.* [5] formulate an optimization problem for ISP-friendly rate allocation, aiming at guaranteed QoS for users, reduced server load and reduced ISP-unfriendly traffic. The rate allocation optimization problem is modeled and solved in the fluid level (optimal flow rate computation), and the results are translated into a packet-scheduling algorithm for implementation.

This paper presents our initial attempt to design P2P protocols that encourage ISP-aware peer selection, in a fully distributed algorithm framework. We first formulate a social welfare maximization framework for dynamical construction of the P2P content distribution topology (namely deciding who is downloading which content chunk from whom), where each peer's welfare is decided by its gain for receiving the chunks and the network cost incurred due to receiving chunks from different ISPs. Instead of flow-level optimization, our optimization framework models chunk-level content distribution among the peers, to enable straightaway implementation of the optimal chunk scheduling strategies in a real-world P2P system, and to avoid loss of optimality due to the flow rate-to-packet scheduling translation.

The optimization is a more difficult integer optimization problem though. Nevertheless, given its nature of an assignment problem, we resort to a primal-dual framework proposed by Bertsekas *et al.* [6] to derive the solution. The solution algorithm can be implemented as a set of distributed, interleaving auctions in the P2P system: Each peer acts as an auctioneer and hosts an auction to allocate its upload bandwidth for serving chunks to requesting peers; each peer also bids in the auctions hosted by different other peers for bandwidth to download chunks they want. The bidding price for a chunk cached at a neighbor is decided by the utility gain the peer can obtain after acquiring the chunk, minus the network cost of receiving the chunk from the neighbor. Such an auction-based algorithm framework encourage peers to download from neighbors with low network costs in between, in order to succeed in bandwidth acquisition for chunk downloading.

We also design practical protocols for carrying out the auctions consecutively in a dynamic P2P system, where peers may come and go, and upload bandwidth can be repeatedly sold to serve different chunks to different neighbors over time. We prove that the distributed auctions can collectively maximize

the social welfare of all peers in the system in Theorem 1. We implement an emulator of a large-scale, distributed P2P streaming system where content distribution is carried out through the auctions. Extensive experiment evaluations verify the good performance of the system in realistic environments with real P2P traffic.

The remainder of the paper is organized as follows: Sec. III presents the P2P content distribution system model and formulates the optimal chunk scheduling problem. Sec. IV presents a primal-dual auction algorithm to solve the problem and discusses its practical implementation as a set of distributed auctions. Sec. V presents our experimental results and Sec. VI concludes the paper.

II. Related Work

Developing ISP-aware P2P protocols has attracted significant attention from both the content distribution service providers and the research community. These research work can be categorized into three camps. The first one is to achieve the network awareness through cooperation between ISPs and P2P systems. Aggarwal et al. [2] propose a cooperative mechanism between ISPs and P2P users for a better neighbor selection process as follows: the ISPs offer an "oracle" to the P2P users, the peers send their lists of possible neighbors to the "oracle", and then the oracle ranks the possible peer neighbors according to certain criteria, such as their proximity to the peer or higher bandwidth links in between. Xie et al. [3] propose P4P, the provider portal for applications. P4P provides a control plane that can provide network information, such as network policy, p4p-distance, capabilities, to peer users. Additional infrastructures are necessary for P4P, which may not be easy to deploy. These mechanisms require trust between ISPs and P2P users. The second group of work achieves network awareness through inferring network information by peers or based on peers' self-adaptive protocols. Choffnes et al. [7] propose to use the information collected from content distribution networks to guide the biased peer selection. The rationale is as follows: if two clients are dispatched to a similar set of replica servers, they are likely to be close to these servers and more importantly, to each other. This leads to clustered overlay with a topology following the underlying physical one. Picconi et al. [4] propose an adaptive protocol for P2P live streaming with a large number of links between peers located in different ISPs. It builds a highly clustered primary overlay with dynamically unchoked secondary inter-cluster links. The clustered primary overlay can reduce the unnecessary inter-ISP traffic. The dynamically unchoked secondary inter-cluster links can ensure that the QoS is not impacted. These mechanisms use heuristic self-adaptive protocols to reduce the inter-ISP traffic and keep good performance. Our algorithm explores the optimality of peers' utility in an auction based on a primal-dual optimization framework. The third camp of work exploits rate control on inter-ISP links. Wang et al. [5] propose and formulate an optimization problem for rate allocation among peers in a P2P VoD system. The rate allocation optimization problem is modeled in the fluid level. It then translates the fluid-level rate allocation algorithm into an implementable packet-level scheduling algorithm. Our approach formulates a packet-level optimization problem for the rate allocation problem directly and avoids the loss of optimality due to the flow rate-to-packet scheduling translation.

A sequence of work by Bertsekas et al. [8] [6] theoretically study the auctions based on a primal-dual optimization framework for the assignment problems and transportation problems. Such auction-based optimization has also been applied in improving the performance of P2P content distribution [9] [10] [11]. These papers do not take ISP-awareness into consideration. To the authors' knowledge, our paper is the first in applying an auction for achieving the social optimality of ISP-aware P2P content distribution.

III. Problem Model

A. System Model

We consider a mesh-based P2P content distribution system, *e.g.*, a P2P VoD streaming system, deployed over the networks of M Internet Service Providers (ISPs). Let $\mathcal{P}_m$ denote the set of peers in ISP $m \in [1, M]$. There exist a number of tracker servers, from which the peers can obtain a set of neighbors which may potentially cache the content they want, upon joining the system. Let $\mathcal{N}_n(d)$ denote peer d's neighbor set in ISP n. The set of all neighbors of peer d is hence $\cup_{n=1}^{M}\mathcal{N}_n(d)$.

Each content (a file or a video stream) in the system is divided into multiple equal-sized chunks and distributed. The system works in a time slotted fashion over $t = 0, 1, 2, \ldots, T$, where T is a potentially larger integer. Each peer maintains a moving window of interest, specifying the chunks it wishes to download in each time slot (*e.g.*, the chunks to be played next in a streaming system). A peer exchanges buffer maps of chunk availability with its neighbors, and requests chunks of interest from the neighbors which cache the chunks. Let $\mathcal{R}_t(d)$ denote the set of chunks that peer d intends to download from neighbors at time slot t. A request in the system can be represented by a three-tuple (I_d, I_u, c), where I_d is the id of the downstream peer which issues the request, I_u is the id of the upstream peer being requested, and c is the identifier of the requested chunk. We use $B(u)$ to denote the upload bandwidth of peer u, which represents the number of chunks peer u can upload in a time slot (suppose one unit of bandwidth is used to upload one chunk). We assume that peers' upload bandwidth renders the bandwidth bottleneck in the system, while the download bandwidth is much more sufficient comparably.

Let $v^{(c)}(d)$ denote peer d's valuation for receiving chunk c, *i.e.*, the value chunk c brings to peer d. Let $a_{u\to d}^{(c)}$ be the indicator of whether request (I_d, I_u, c) is served, *i.e.*, $a_{u\to d}^{(c)} = 1$ if the request is served by the corresponding upstream peer u, and $a_{u\to d}^{(c)} = 0$ otherwise. The network cost for peer d to receive a chunk from peer u is $w_{u\to d}$, which has different values between peers in different pairs of ISPs. Such a network cost can represent network latency for sending a chunk between peers, or the possibility that the chunk is

TABLE I
IMPORTANT NOTATION

M	No. of ISPs
$\mathcal{P}_m$	Peers in ISP m
$\mathcal{N}_n(d)$	Peer d's total neighbor set in ISP n
$B(u)$	# of chunks peer u can upload in a time slot
I_d	Id of request source peer
I_u	Id of request destination peer
c	Index of requested chunk
$\mathcal{R}_t(d)$	set of peer d's interested chunks at time t
$\mathcal{N}_n^{(c)}(d)$	set of peer d's neighbors in ISP n with chunk c
$a_{u\to d}^{(c)}$	indicator of whether request r receives the bandwidth allocation
$v^{(c)}(d)$	valuation for peer d receiving chunk c
$w_{u\to d}$	network cost for transmitting a chunk from u to d
λ_u	dual variables for peer u's upload bandwidth
$\eta_d^{(c)}$	dual variables for request (I_d, c)

being blocked due to filtering of egress/ingress P2P traffic at one ISP. The net utility peer d receives by downloading chunk c from peer u is $v^{(c)}(d) - w_{u\to d}$.

The important notation in this paper is summarized in table I for ease of reference.

B. Social Welfare Maximization Problem

In each time slot, we seek to decide the optimal chunk scheduling strategy, $a_{u\to d}^{(c)}$, for all the chunk requests issued by all the peers, $\forall d \in \cup_{m=1}^{M}\mathcal{P}_m, c \in \mathcal{R}_t(d), u \in \cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d)$, to maximize the social welfare, *i.e.*, the total utility of peers from chunk downloading, as follows:

$$\max \sum_{d\in\cup_{m=1}^{M}\mathcal{P}_m} \sum_{c\in\mathcal{R}_t(d)} \sum_{u\in\cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d)} a_{u\to d}^{(c)}[v^{(c)}(d) - w_{u\to d}] \tag{1}$$

s.t.

$$\sum_{d,c:u\in\cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d), c\in\mathcal{R}_t(d)} a_{u\to d}^{(c)} \le B(u), \quad \forall u \in \cup_{m=1}^{M}\mathcal{P}_m, \tag{2}$$

$$\sum_{u\in\cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d)} a_{u\to d}^{(c)} \le 1, \quad \forall d \in \cup_{m=1}^{M}\mathcal{P}_m, c \in \mathcal{R}_t(d), \tag{3}$$

$$a_{u\to d}^{(c)} \in \{0,1\}, \quad \forall d \in \cup_{m=1}^{M}\mathcal{P}_m, c \in \mathcal{R}_t(d), u \in \cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d). \tag{4}$$

Objective function (1) is to maximize the total utility of all peers. Constraint (2) states that the total number of chunks a peer uploads to its neighbors should not exceed its upload bandwidth limit. Constraint (3) specifies that a peer will download a chunk from no more than one neighbor.

The problem in (1) is an integer linear program. We will design an efficient primal-dual auction algorithm to solve this integer linear program. Introducing dual variables λ_u, $\eta_d^{(c)}$ to constraints (2) and (3) respectively, the dual problem of (1) can be formulated as follows:

$$\min \sum_{u\in\cup_{m=1}^{M}\mathcal{P}_m} \lambda_u B(u) + \sum_{d\in\cup_{m=1}^{M}\mathcal{P}_m} \sum_{c\in\mathcal{R}_t(d)} \eta_d^{(c)} \tag{5}$$

s.t.

$$\lambda_u + \eta_d^{(c)} \ge v^{(c)}(d) - w_{u\to d}, \quad \forall d \in \cup_{m=1}^{M}\mathcal{P}_m, c \in \mathcal{R}_t(d), u \in \cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d), \tag{6}$$

$$\lambda_u \ge 0, \quad \forall u \in \cup_{m=1}^{M}\mathcal{P}_m, \tag{7}$$

$$\eta_d^{(c)} \ge 0, \quad \forall d \in \cup_{m=1}^{M}\mathcal{P}_m, c \in \mathcal{R}_t(d). \tag{8}$$

Note that we omit the integrality constraint (4) in the primal problem when formulating the dual. Nevertheless, we will show in the next section that our auction algorithm exactly solves the primal and dual problems, with optimal binary solutions to the primal problem.

IV. The Primal-Dual Auction Algorithm

A. Conversion to An Assignment Problem

The social welfare maximization problem in (1) can be treated as a transportation problem [6]. In a transportation problem, a set of source nodes are connected to a set of sink nodes in a bipartite graph. The set of matching edges between the sources and the sinks are being sought such that each source is connected to no more than α sinks and each sink is connected to no more than β sources, and the total weight on the selected edges is the largest. In our problem, a request for a specific chunk c from a peer d, *i.e.*, (I_d, c), can be treated as a source. Each peer u is a sink. An edge connects a source to a sink if the corresponding sink (peer u) is a neighbor of the requesting peer d and caches chunk c. The weight on an edge is $v^{(c)}(d) - w_{u\to d}$. By solving problem (1), we wish to find the subset of edges between the sources and the sinks, such that each source is connected to no more than one edge in the set (constraint (3)) and each sink (peer u) is connected to no more than $B(u)$ edges (constraint (2)).

A transportation problem is an assignment problem in nature. In Bertsekas *et al.*'s work [6] [8], an auction-like primal-dual algorithm is designed to solve the classical assignment problem, where X distinct objects are to be assigned to Y persons, such that each person receives one object, and the total weight of the person-object matchings is the largest. The transportation problem can be converted to an assignment problem by replacing each source (sink) with α (β) copies of persons (objects). To convert our problem to an assignment problem, each sink (peer u) is replaced by $B(u)$ units of upload bandwidth (treating one unit of upload bandwidth as one object). Each of the $B(u)$ objects connects to the sources that sink u connects to in the original problem, and the same weight as on the original edge is applied on the new edges.

An illustration of our problem in the transportation problem model, as well as its conversion to the assignment problem, is given in Fig. 1. Based on this conversion, we are able to design a primal-dual auction algorithm to solve problem (1), based on the idea of the auction algorithm in [6].

B. The Primal-Dual Auction Algorithm

The main idea of the primal-dual algorithm is as follows: Each peer maintains a unit price λ_u for one unit of its upload bandwidth, which corresponds to the dual variable in the dual

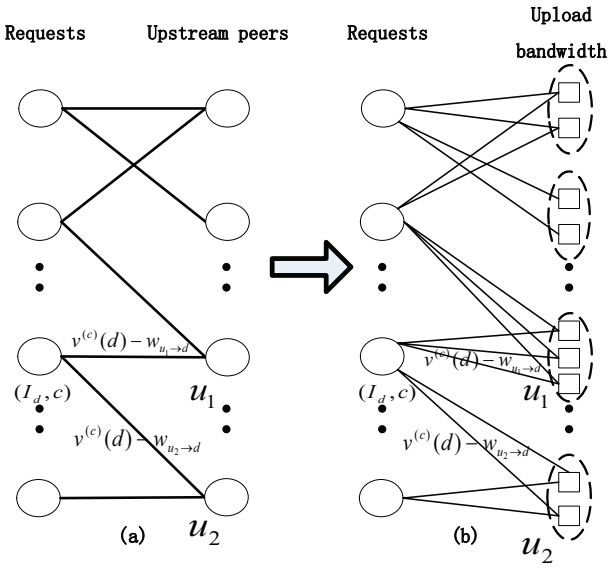

Fig. 1. An illustration of our transportation problem (a) and its conversion to the classical assignment problem (b).

problem (5). Peer u updates the price iteratively, according to the level of competition for its upload bandwidth, *i.e*, the relationship between the number of requests it receives $\sum_{d,c:u\in\cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d),c\in\mathcal{R}_t(d)} a_{u\to d}^{(c)}$ and its overall upload bandwidth $B(u)$, and allocates its upload bandwidth for serving chunks to peers (*i.e.*, computes $a_{u\to d}^{(c)}$) according to the utility that each chunk can bring to the corresponding requester, $v^{(c)}(d) - w_{u\to d}$. When the iterative process converges, we are able to show that the optimal binary solution $a_{u\to d}^{(c)*}$ to the primal problem and optimal solution λ_u^* to the dual problem are achieved. The optimal values of the other dual variables, $\eta_d^{(c)}$'s, are decided by $\eta_d^{(c)*} = \max_{u\in\cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d)}\{v^{(c)}(d) - w_{u\to d} - \lambda_u^*\}$, which is the minimum value (in order to minimize the objective function of the dual problem) that satisfies constraint (6).

Based on the above idea, we design a set of distributed, interleaving auctions to carry out the primal-dual algorithm. In each time slot, each peer u is an auctioneer, hosting an auction to sell its $B(u)$ units of upload bandwidth. The peers who wish to acquire one unit of upload bandwidth at peer u for retrieving one chunk c that u caches, are the bidders in this auction. We next describe the bidding strategy of the bidders and the allocation strategy at the auctioneers, respectively.

Bidding of Peer d: Peer d determines the set of chunks to download in this time slot, $\mathcal{R}_t(d)$, and values each chunk in $\mathcal{R}_t(d)$, *i.e.*, computes $v^{(c)}(d), \forall c \in \mathcal{R}_t(d)$ (*e.g.*, according to the playback deadline of a chunk in a P2P streaming system). Based on exchanged bitmaps with neighbors, peer d can decide the set of neighbors which cache chunk c, *i.e.*, $\mathcal{N}_n^{(c)}(d), n = 1,\ldots,M$. It then decides the following:

(1) From which neighbor to bid for one unit of upload bandwidth for retrieving chunk c. The net utility peer d can acquire by downloading c from peer $u \in \cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d)$ is decided by $v^{(c)}(d) - w_{u\to d} - \lambda_u$ (the bandwidth price λ_u at peer u is considered). Let u^* be the upstream peer that provides the largest utility, *i.e.*, $u^* = \text{argmax}_{u\in\cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d)} v^{(c)}(d) - w_{u\to d} - \lambda_u$. Peer d will send the request for chunk c to peer u^*.

(2) How much peer d should bid for one unit of upload bandwidth at peer u^*. Let $\varphi(d,c,u^*) = v^{(c)}(d) - w_{u^*\to d} - \lambda_{u^*}$ denote the largest net utility peer d can obtain by downloading chunk c. Suppose $\hat{u}$ is the neighbor which can provide the second largest net utility $\varphi(d,c,\hat{u}) = v^{(c)}(d) - w_{\hat{u}\to d} - \lambda_{\hat{u}}$m if peer d was to download c from $\hat{u}$. Peer d bids $b(d,c,u^*) = \lambda_{u^*} + \varphi(d,c,u^*) - \varphi(d,c,\hat{u}) = w_{\hat{u}\to d} - w_{u^*\to d} + \lambda_{\hat{u}}$ for one unit of bandwidth to download chunk c from u^*. If bid $b(d,c,u^*) = \lambda_{u^*}$, peer d will not send a bid to auctioneer u^*, since the bid will nevertheless be unsuccessful, according to the bandwidth allocation mechanism below. Instead, peer d waits until the bandwidth prices at the upstream peers change such that its optimal bid becomes larger than a respective bandwidth price.

Bandwidth Allocation at Peer u: Peer u maintains an assignment set containing the requests (I_d, c) with the highest bids, to which it will allocate one unit of its upload bandwidth (corresponding to $a_{u\to d}^{(c)} = 1$). The maximum size of the assignment set is $B(u)$. At the beginning of each time slot, the set is empty and the initial unit bandwidth price is set to $\lambda_u = 0$. Upon receiving a bid $b(d,c,u)$, if the price $b(d,c,u) \le \lambda_u$, peer u rejects the bid. Otherwise, if its assignment set is not full, peer u directly adds the request (I_d, c) to the set; if the set is full, the request whose bidding price is the lowest among all the requests in the assignment set (which equals λ_u), is removed from the set (*i.e.*, set the respective $a_{u\to d}^{(c)} = 0$), and request (I_d, c) is added. If the size of the assignment set is $B(u)$ (*i.e.*, all $B(u)$ units of its upload bandwidth are allocated), Peer u updates λ_u to the smallest bidding price among all accepted requests in the current assignment set, and informs its neighbors this updated bandwidth price.

A bid at an upstream peer u can be unsuccessful due to concurrent bids from other peers which push the price λ_u up, or can be accepted first but removed from the assignment set later on, due to the arrival of higher bids. In these cases, the bidder can compute its new bid according to the updated prices from the upstream peers, and bid again either to the same upstream peer u (if $v^{(c)}(d) - w_{u\to d} - \lambda_u$ is still the largest), or to another upstream peer u' (if $v^{(c)}(d) - w_{u'\to d} - \lambda_{u'}'$ becomes the largest).

The auction in each time slot repeats iteratively, until the bidding process converges, *i.e.*, no auctioneer u wishes to change its bandwidth allocation $a_{u\to d}^{(c)}$'s and price λ_u, and no bidder wishes to bid again. Then the chunks corresponding to the winning bids are transmitted to the respective bidders, using the acquired bandwidth. The auction algorithm for one time slot is summarized in Alg. 1. The fully distributed auction algorithm achieves maximized social welfare, as given in the following theorem.

Theorem 1: Under the assumption that the upload band-

width and the distribution of peers' cached chunks in the system can satisfy peers' downloading requirements in each time slot, Alg. 1 terminates and gives the optimal solution $a^{(c)*}_{u\to d}$ to the primal problem (1) and λ^*_u to the dual problem (5) upon termination.

The proof is given in Appendix A.

Algorithm 1 The Auction Algorithm in Time Slot t

*****At Bidder Peer d:*****

1: exchange buffer maps with neighbors and decide $\mathcal{R}_t(d)$
2: **for** each chunk c in $\mathcal{R}_t(d)$ **do**
3: calculate net utility $v^{(c)}(d) - w_{u\to d} - \lambda_u$ for all neighbors which cache c, and select neighbor u^* providing the largest net utility
4: send bid $b(d, c, u^*) = w_{\hat{u}\to d} - w_{u^*\to d} + \lambda_{\hat{u}}$ to neighbor u^*, where $\hat{u}$ is the neighbor providing the second largest net utility
5: **end for**
6: upon failure to acquire a unit of bandwidth for a chunk c at a neighbor u and price updates from neighbors
7: repeat Lines 3 and 4 with updated prices

*****At Auctioneer Peer u: *****

1: **Initialization:** $\lambda_u = 0$, assignment set $\mathcal{A} = \emptyset$
2: **while** a bid $b(d, c, u)$ is received **do**
3: **if** $b(d, c, u) \le \lambda_u$ **then**
4: reject the bid
5: **else**
6: **if** size of $\mathcal{A}$ equals $B(u)$ **then**
7: find a request $(I_{d'}, c')$ in $\mathcal{A}$ whose bid is the lowest, $\mathcal{A} \leftarrow \mathcal{A} - \{(I_{d'}, c')\}$
8: **end if**
9: $\mathcal{A} \leftarrow \mathcal{A} + \{(I_d, c)\}$
10: **if** size of $\mathcal{A}$ equals $B(u)$ **then**
11: update λ_u to the smallest bid among all requests in $\mathcal{A}$, and inform neighbors the new price
12: **end if**
13: **end if**
14: **end while**

C. Implementation Issues in a Dynamic P2P System

Over time, the auctions according to Alg. 1 repeat in each time slot, and upload bandwidth at each peer is repeatedly sold to serve different chunks to different neighbors. The time slot in our model can be treated as the bidding cycle, according to which peers decide the next batch of chunks to download and seek the upstream peers which can serve the chunks through the auctions. The actual chunk transfers happen as soon as the auction algorithm converges in each time slot (*i.e.*, when the optimal chunk scheduling is decided), and can be finished into the next time slot (*i.e.*, bidding for bandwidth for retrieving the next batch of chunks can happen concurrently with the transfer of the previous batch of chunks).

Peers may come and go in a dynamic P2P system. When an auctioneer peer u receives new bids from newly joined peers in the middle of a time slot, it delays handling of these bids until the start of the next time slot, such that the convergence of the auction process in this time slot is not disturbed. Upon departure of a peer when the auction algorithm is still running in a time slot, the algorithm can handle it smoothly and converge to the maximum social welfare where the departed peer is excluded. If an auctioneer peer departs when chunk transfers have started in a time slot, the unaffected chunk transfer schedules remain and the impact of the peer departure will be taken into consideration in the next time slot.

V. Performance Evaluation

To evaluate our auction based content distribution algorithm, we emulate an efficient multi-threaded P2P VoD system in Java and deploy it on a cluster of 6 high-performance blade servers with a 16-core Intel Xeon E5-2600 processor and 80GB RAM. Each peer in the system is emulated by one process. Real network traffic is sent between peers in the system. The program at each peer includes the following components: a neighbor manager for updating the peer's neighbors; a buffer manager for retaining chunks and exchanging bitmaps with the neighbors; a bidding module for calculating bids and sending them to auctioneers; an allocator module for determining the allocation of upload bandwidth; a transmission manager for transmitting chunks to the winning bidders. We emulate 5 ISPs. Peers of the same ISP are deployed in the same server. There is a track server which keeps track of online peers and bootstraps new joining peers with a list of neighbors with close playback positions.

We set up the experiments to emulate a realistic P2P video streaming system, such as YouTube [12], YouKu [13]. We use short video files just like most videos on YouTube, and the size of a video file is around 20 MB. The playback bitrate of a video is 640 Kbps, which is similar to the bitrate of a YouTube 360p video. We choose the chunk size of 8 KB just as the size of a sub-piece in PPStream [14]. There are 100 videos in the system.

Our emulator supports dynamic peer joins and departures. Peers join the system as a poison process with rate 1 peer per second, and are distributed in the 5 ISPs evenly. When a peer joins the system, it will select video i ($1 \le i \le 100$) to watch according to the Zipf-Mandelbrot distribution $p(i) = \frac{\frac{1}{(i+q)^\alpha}}{\sum_{i=1}^{100}\frac{1}{(i+q)^\alpha}}, \alpha = 0.78, q = 4$ [15]. The default number of neighbors for each peer is 30. A peer tries to pre-fetch 10 seconds of the video, *i.e.*, it tries to download the next 100 chunks in advance of the playback position. In each ISP, for each video, there are 2 seed peers with a upload bandwidth that is 8 times of the streaming rate, which cache the complete video. We know that there are different types of Internet connection services with different levels of upload bandwidth [16]. As the technology develops, both the streaming rates of Internet videos and the upload bandwidths of peers are moving to higher levels. Hence, we set the upload capacity of peers following the uniform distribution within the range of $[1, 4]$ times of the streaming bitrate in our system.

We use a deadline-based valuation function, $\frac{\alpha_d}{log(\beta_d+d)}$, for chunk evaluation at the peers [9], emphasizing how urgent a chunk is for playback. Here d is the time to the playback deadline of the chunk, α_d and β_d are constants with default values $\alpha_d = 2$ and $\beta_d = 1.2$. Hence, the deadline-based valuation is within the range of $[0.8, 8]$.

We use network latency as the network cost in our experiments. The inter-ISP link delay costs and intra-ISP link delay costs follow truncated normal distributions [17]. The distribution of inter-ISP link costs has a mean 5 and a standard variance 1, truncated within range $[1, 10]$. The distribution of intra-ISP link cost has a mean 1 and a standard variance 1, truncated within range $[0, 2]$.

A. Convergence of the Bandwidth Price

We first study the evolution of the price for one unit of upload bandwidth, λ_u, with our auction algorithm, in a static network with 500 peers. Each time slot lasts 10 seconds. During one time slot, a peer keeps bidding in order to acquire the bandwidth to receive the 100 chunks it wants next. Fig. 2 plots the evolution of the price λ_u at a representative peer. For better illustration, we only show the evolution of the price in the time slots between 150 seconds and 250 seconds, and the evolution of the price is similar in other time slots. We can see that the price converges after around 5 seconds in each time slot. This verifies the convergence of our auction-like primal-dual algorithm.

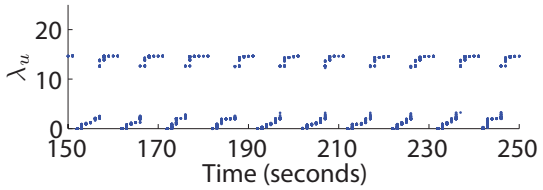

Fig. 2. The evolution of a peer's price λ_u.

In the following subsections, we compare the social welfare, inter-ISP traffic and chunk miss rate under our auction algorithm with a simple locality-aware chunk scheduling algorithm, as follows: each downstream peer requests chunks from upstream neighbors with the lowest network costs in between as much as possible; for bandwidth allocation at an upstream peer, it always prioritizes to transmit chunks with more urgent deadlines.

B. Social Welfare

Fig. 3 shows the evolution of the system's social welfare in each time slot in a dynamic P2P network, where peers arrive following the dynamic model described at the beginning of this section, and stay until they finish watching the respective video. We can see that as more peers join the system over time, larger social welfare per time slot can be achieved with our auction algorithm. However, the social welfare achieved by the simple locality-aware algorithm drops due to more inter-ISP traffic incurred with more peers in the system. The negative values of the social welfare with this algorithm are because it does not consider peers' chunk valuation when scheduling chunk transmissions (such that $v^{(c)}(d) - w_{u\to d}$ can be negative).

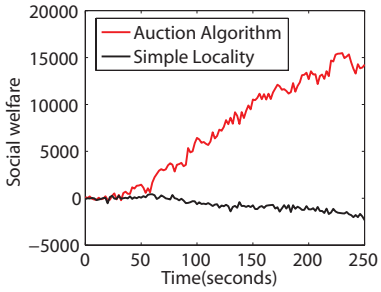

Fig. 3. Comparison of social welfare.

C. Inter-ISP Traffic

Fig. 4 shows the percentage of inter-ISP traffic incurred in all the traffic in the system in each time slot in a static network of 500 peers. We can see that the percentage of inter-ISP traffic is smaller with our auction algorithm, since with our algorithm, a peer only downloads a chunk from an ISP with a large network cost in between when its valuation for the chunk is large enough, reducing unnecessary inter-ISP traffic as much as possible.

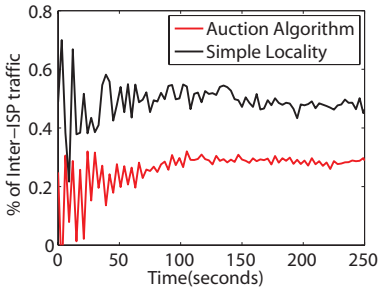

Fig. 4. Comparison of inter-ISP traffic.

D. Chunk Download Performance

Fig. 5 plots the averaged chunk miss rate of all peers in a static network of 500 peers, which is the percentage of chunks which fail to be downloaded before the respective playback deadlines. With our auction algorithm, the averaged chunk miss rate is smaller. This verifies the efficiency of upload bandwidth allocation in our auction algorithm, which takes downstream peers' valuation of the chunks into consideration.

E. Comparison under Peer Dynamics

Fig. 6(a), 6(b), and 6(c) show the social welfare, inter-ISP traffic and chunk miss rate with our algorithm and the simple locality protocol, in a dynamic P2P network where peers arrive following the dynamic model described at the beginning of this section, and depart at any time with probability 0.6. We can see that our algorithm still performs better in general than the simple locality-aware algorithm in case of peer dynamics.

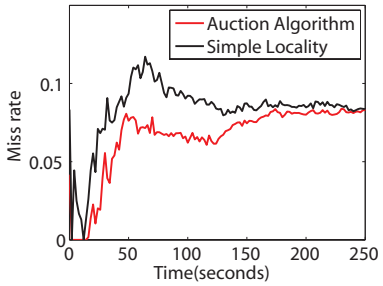

Fig. 5. Comparison of the chunk miss rate.

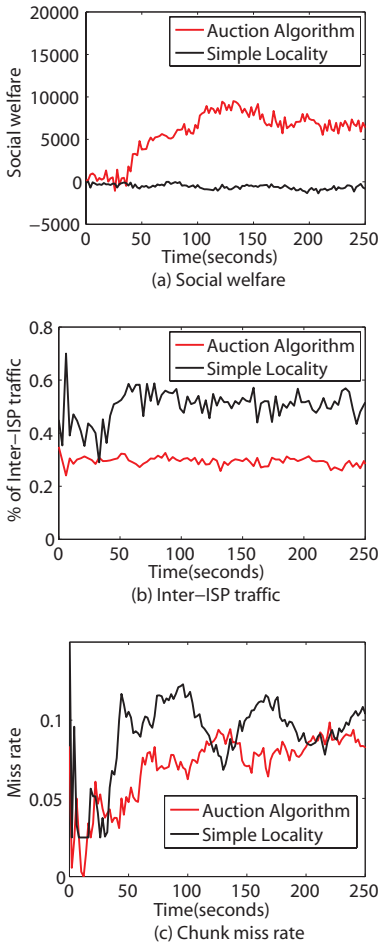

Fig. 6. Comparison of social welfare, inter-ISP traffic and chunk miss rate under peer dynamics.

VI. Conclusions

This paper addresses the optimal chunk dissemination topology construction problem in a P2P content distribution system, with the objective of social welfare maximization with minimal inter-ISP traffic. We utilize a primal-dual optimization framework and design an efficient auction algorithm to achieve the optimal dissemination topology in a fully distributed manner. Our experiments under realistic settings based on emulator implementation of a P2P streaming system verify the algorithm's efficacy in reducing ISP-unfriendly traffic and maintaining good chunk download performance. This work represents our initial attempt to design an auction-like mechanism to encourage peers' voluntary download from neighbors with low network costs in between. We are improving the auction mechanism design to enforce truthfulness of the bids in cases of selfish peers that may manipulate the mechanism, in our ongoing work.

Acknowledgement

The research was supported in part by a grant from Hong Kong RGC under the contract (Ref: HKU718710E).

References

[1] I. Cisco, "Cisco visual networking index: forecast and methodology, 2009-2014," *White Paper*.
[2] C. S. V. Aggarwal, A. Feldmann, "Can ISPs and P2P Users Cooperate for Improved Performance?" *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 31–40, 2007.
[3] H. Xie, Y. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider Portal for Applications," in *Proc. of ACM SIGCOMM*, August 2008.
[4] F. Picconi and L. Massoulie, "ISP Friend or Foe? Making P2P Live Streaming ISP-Aware," in *Proc. of IEEE ICDCS*, June 2009.
[5] J. Wang, C. Huang, and J. Li, "On ISP-friendly Rate Allocation for Peer-assisted VoD," in *Proc. of ACM Multimedia*, October 2008.
[6] D. P. Bertsekas and D. A. Castanon, "The Auction Algorithm for the Transportation Problem," *Annals of Operational Research*, vol. 20, pp. 67–96, 1989.
[7] F. E. B. D. R. Choffnes, "Taming the Torrent A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems," in *Proc. of ACM SIGCOMM*, August 2008.
[8] D. P. Bertsekas, "The Auction Algorithm: a Distributed Relaxation Method for the Assignment Problem," *Annals of Operational Research*, vol. 14, pp. 105–123, 1988.
[9] C. Wu, Z. P. Li, X. J. Qiu, and F. C. M. Lau, "Auction-based P2P VoD Streaming: Incentives and Optimal Scheduling," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 8, pp. 14:1–14:22, 2012.
[10] C. Wu, B. Li, and Z. Li, "Dynamic Bandwidth Auctions in Multioverlay P2P Streaming with Networking Coding," *Proc. of IEEE TPDS*, vol. 19, pp. 806–820, 2008.
[11] X. Chu, K. Zhao, Z. Li, and A. Mahanti, "Auction-Based On-Demand P2P Min-Cost Media Streaming with Network Coding," *Proc. of IEEE TPDS*, vol. 20, pp. 1816–1829, 2009.
[12] "YouTube," http://www.youtube.com.
[13] "YouKu," http://www.youku.com.
[14] J. Jia, C. Li, and C. Chen, "Characterizing PPStream across Internet," in *IFIP Network and Parallel Computing Workshop*, September 2007.
[15] J. Dai, B. Li, F. M. Liu, B. C. Li, and H. Jin, "On the Efficiency of Collaborative Caching in ISP-aware P2P Networks," in *Proc. of IEEE INFOCOM*, April 2011.
[16] C. Huang, J. Li, and K. W. Ross, "Can Internet Video-on-Demand Be Profitable?" in *Proc. of ACM SIGCOMM*, October 2007.
[17] H. Jiang and C. Dovrolis, "Passive estimation of TCP round-trip times," *Proc. of ACM SIGCOMM Computer Communications Review*, vol. 32, pp. 75–88, 2002.

APPENDIX

Proof: The proof consists of two parts: (i) The auction algorithm terminates in a finite number of iterations, and (ii) upon termination, the complementary slackness of the primal and dual optimization problems in (1) and (5) is satisfied.

(i) The termination of the auction algorithm can be proved by way of contradiction. Suppose it never terminates. Then the number of units of allocated bandwidth is non-decreasing, because one unit of bandwidth, once allocated, remains allocated throughout the auction. The total number of units of allocated bandwidth is upper-bounded by the overall bandwidth demand from downloading peers in the system. Under the assumption that the overall upload bandwidth is sufficient to serve each chunk, there exist units of upload bandwidth that are never allocated. Therefore, since the algorithm does not terminate, we can infer that there exists a peer wanting to download a chunk, which bids for one unit of bandwidth at an upstream peer u_1 whose bandwidth has all been allocated and whose price is growing unboundedly, rather than bids at another peer u_2 with a unit of unallocated bandwidth and bandwidth price 0. This implies that the valuation of downloading the chunk from peer u_2 with bandwidth price 0 is negative infinity (this is the only possibility when u_1 is always selected rather than u_2), contradicting the fact that the valuation should be finite.

(ii) We first list the complementary slackness conditions of the primal and dual problems:

$$\begin{cases} \lambda_u > 0 \rightarrow \sum_{d,c:u\in\cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d),c\in\mathcal{R}_t(d)} a_{u\rightarrow d}^{(c)} = B(u), \\ \qquad \forall u \in \cup_{m=1}^{M}\mathcal{P}_m, \\ a_{u\rightarrow d}^{(c)} > 0 \rightarrow \lambda_u + \eta_d^{(c)} = v_d^{(c)} - w_{u\rightarrow d}, \\ \qquad d \in \cup_{m=1}^{M}\mathcal{P}_m, c \in \mathcal{R}_t(d), u \in \cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d). \\ \eta_d^{(c)} > 0 \rightarrow \sum_{u\in\cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d)} a_{u\rightarrow d}^{(c)} = 1, \\ \qquad \forall d \in \cup_{m=1}^{M}\mathcal{P}_m, c \in \mathcal{R}_t(d). \end{cases}$$

The first condition means that the upload bandwidth at a peer u with a non-zero bandwidth price must have all been allocated. This is obviously true with our algorithm, since if there is one unit of unallocated bandwidth, the bandwidth price at peer u should be $\lambda_u = 0$.

The second condition states that when a request (I_d, c) obtains a unit of bandwidth from peer u, the optimal solution $\eta_d^{(c)}$ should be equal to $v_d^{(c)} - w_{u\rightarrow d} - \lambda_u$. Recall that the optimal value of $\eta_d^{(c)}$ is computed as $\eta_d^{(c)} = \max_{u'\in\cup_{n=1}^{M}\mathcal{N}_n^{(c)}(d)}\{v^{(c)}(d) - w_{u'\rightarrow d} - \lambda_{u'}\}$. Since the net utility $v_d^{(c)} - w_{u\rightarrow d} - \lambda_u$ for peer d to download chunk c from peer u is the largest among the net utilities from all neighbors that can provide chunk c to peer d, the second condition is satisfied.

The third condition states that when a request (I_d, c)'s achieved maximum utility is larger than 0, it definitely has acquired a unit of upload bandwidth. This is obviously true with the algorithm, since if a request does not receive a unit of upload bandwidth, its achieved maximum utility $\eta_d^{(c)}$ will be 0.