Computer Networks 55 (2011) 4033–4042

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

On tradeoffs between cross-ISP P2P traffic and P2P streaming performance ☆

Peng Yang *, Lisong Xu

Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588-0115, United States

A R T I C L E   I N F O

Article history:
Received 1 December 2010
Received in revised form 2 June 2011
Accepted 11 July 2011
Available online 30 July 2011

Keywords:
Peer-assisted video-on-demand streaming
Cross-ISP P2P traffic
Media server load
Goal Attainment method

A B S T R A C T

Peer-to-peer (P2P) technology greatly scales up traditional Internet streaming service. However, the philosophy that peers help one another distribute video content produces a large amount of cross-ISP P2P traffic. Newly proposed ISP-friendly P2P mechanisms reduce the cross-ISP P2P traffic by using locality-based peer selection mechanisms, in which the peers tend to connect to other peers in the same ISP domain. However, one unanswered question is how can we achieve the tradeoffs between the cross-ISP P2P traffic and the P2P streaming performance if there are conflicts between them. In this paper, we propose a rate allocation mechanism for achieving the tradeoffs between the cross-ISP P2P traffic and the P2P streaming performance. We model the tradeoffs as a multiobjective optimization problem and solve it using the Goal Attainment method. The evaluation result shows that our mechanism is able to achieve the tradeoffs between the conflicts and maintain good fairness among the peers and the streaming overlays.

1. Introduction

Peer-to-peer (P2P) technology has shown a great success in Internet streaming (video-on-demand, live streaming, etc.) and other content distribution services. However, it is facing increasing obstructions from Internet service providers (ISPs), which accuse the P2P technology of producing a large amount of cross-ISP P2P traffic. The tussles between the ISPs and the P2P technology have already caused some ISPs to throttle or even prohibit P2P traffic. P2P streaming applications, being bandwidth demanding, are especially hurt by such a hostile attitude held by the ISPs.

Recent ISP-friendly P2P mechanisms [2–5] reduce the cross-ISP P2P traffic by proposing locality-based peer selection mechanisms, where the peers connect to other peers in the same ISP domain. However, one concern about the locality-based peer selection is that it prevents peers from utilizing idle bandwidth resources at the remote peers. Sometimes, such resources are critical to the performance of the P2P streaming applications, since it helps the peers combat peer churn and maintain smooth playback. Thus, a P2P traffic control mechanism is needed to achieve the tradeoffs between the cross-ISP P2P traffic and the P2P streaming performance.

In this paper, we propose a rate allocation mechanism to achieve the tradeoffs between the cross-ISP P2P traffic and the P2P streaming performance for P2P assisted video-on-demand streaming. The mechanism proposed is also applicable to other types of P2P systems. The reason that we consider peer-assisted video-on-demand systems is that it is more challenging to design such systems than to design peer-assisted file-sharing systems. We consider two performance metrics: upload bandwidth utilization and media server load. Upload bandwidth is used by the peers to distribute video content to one another. Reducing the cross-ISP P2P traffic may lower the upload bandwidth utilization and cause the download rates of some peers to drop below their required streaming rates. To maintain smooth playback, those peers have to download a portion of the video content from their media servers, which increases the media server load.

☆ This paper is an extension of the work reported in [1].

* Corresponding author.
E-mail addresses: pyang@cse.unl.edu (P. Yang), xu@cse.unl.edu (L. Xu).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.07.013

The proposed rate allocation mechanism models the above tradeoffs as a multiobjective optimization problem and solves it using the Goal Attainment method [6]. Our mechanism optimizes the cross-ISP P2P traffic, the upload bandwidth utilization, and the media server load simultaneously and achieves tradeoffs among them if there is any conflict. We also incorporate a utility-based resource allocation mechanism into the multiobjective optimization problem, which enables our mechanism to fairly allocate the upload bandwidth to the competing peers according to their current download rates and required streaming rates. We evaluate our mechanism and compare it with the Weighted Sum method described in [7] and the even allocation method where the upload bandwidth is equally allocated to the competing connections. The advantage of our method is that it achieves finer tradeoff tuning than the Weighted Sum method. Finer tradeoff tuning allows more precise control of the cross-ISP P2P traffic and the P2P streaming performance. Compared with the even allocation method, our method not only provides the tradeoffs among the conflicting objectives but also produces fairer rate allocation result.

The rest of the paper is organized as follows: in Section 2, the related work is discussed; in Section 3, the problem is formulated; in Sections 4 and 5, the details of our proposed mechanism are discussed; in Section 6, the proposed mechanism is evaluated and compared with the Weighted Sum method and the even allocation method; finally, we conclude the paper in Section 7.

2. Related work

There is very little work studying the tradeoffs between the cross-ISP P2P traffic and the performance of streaming overlays. The work by Wang et al. [7] is the most related. The authors use the Weighted Sum method to achieve the tradeoffs. However, as shown in the paper, such a method can only achieve very coarse tradeoff tuning; whereas our approach is able to provide much finer tradeoff tuning.

The tussles between ISPs and P2P applications is studied in [8,2,3], etc. Liu et al. [8] study the noncooperative interaction between the ISPs and the P2P applications, where the P2P applications and the ISPs act independently to realize their own objectives. The authors show that such non-cooperative interaction may result in sub-optimal performance for both the ISPs and the P2P applications. Choffnes and Bustamante [3] and Xie et al. [2] focus on the cost of the cross-ISP P2P traffic and propose methods to greatly reduce it. Choffnes and Bustamante [3] propose a locality-based peer selection mechanism. The mechanism uses content distribution networks to discover local peers. Xie et al. [2] propose a cooperative framework between the ISPs and the P2P applications, where the information from the ISPs is used to guide the peer selection and to reduce the cross-ISP P2P traffic.

Bandwidth allocation among multiple overlays is studied in [9–11], etc. In [9], the problem is studied using a constrained utilization maximization framework, where the upload bandwidth is allocated to different overlays according to their priorities. In [10], the same problem is visited again using game theory where the upload bandwidth is auctioned by upstream peers to maximize their revenue, and downstream peers submit their bids to minimize their costs. Wang et al. [11] improve the allocation efficiency by utilizing a divide-and-conquer strategy, which allocates the upload bandwidth for a set of peers instead of for every single peer. The strategy improves the scalability of the upload bandwidth allocation.

3. Problem formulation

We consider a very general P2P streaming overlay structure, where a P2P streaming overlay is modeled as a set of peers (we use "peer" to refer to a logical node in a P2P overlay and use "node" to refer to a physical node) that are connected by overlay connections. All overlay connections are unidirectional; a peer on one end of a connection downloads the video content from an upstream peer on the other end.

Our goal is to design an efficient connection rate allocation mechanism that fulfills the following design goals and achieves the tradeoffs among them if there is any conflict.

- The cross-ISP P2P traffic is minimized.
- The upload bandwidth utilization is maximized.
- The media sever load is minimized.
- The fairness among the peers and the overlays is maintained.

Optimizing the first three design goals simultaneously is difficult. Firstly, minimizing the cross-ISP P2P traffic may cause the download rates of some peers to drop below their required streaming rates. This may increase the media server load, since those peers have to download a portion of video content from the media servers. Secondly, even if the media server load is minimized, minimizing the cross-ISP P2P traffic may still lower the upload bandwidth utilization by limiting cross-ISP prefetching [7]. The cross-ISP prefetching happens when the download rate of a peer is larger than its required streaming rate and the peer downloads extra streaming data from some remote peers. Such a behavior is very important to the performance of peers, since it helps peers combat against peer churn and bandwidth jitter.

Fairness is another important factor. Good fairness means that the upload bandwidth should be allocated to the competing peers according to their current download rates and their required streaming rates. The peers with smaller download rates or larger required streaming rates should be favored over the peers with larger download rates or smaller required streaming rates, i.e., the bandwidth should be allocated according to the law of diminishing returns.

3.1. Model and notation

There are n^o peers in streaming overlay o and we denote them by $p_i^o,\ i = 1, 2, \ldots, n^o$. Each peer p_i^o has a set of upstream neighbors (peers) from which it receives the video content; it also has a set of downstream neighbors to which it sends the video content. Let S_i^o denote the set of upstream neighbors of peer p_i^o, x_{ij}^o denote the rate from peer p_j^o to peer p_i^o, and x_i^o denote the total download rate of peer p_i^o, we have $x_i^o = \sum_{p_j^o \in S_i^o} x_{ij}^o$. If x_i^o is smaller than the required streaming rate R^o, peer p_i^o will have to download a portion of video content from the media server at rate $R^o - x_i^o$, which is denoted by x_{i0}^o. On the other hand, if x_i^o is larger than R^o, the extra streaming data will be cached for further use (i.e., prefetching).

Connection rate x_{ij}^o is limited by several factors such as the bandwidth bottlenecks inside the network and the upload bandwidth offered by the upstream peers. For most P2P systems, the upload bandwidth is the primary limitation. Thus, we only consider the upload bandwidth limitation in this paper. Let D_i^o denote the set of the downstream neighbors of peer p_i^o. The upload rate of peer p_i^o is $\sum_{p_j^o \in D_i^o} x_{ji}^o$ and is limited by the upload bandwidth offered by the node hosting peer p_i^o. In general, a node may host multiple peers in different P2P overlays. We denote a node by h, its upload bandwidth by u_h, and the set of peers hosted by node h by P_h, The upload bandwidth limitation can be expressed by the following inequality

$$\sum_{p_i^o \in P_h} \sum_{p_j^o \in D_i^o} x_{ji}^o \leqslant u_h, \quad \forall h. \tag{1}$$

The peers are distributed in different autonomous systems (ASs or ISPs, depending on the context, we use these two terms interchangeably). The connections among the peers are either created randomly, which is referred to as *random peer selection*, or created mostly within the same AS, which is referred to as *locality-based peer selection*. To quantify the locality-based peer selection, we describe it by a metric called a *locality degree*, which is a number between [0, 1.0]. A locality degree of m means that $m \times 100$ percent of the neighbors of a peer must come from the same AS as that of the peer. A locality degree of 0.0 is particular, and means that the neighbors of a peer are completely randomly sampled from all ASs. In this case, the locality-based peer selection reduces to the random peer selection.

We summarize the notation in Table 1.

3.2. Representing the design goals as optimization problems

The design goals presented above can be represented by separate optimization problems:

3.2.1. Minimizing the cross-ISP P2P traffic

The connections that cross the boundaries of the ISPs without peering agreements are more expensive than other connections. Thus we associate a connection cost with each connection. We use $c(x_{ij}^o),\ j \neq 0$ to denote the connection cost for connection rate $x_{ij}^o,\ j \neq 0$, and use $c(x_{i0}^o)$ to denote the cost for peer p_i^o to receive the video content from its media server.

Table 1
Notation.

Notation	Meaning
n^o	Number of peers in streaming overlay o
p_i^o	A peer in streaming overlay o
S_i^o	The set of upstream peers of peer p_i^o
D_i^o	The set of downstream peers of peer p_i^o
x_{ij}^o	Streaming rate from peer p_j^o to p_i^o
x_i^o	Total download rate of peer p_i^o
R^o	Required streaming rate of streaming overlay o
x_{i0}^o	Streaming rate from media servers to peer p_i^o
h	a node
u_h	Upload bandwidth of node h
P_h	The set of peers hosted by node h

Minimizing the cross-ISP P2P traffic can be modeled by the following optimization problem:

$$\min \left(\sum_o \sum_{p_i^o, p_j^o} c(x_{ij}^o) + \sum_o \sum_{p_i^o} c(x_{i0}^o) \right) \tag{2}$$

s.t.

$$\sum_{p_i^o \in P_h} \sum_{p_j^o \in D_i^o} x_{ji}^o \leqslant u_h, \quad \forall h \tag{3}$$

and

$$x_{ij}^o \geqslant 0, \quad \forall o,\ p_i^o,\ p_j^o. \tag{4}$$

3.2.2. Maximizing the upload bandwidth utilization

Maximizing the upload bandwidth utilization is crucial to the performance of the streaming overlays. We model this design goal by the following utility maximization problem:

$$\max\ u_i^o(x_i^o), \quad \forall p_i^o, \tag{5}$$

which is subject to Constraints (3) and (4), where $u_i^o(x_i^o),\ \forall p_i^o$ is an increasing concave function of x_i^o, which is called the utility of peer p_i^o. The following theorem shows that by maximizing the utility, the utilization of the upload bandwidth is maximized.

Theorem 1. *The solution $(x_i^o)^*,\ \forall p_i^o$ of the utility maximization problem maximizes the upload bandwidth utilization.*

Proof. We prove this theorem by contradiction. Suppose that $(x_i^o)^*,\ \forall p_i^o$ can not maximize the upload bandwidth utilization. Then, there must be some idle upload bandwidth at some node h. Suppose that peer p_i^o downloads the video content from peer p_j^o on h. Then, we can increase connection rate x_{ij}^o to take up the idle upload bandwidth. This leads to larger $u_i^o(x_i^o)$, which contradicts the fact that $u_i^o(x_i^o)$ is maximized. □

3.2.3. Minimizing the media server load

Minimizing the media server load can be modeled by the following optimization problem:

$$\min \sum_{p_i^o} x_{i0}^o, \quad \forall o, \tag{6}$$

which is subject to Constraints (3) and (4).

3.2.4. Fairness

The fairness among the peers and the streaming overlays can be maintained by using a proper utility function. We propose to use $u_i^o(x_i^o) = R^o \times \log(x_i^o),\ \forall p_i^o$. The logarithm utility function is studied in [12,13] etc. As the logarithm utility function is strictly concave, it realizes the law of diminishing returns. That is, the peers with smaller total download rates are favored in the upload bandwidth allocation over the peers with larger total download rates. Also, since the logarithm utility function realizes proportional fairness (i.e., proportional to R^o), peers in different streaming overlays are allocated upload bandwidth proportional to their required streaming rates, i.e., the peers with larger required streaming rates get more upload bandwidth than the peers with smaller required streaming rates.

The fairness is only achieved by the cooperation of peers to abide by the above utility function. In practice, there may be non-cooperating peers, which break the utility function to gain unfair advantage over cooperating peers. Although the logarithm function itself can be coded into P2P software, an upstream peer still needs two parameters from each downstream peer in order to allocate the upload bandwidth to them: one is the required streaming rate; the other one is the total download rate. To gain unfair advantage, a non-cooperating peer may report false information about the total download rate and the required streaming rate. The problem of a false required streaming rate can be solved by letting a trusted media server report the required streaming rate for the peers in the corresponding streaming overlay. The problem of a false total download rate is hard to tackle, since the total download rate is normally maintained by a peer itself. Handling these problems is out of the scope of this paper. However, similar problems are studied in [14,15], etc. to handle false reputation reports sent by the peers in reputation-based P2P systems. Interested readers may refer to those references for related information.

4. The tradeoffs among the conflicting design goals

We achieve the tradeoffs among the conflicting design goals by modeling the tradeoffs as an optimization problem and solving it using the Goal Attainment method. Along with the Goal Attainment method, we also briefly discuss other alternatives including the Weighted Sum method proposed in [7] and the basic even allocation mechanism, where the upload bandwidth is equally allocated to the competing connections.

4.1. The Goal Attainment method

We use the Goal Attainment method, which is abbreviated to *GA*, to optimize the design goals simultaneously and achieve the tradeoffs among them if there is any conflict. The method can be modeled by the following optimization problem, which is also referred to as *GA*:

$$\min\ r \tag{7}$$

subject to

$$\sum_{p_i^o} x_{i0}^o - w_s^o \times r \leqslant g_s^o, \quad \forall o, \tag{8}$$

$$\left(\sum_o \sum_{p_i^o, p_j^o} c(x_{ij}^o) + \sum_o \sum_{p_i^o} c(x_{i0}^o) \right) - w_c \times r \leqslant g_c, \tag{9}$$

$$(-u_i^o(x_i^o)) - w_i^o \times r \leqslant -g_i^o, \quad \forall o,\ p_i^o, \tag{10}$$

$$\sum_{p_i^o \in P_h} \sum_{p_j^o \in D_i^o} x_{ji}^o \leqslant u_h, \quad \forall o,\ p_i^o,\ p_j^o, h, \tag{11}$$

$$x_{ij}^o \geqslant 0, \quad \forall o,\ p_i^o,\ p_j^o. \tag{12}$$

The design goals become a part of the constraints in *GA*. For example, minimizing the media server load corresponds to Constraint (8). The constraint limits the media server load to $g_s^o + w_s^o \times r$ at most. The objective min r further minimizes $g_s^o + w_s^o \times r$, which minimizes the media server load indirectly. Parameters g_s^o reflects our expectation of the media server load. Parameter w_s^o is the weight of the media server load. For a fixed solution r^*, smaller weight w_s^o means that the media server load may be closer to our expectation. Constraints (9) and (10) can be interpreted similarly for minimizing the cross-ISP P2P traffic and maximizing the upload bandwidth utilization, respectively.

Depending on solution r^*, the total connection cost (cross-ISP P2P traffic) in Constraint (9), the utility functions (upload bandwidth utilization) in Constraint (10), and the media server load in Constraint (8) may over-attain (meet) or under-attain their expectations. A negative solution r^* means that they all over-attain their expectations so that no more tradeoffs are needed. On the other hand, a positive solution r^* means that at least one of them under-attains its expectation. In this case, the weights in the constraints represent the relative under-attainment of their corresponding expectations, as suggested by Theorem 2. By adjusting the weights, we adjust the relative under-attainment of those design goals, and achieve the tradeoffs among them.

Theorem 2. *By using $w_s^o = \rho_s^o \times g_s^o,\ w_c = \rho_c \times g_c$ and $w_i^o = \rho_i^o \times g_i^o$, where $\rho_s^o,\ \rho_c$ and ρ_i^o are nonnegative parameters, the expectations of their corresponding design goals are under-attained by at most $r \times \rho_s^o \times 100,\ r \times \rho_c \times 100$ and $r \times \rho_i^o \times 100$ percents, respectively.*

Proof. From Constraint (8), the media server load is at most $g_s^o + w_s^o \times r$. Thus the expectation is under-attained by $\frac{w_s^o \times r}{g_s^o} = r \times \rho_s^o$ at most. From Constraint (9), the total connection cost is at most $g_c + w_c \times r$. Thus the expectation is under-attained by $\frac{w_c \times r}{g_c} = r \times \rho_c$ at most. Similarly, from Constraint (10), the utility of peer p_i^o is at least $g_i^o - w_i^o \times r$. Thus the expectation is under-attained by $\frac{w_i^o \times r}{g_i^o} = r \times \rho_i^o$ at most. □

Other parameters of *GA* can be set as follows. The expectation of media server load g_s^o should be set by its operator according to the computational power of that media server and its outgoing bandwidth. The expectation of total connection cost g_c should be set by all ISPs jointly. The expectation g_i^o is set to $u_i^o(R^o)$ at least, i.e., every peer in streaming overlay o should receive a streaming rate of at least R^o from its upstream peers to ensure smooth playback.

4.2. The Weighed Sum method

The Weighted Sum method, which is abbreviated to *WS*, is another method to achieve the tradeoffs among the design goals. It optimizes the design goals by solving an optimization problem whose objective function is the weighted sum of those design goals. The method can be modeled by the following optimization problem, which is also referred to as *WS*:

$$\min\ (a-b)\sum_{p_i^o} x_{i0}^o \tag{13}$$

$$+\,b\sum_{\forall h}\left(u_h-\sum_{p_i^o\in P_h}\sum_{j\in D_i^o} x_{ji}^o\right) \tag{14}$$

$$+\left(\sum_o\sum_{p_i^o,p_j^o} c(x_{ij}^o)+\sum_o\sum_{p_i^o} c(x_{i0}^o)\right), \tag{15}$$

which is subject to Constraints (3) and (4). Variables a and b are parameters that adjust the weights associated with the design goals.

4.3. The even allocation mechanism

The even allocation mechanism, which is abbreviated to *EA*, is a basic upload bandwidth allocation scheme. Unlike *GA* and *WS*, in which the connection rates are determined by solving some optimization problems, the connection rates in *EA* are determined by a much simpler rule, i.e., it allocates an equal share of the upload bandwidth to the competing connections. Because of this, *EA* cannot achieve the tradeoffs among the design goals. In this paper, we use *EA* as a performance baseline.

4.4. Performance comparison among GA, WS and EA

The adoption of *GA* rests with its performance. We claim that by setting appropriate parameters for *GA*, its performance is never worse than *EA* and *WS*. We consider the following performance metrics: the media server load, the ISP cost, and the download rates of the peers. For simplicity, we only consider one overlay o. However, the result can be applied to multiple overlays as well. We denote the performance of *EA* or *WS* by a vector $\langle M_s^o,\ M_c^o,\ M_i^o,\ i=1\ldots n^o\rangle$, where M_s^o is the media server load produced by applying *EA* or *WS*; M_c^o is the ISP cost; and $M_i^o,\ i=1\ldots n^o$ are the download rates of the peers in overlay o. We further denote the connection rates, which lead to this performance, by M_{ij}^o for the connection rate from peer p_j^o to p_i^o and denote the rate at which peer p_i^o receives the video content from the media server by M_{i0}^o. We validate our claim by proving that $M_{ij}^o,\ \forall i,\ \forall j$ is a feasible solution to Problem *GA*.

Theorem 3. *The connection rate $M_{ij}^o,\ \forall i,\ \forall j$ produced by applying EA or WS is a feasible solution to GA.*

Proof. Since $M_{ij}^o,\ \forall i,\ \forall j$ is a solution to EA or WS, it automatically satisfies Constraints (3) and (12) in GA. To see that it satisfies Constraints (8)–(10), we set r in those constraints to 0 and rewrite the constraints to

$$\sum_{p_i^o} M_{i0}^o \leqslant g_s^o, \tag{16}$$

$$\left(\sum_o\sum_{p_i^o,p_j^o} c(M_{ij}^o)+\sum_o\sum_{p_i^o} c(M_{i0}^o)\right)\leqslant g_c, \tag{17}$$

$$u_i^o(M_i^o)\geqslant g_i^o,\quad \forall i. \tag{18}$$

Note that $g_s^o,\ g_c$ and $g_i^o,\ \forall i$ are all adjustable parameters of GA. By setting appropriate values for them, we can guarantee that the above constraints are all satisfied. Thus $M_{ij}^o,\ \forall i,\ \forall j$ is a feasible solution to GA. □

Since, $M_{ij}^o,\ \forall i,\ \forall j$ is only a feasible solution, not necessarily the optimal solution to *GA*, the performance of *GA* is no worse than $M_{ij}^o,\ \forall i,\ \forall j$. Thus, we prove our claim.

As explained in the next paragraph, the performance metrics achieved by *WS* are extreme tradeoff points, e.g. the cross-ISP P2P traffic is either maximized or completely eliminated. Although, *GA* can certainly achieve those extreme tradeoff points as proved by Theorem 3, it cannot go beyond the extreme tradeoff points. However, the advantage of *GA* is that it can also achieve a wide range of tradeoff points between the extreme points as shown in the next paragraph.

GA can produce finer tradeoff tuning than *WS*. Actually, *WS* can only produce on–off tradeoffs. Let's focus on the cross-ISP P2P traffic x_{ij}^o in *WS*. The objective function (13) is either an increasing function of x_{ij}^o or a decreasing function of x_{ij}^o depending on the values of Parameter a and b in *WS* and the parameters in the connection cost. Suppose that the cost $c(x_{ij}^o)$ can be expressed by $c(x_{ij}^o)=c\times x_{ij}^o$ and the cost $c(x_{i0}^o)$ can be expressed by $c(x_{i0}^o)=c\times x_{i0}^o$ for the cross-ISP P2P traffic x_{ij}^o and x_{i0}^o, respectively, where $c>0$ is some positive constant. For peer p_i^o, we have the following two cases: (1) $c(x_{io}^o)=0$ and (2) $c(x_{i0}^o)\neq 0$. For the former case, the derivative of the objective function (13) in *WS* to x_{ij}^o is $c-b$. If $c>b$, then the objective function is an increasing function of x_{ij}^o. We can decrease x_{ij}^o to 0 to optimize the objective function, which eliminates the cross-ISP P2P traffic. If $c<b$, then the objective function is a decreasing function of x_{ij}^o. We can increase x_{ij}^o as much as possible to optimize the objective function, which maximizes the cross-ISP P2P traffic. For the latter case, the derivative is $-a$. For any $a>0$, the objective function is a decreasing function of x_{ij}^o. We can increase x_{ij}^o as much as possible to optimize the objective function, which maximizes the cross-ISP P2P traffic. Thus, the cross-ISP P2P traffic is either maximized or completely eliminated. No tradeoff point between these two extreme points exists. On the contrary, *GA* can achieve a range of wide tradeoff points between the two extreme points as shown in Section 6. Further advantages of *GA* over *WS* include that the expectation of each design goal can be explicitly specified; and important design goals can be enforced to meet their expectations by setting their weights to 0.

5. A distributed algorithm for *GA*

5.1. Distributed algorithm

We propose a distributed algorithm to solve *GA* based on the primal–dual decomposition. It divides problem *GA* into sub-problems that can be solved individually by peers, media servers and ISPs.

For each node h, the algorithm initially allocates its upload bandwidth u_h to the peers in P_h proportional to their required streaming rates. We denote the upload bandwidth of peer $p_i^o\in P_h$ by u_h^o. Then the algorithm sets connection rate x_{ji}^o, $\forall j \in D_i^o$ to $\frac{u_h^o}{|D_i^o|}$ initially, and updates the connection rate at a regular interval as follows:

$$\dot{x}_{ji}^o = \Delta_j^u \cdot \frac{\partial u_j^o(x_j^o)}{\partial x_{ji}^o} + \Delta_j^s - \Delta_c \cdot c_{ji}^o - \Delta_h, \tag{19}$$

$$x_{ji}^o = [x_{ji}^o + \delta_{ij}^o \times \dot{x}_{ji}^o]^+, \tag{20}$$

where variables Δ_j^u, Δ_j^s, Δ_c and Δ_h are either maintained by peer p_i^o or needed to be transmitted to it, and δ_{ij}^o is the step size for the update. Specifically, variable Δ_h is maintained by peer p_i^o, and is updated at a regular interval as follows:

$$\Delta_h = \left[\Delta_h + \left(\sum_{p_i^o \in P_h} \sum_{j \in D_i^o} x_{ji}^o - u_h\right)\right]^+. \tag{21}$$

Variable Δ_h indicates the usage of the upload bandwidth offered by node h. Larger Δ_h means that more video content is queued at node h waiting to be sent. According to Eq. (19), peer p_i^o tries to lower connection rate x_{ij}^o to reduce the video content queued at node h.

Variable Δ_j^u is updated by peer p_j^o as follows and transmitted to peer p_i^o.

$$\Delta_j^u = \left[\Delta_j^u + (-u_j^o(x_j^o) - w_j^o \cdot r + g_j^o)\right]^+. \tag{22}$$

Variable Δ_j^u reflects the performance of peer p_j^o. Larger Δ_j^u means that the total download rate of peer p_j^o is smaller. According to Eq. (19), larger Δ_j^u leads to a larger increment of connection rate x_{ji}^o. This feature ensures that our proposal conforms to the law of diminishing returns.

Variable Δ_j^s is updated by its corresponding media server and transmitted to peer p_j^o, then to peer p_i^o. The updating rule is shown below.

$$\Delta_j^s = \left[\Delta_j^s + \left(\sum_{p_i^o} x_{i0}^o - w_s^o \times r - g_s^o\right)\right]^+, \quad x_j^o < R^o \tag{23}$$

or

$$\Delta_j^s = 0, \quad x_j^o \geqslant R^o. \tag{24}$$

When $x_j^o < R^o$, i.e., peer p_j^o receives video content from its media server, variable Δ_j^s is updated according to Eq. (23). Variable Δ_j^s becomes larger when more and more video content has to be streamed by the media server itself. According to Eq. (19), peer p_i^o tries to increase connection rate x_{ji}^o so that peer p_j^o will download less video content from the media server. When $x_j^o \geqslant R^o$, variable Δ_j^s is set to 0. This is because when peer p_j^o does not receive any video content from the media server, the value of variable Δ_j^s should not impact its download rate.

Variable Δ_c is maintained by the ISPs as follows and transmitted to peer p_i^o.

$$\beta = \sum_o \sum_{p_i^o, p_j^o} c(x_{ij}^o) + \sum_o \sum_{p_i^o} c(x_{i0}^o), \tag{25}$$

$$\Delta_c = [\Delta_c + (\beta - w_c \times r - g_c)]^+. \tag{26}$$

Variable Δ_c becomes larger when the total connection cost increases. According to Eq. (19), peer p_i^o tries to decrease the rate to its downstream peers and lower the total connection cost. Note that variable c_{ji}^o in (19) scales variable Δ_c at peer p_i^o. It represents the cost policy of ISPs. For example if rate x_{ji}^o is local to an ISP, variable c_{ji}^o can be set to 0. Then variable Δ_c has no impact on rate x_{ji}^o. On the contrary, if rate x_{ji}^o is cross-ISP P2P traffic, it can be associated with a large value of variable c_{ji}^o to amplify the impact of variable Δ_c on the rate.

5.2. Implementation concerns

In real implementation, the distributed algorithm is solved by peers, media servers, and ISPs together. The key to implement the distributed algorithm is to transmit variables Δ_j^u, Δ_j^s, and Δ_c to peer p_i^o. Variable Δ_j^u can be transmitted from peer p_j^o to peer p_i^o in their periodic communications. In P2P assisted streaming systems (or any other P2P systems), two connected peers often exchange control information between each other (e.g., PPStream periodically sends control packets between two connected peers [16]). Likewise, variable Δ_j^s can be transmitted from a media server to peer p_j^o in the control packets sent from the media server to peer p_j^o, and then from peer p_j^o to peer p_i^o in the control packets exchanged between these two peers. Variable Δ_c is transmitted from ISPs to peer p_i^o. Recent studies on ISP-friendly P2P traffic control, such as P4P [2], propose to use ISP portals (i.e. special servers run by ISPs) to provide information to peers. For example, in P4P, ISP portals can provide peers with the information about financial cost, ISP policy (e.g., avoiding using certain backbone links during peak time), etc. If P4P is employed, Δ_c can be transmitted from ISPs to peers through ISP portals.

Another concern about the implementation of the distributed algorithm is how to deal with peer join and peer leave. Upon peer p_j^o joins a streaming overlay and receives a set of peers from its tracker to communicate with, each upstream peer can allocate a very small amount of upload bandwidth to peer p_j^o (avoiding a 0 total download rate as the input to the utility function). Since the total download rate of peer p_j^o is small, variable Δ_j^u is large, which motivates its upstream peers to increase their connection rates to the peer. Similarly, after peer p_j^o establishes a connection to a new upstream peer during its lifetime, the new upstream peer can allocate a very small amount of upload bandwidth to peer p_j^o, the connection rate from the upstream peer to peer p_j^o will then be determined according to Eqs. (19) and (20). Upon peer p_j^o leaving a streaming overlay, the freed upload bandwidth will lower variable Δ_h, which motivates the upstream peers to increase their connection rates to other peers. Thus, the utilization of upload bandwidth can be maintained as high as possible.

6. Evaluation

In this section, we provide the evaluation of *GA* and compare it with *WS* and the even allocation *EA*. The evaluation is based on Matlab optimization toolbox and does not consider the dynamics of the distributed algorithm. That is, we solve problem *GA* directly using the Goal Attainment method provided by the optimization toolbox. Also, the evaluation is at the flow level as opposed to the packet level, which means we do not consider data availability, data scheduling and delays in packet transmission. We evaluate *GA*, *WS*, and *EA* under both the random peer selection and the locality-based peer selection with the following common simulation setup. We consider a network of four ASs as shown in Fig. 1.

AS 1 is a transport AS that provides interconnection service to the other 3 ASs. AS 2, AS 3, and AS 4 are normal ASs that provide the Internet access service to their users. AS 2 and AS 3 have peering agreement so that the P2P traffic between them does not incur any cost. However, the P2P traffic between AS 2 and AS 4, and between AS 3 and AS4 incurs the cost $c(x^o_{ij}) = c \times x^o_{ij}$ and $c(x^o i0) = c \times x^o i0$, $c > 0$ for the cross-ISP traffic x^o_{ij} and x^o_{i0}, respectively.

6.1. The tradeoffs among the conflicting design goals

In this subsection, we demonstrate that *GA* can explore the tradeoff space of the conflicting design goals. We consider the tradeoffs between the cross-ISP P2P traffic and the media server load, and the tradeoffs between the cross-ISP P2P traffic and the cross-ISP prefetching. In both cases, *WS* only achieves on–off style tradeoffs: either the cross-ISP P2P traffic is minimized; or the media server load is minimized; or the cross-ISP prefetching is maximized.

6.1.1. Example 1, the tradeoffs between the cross-ISP P2P traffic and the media server load

To show the tradeoffs between the cross-ISP P2P traffic and the media server load, we suppose that one streaming overlay runs in the network with 5 peers in each of AS 2, AS 3, and AS 4. Each peer connects to five other peers either randomly or according to a locality degree of 0.8. The media server is in AS 4. The required streaming rate is 300 Kbps. The nodes in AS 2 and AS 3 have upload bandwidth of 400 Kbps, and the nodes in AS 4 have upload bandwidth of 200 Kbps. We set the parameters of *GA* as follows. We set the expectation of the total connection cost to 300 and set the expectation of the media server load to 750 Kbps. We set the weights in Constraint (10) and the weight in Constraint (8) to 17.12 and 750, respectively, and vary the weight in Constraint (9) from 0 to 600.

In this simulation setup, the peers in AS 4 can not maintain smooth playback by just using the local P2P traffic in the same AS. They have to rely on both the media server and the P2P traffic from other ASs to ensue smooth playback. If we allow bigger cross-ISP P2P traffic, the load on the media server is smaller; on the contrary, if we allow smaller cross-ISP P2P traffic, the load on the media server is bigger.

Fig. 2 shows the media server load and the cross-ISP P2P traffic at the different weights in Constraint (9) for the random and the locality-based peer selection. By varying the weights, *GA* gradually increases the cross-ISP P2P traffic and decreases the media server load, which shows its ability to achieve the tradeoffs between these two. We also evaluate *WS* with $a = 1$, $b = 0.5$ and a varying c in the cost function from 0.2 to 2.0. It only achieves two tradeoff points: when $c < 0.5$, the cross-ISP P2P traffic is eliminated, the media server load is 900 Kbps for the random peer selection and 500 Kbps for the locality-based peer selection; when $c > 0.5$, the media server load becomes 0 Kbps, the cross-ISP P2P traffic is 700 Kbps for the random peer selection and 500 Kbps for the locality-based peer selection.

6.1.2. Example 2, the tradeoffs between the cross-ISP P2P traffic and the cross-ISP prefetching

To show the tradeoffs between the cross-ISP P2P traffic and the cross-ISP prefetching, we modify the parameters in Example 1. We set the required streaming rate to 100 Kbps and place an upper bound of 80 Kbps for each connection so that there is upload bandwidth available for the cross-ISP prefetching. Furthermore, we set the expectation of the total connection cost to 100 and vary its weight from 0 to 1.0.

In this simulation setup, all the peers can maintain smooth playback by just using the local P2P traffic. The

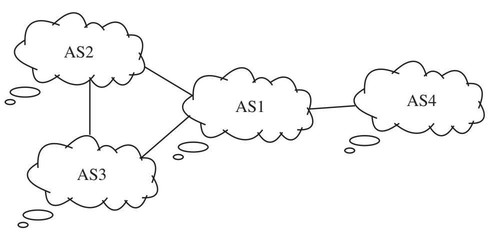
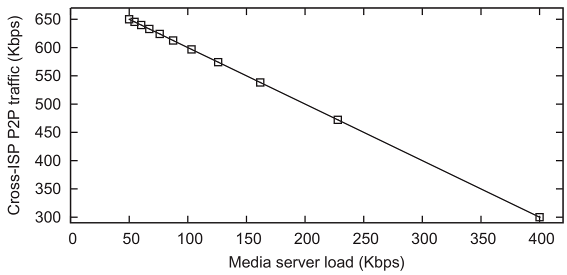
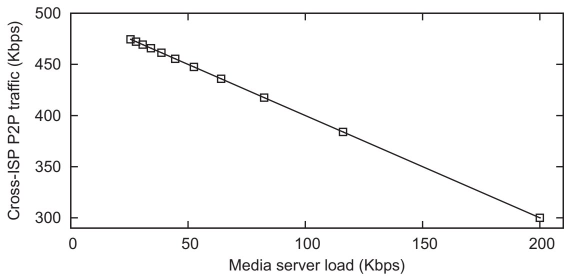


Fig. 1. Network topology.



(a) *GA* achieves the tradeoffs between the cross-ISP P2P traffic and the media server load under the random peer selection.



(b) *GA* achieves the tradeoffs between the cross-ISP P2P traffic and the media server load under the locality-based peer selection.

Fig. 2. The tradeoffs between the cross-ISP P2P traffic and the media server load.

cross-ISP P2P traffic is used to prefetch the video content. If we allow more cross-ISP P2P traffic, the peers can prefetch more video content and are more resistant to the peer churn; on the contrary, if we allow smaller cross-ISP P2P traffic, the peers will prefetch less video content and are more susceptible to the peer churn.

Fig. 3 shows the cross-ISP P2P traffic and the cross-ISP prefetching at different weights for the random and locality-based peer selection (we do not show the point at weight 0.4 for the random peer selection and the point at weight 0.9 for the locality-based peer selection, since they do not converge to the optimal solution). We can see that by varying the weight from 0 to 1.0, *GA* allows more and more cross-ISP P2P traffic and cross-ISP prefetching. We also evaluate *WS* with $a = 2$, $b = 1$ and a varying c from 0.1 to 2.1. Again, it can only achieve two tradeoff points: when $c < 1.0$, both the cross-ISP P2P traffic and the cross-ISP prefetching are 1880 Kbps for the random peer selection and 320 Kbps for the locality-based peer selection; when $c > 1.0$, both of them are 220 Kbps for the random peer selection and 0 Kbps for the locality-based peer selection.

6.1.3. Example 3, expectation enforcement

In addition to finer tradeoff tuning, *GA* can also enforce important design goals to meet their expectations, if possible. Suppose that the ISPs want to place a hard limit, say, 100, on the total connection cost in Example 1. We set the expectation of the total connection cost to 100 and set the corresponding weight to 0. We vary the number of the peers in AS 4 from 5 to 10.

In this simulation setup, the peers in AS 4 do not have enough upload bandwidth and have to rely on the media server and the cross-ISP P2P traffic to maintain smooth playback. As we increase the number of the peer in AS 4, we can expect an increase in the cross-ISP P2P traffic.

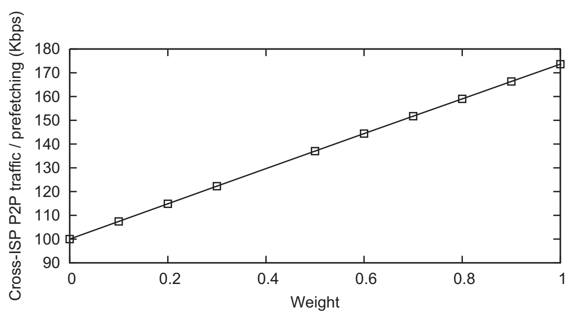
We plot the total connection cost in Fig. 4 for the random and locality-based peer selection (the fluctuation of the cross-ISP P2P traffic for *WS* at in Fig. 4 is due to the fact that *WS* converges to the solutions with more localized traffic). As a comparison, we also plot the result of *WS* with $a = 2$, $b = c = 1$. As the number of the peers in AS 4 increases, the number of the cross-ISP connections also increases. Despite that, *GA* is still able to maintain the total connection cost under the expectation. However, *WS* can not meet the expectation when facing increasing number of the cross-ISP connections. Placing a hard limit on the total connection cost is helpful in some cases, such as the cross-ISP routes are extremely congested. However, it can not adapt to the peer dynamics and should be used with caution.

6.2. Fairness

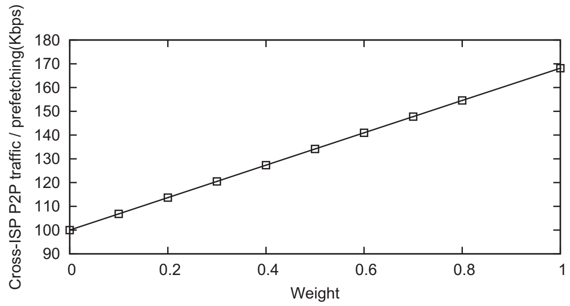
In this subsection, we demonstrate that our proposal is able to maintain good fairness among peers and streaming overlays.

6.2.1. Example 4, the fairness among the peers in the same streaming overlay

We use the same simulation setup as the one in Example 1. However, to minimize the impact of the connection cost and the media server load on the rate allocation, we

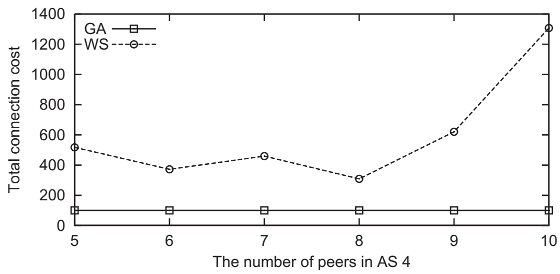


(a) *GA* achieves tradeoffs between the cross-ISP P2P traffic and the cross-ISP prefetching under the random peer selection.

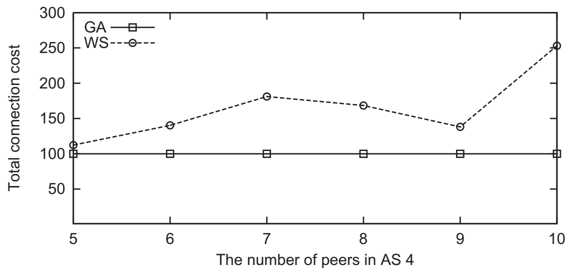


(b) *GA* achieves tradeoffs between the cross-ISP P2P traffic and the cross-ISP prefetching under the locality-based peer selection.

Fig. 3. The tradeoffs between the cross-ISP P2P traffic and the cross-ISP prefetching.

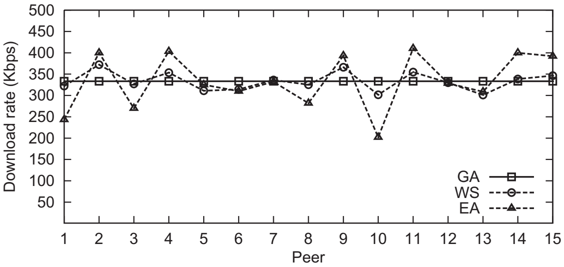


(a) GA enforces the expectation of the total connection cost under the random peer selection.

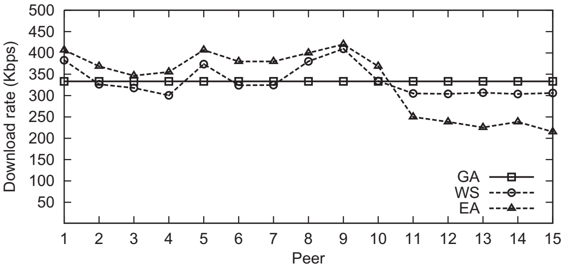


(b) GA enforces the expectation of the total connection cost under the locality-based peer selection.

Fig. 4. Expectation enforcement.



(a) *GA* allocates the upload bandwidth to the peers to meet their required streaming rate under the random peer selection.



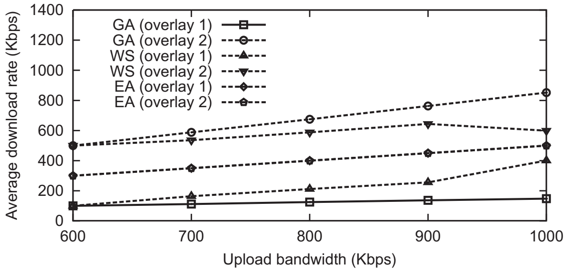
(b) *GA* allocates the upload bandwidth to the peers to meet their required streaming rate under the locality-based peer selection.

Fig. 5. The fairness among the peers in the same streaming overlay.

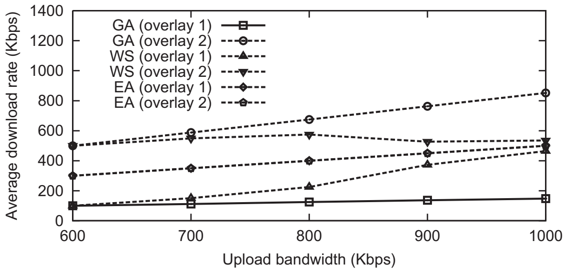
do not impose any cost on the cross-ISP P2P traffic, and assume that the media server offers enough bandwidth to assist the peers. We plot the download rate received by each peer in Fig. 5 for random and locality-based peer selection. We can see that both *GA* and *WS* can maintain the download rate of every peer above the required streaming rate, whereas *EA*, being equal to every peer, is unable to maintain the download rates of some peers above the required streaming rate.

6.2.2. Example 5, the fairness among the streaming overlays

To demonstrate fairness among the streaming overlays, we suppose that two streaming overlays run in the network. Every overlay has 5 peers in each of AS 2, AS 3, and AS 4, and each peer connects to five other peers randomly chosen or according to a locality degree of 0.8. The media servers of both overlays are in AS 1. The required streaming rates for overlay 1 and overlay 2 are 100 Kbps and 500 Kbps, respectively. In order to show the competition between the two overlays, we assume that the nodes in the network always join the two overlays simultaneously, and the two overlays have the same topology. We set the weight in Constraint (10) to 4.16 and 31.08 for the first overlay and second overlay, respectively, and keep the assumptions about the cross-ISP P2P traffic and the media servers from Example 4. We vary the upload bandwidth of all nodes from 600 Kbps to 1000 Kbps and plot the average download rates of the two overlays in Fig. 6. We can see that both *GA* and *WS* meet the required streaming rates for both overlays. However, *EA* over-provisions overlay 1 and leaves the required streaming rate of overlay 2 unsatisfied.



(a) *GA* allocates the upload bandwidth to multiple streaming overlays according to their required streaming rates under the random peer selection.



(b) *GA* allocates the upload bandwidth to multiple streaming overlays according to their required streaming rates under the locality-based peer selection.

Fig. 6. The fairness among the streaming overlays.

7. Conclusion

In this paper, we propose an upload bandwidth allocation mechanism for P2P assisted video-on-demand streaming with the following design goals: minimizing the media server load, maximizing the upload bandwidth utilization, minimizing the cross-ISP P2P traffic, and maintaining the fairness among the peers in the same or different streaming overlays. The first three design goals may conflict with one another. To this end, the mechanism employs the Goal Attainment method to optimize the first three design goals simultaneously and achieve the tradeoffs among them. Furthermore, it uses logarithm utility functions to guide the upload bandwidth allocation so that fairness can be maintained. Based on the Goal Attainment method, we further propose a distributed algorithm run by the peers, the media servers and the ISPs to solve the goal attainment method. We evaluate the mechanism extensively. The simulation results show that, compared with the related work like [7], our mechanism achieves finer tradeoff tuning. In addition, it can maintain good fairness among the peers in the same or different streaming overlays.

References

[1] P. Yang, L. Xu, On tradeoffs between cross-ISP P2P traffic and P2P streaming performance, in: Proceedings of IEEE GLOBECOM, Miaim, FL, 2010.

[2] H. Xie, Y.R. Yang, A. Krishnamurthy, Y. Liu, A. Silberschatz, P4P: provider portal for applications, in: Proceedings of ACM SIGCOMM, Seattle, WA, 2008.

[3] D.R. Choffnes, F.E. Bustamante, Taming the torrent: a practical approach to reducing cross-ISP traffic in peer-to-peer systems, ACM

SIGCOMM Computer Communication Review 38 (4) (2008) 363–374.
[4] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, 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.
[5] W. Li, S. Chen, T. Yu, UTAPS: An underlying topology-aware peer selection algorithm in BitTorrent, in: International Conference on Advanced Information Networking and Applications, Gino-wan City, Okinawa, Japan, 2008, pp. 539–545.
[6] F. Gembicki, Y. Haimes, Approach to performance and sensitivity multiobjective optimization: the goal attainment method, IEEE Transactions on Automatic Control 20 (6) (1975) 769–771.
[7] J. Wang, C. Huang, J. Li, On ISP-friendly rate allocation for peer-assisted VoD, Proceeding of the 16th ACM international conference on Multimedia, Vancouver, British Columbia, Canada, 2008, pp. 279–288.
[8] Y. Liu, H. Zhang, W. Gong, D. Towsley, On the interaction between overlay routing and underlay routing, in: Proceedings of IEEE INFOCOM, vol. 4, Miami, FL, 2005, pp. 2543– 2553.
[9] C. Wu, B. Li, Diverse: application-layer service differentiation in peer-to-peer communications, IEEE Journal on Selected Areas in Communications 25 (1) (2007) 222–234, doi:10.1109/JSAC.2007.070122.
[10] C. Wu, B. Li, Z. Li, Dynamic bandwidth auctions in multi-overlay P2P streaming with network coding, IEEE Transactions on Parallel and Distributed Systems 19 (6) (2008) 806–820.
[11] M. Wang, L. Xu, B. Ramamurthy, A flexible divide-and-conquer protocol for multi-view peer-to-peer live streaming, in: Proceedings of IEEE P2P, Seattle, WA, 2009.
[12] F. Kelly, Charging and rate control for elastic traffic, European Transaction on Telecommunications 8 (1) (1997) 33–37.
[13] J. Mo, J. Walrand, Fair end-to-end window-based congestion control, IEEE/ACM Transactions on Networking 8 (5) (2000).
[14] S. Buchegger, J.-Y.L. Boudec, A robust reputation system for p2p and mobile ad-hoc networks, in: Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems, Berkeley, CA, 2003.
[15] D. Grolimund, L. Meisser, S. Schmid, R. Wattenhofer, Havelaar: a robust and efficient reputation system for active peer-to-peer systems, in: Proceedings of 1st Workshop on the Economics of Networked Systems, Ann Arbor, MI, 2006.
[16] X. Su, L. Chang, A measurement study of ppstream, in: Proceedings of the 3rd International Conference on Communications and Networking in China, Hangzhou, China, 2008, pp. 1162–1166.

Peng Yang has received his Bachelor and Master degree in Central China Normal University, PR China and now is pursuing his Doctoral degree in the University of Nebraska-Lincoln, U.S.A. His research interests include wireless communication, transmission control protocols, utility-based rate control,and peer-to-peer networks.

Lisong Xu received his B.E. and M.E. degrees in Computer Science from the University of Science and Technology Beijing in 1994 and 1997, respectively. He completed his thesis "Performance Analysis Of Optical Burst Switched Networks" under the direction of Dr. Harry Perros and Dr. George Rouskas, and received his Ph.D. degree in Computer Science from North Carolina State University in 2002. From 2002 to 2004, he was a Post-Doctoral Research Fellow at North Carolina State University, working on Congestion Control for High-Speed Long-Distance Networks (TCP-BIC/CUBIC) with Dr. Injong Rhee, and on PhysViz Project with Dr. James Lester. He is currently an associate professor in Computer Science and Engineering at the University of Nebraska-Lincoln. He is a recipient of NSF CAREER Award (2007), and UNL CSE Student Choice Outstanding Teaching Award (2006, 2007, 2008, 2010).