Dynamic Bandwidth Auctions in Multi-overlay P2P Streaming with Network Coding

Chuan Wu, *Student Member, IEEE*, Baochun Li, *Senior Member, IEEE*
Department of Electrical and Computer Engineering
University of Toronto
{*chuanwu, bli*}*@eecg.toronto.edu*

Zongpeng Li, *Member, IEEE*
Department of Computer Science
University of Calgary
{*zongpeng*}*@ucalgary.ca*

***Abstract*—In peer-to-peer (P2P) live streaming applications such as IPTV, it is natural to accommodate multiple coexisting streaming overlays, corresponding to channels of programming. In the case of multiple overlays, it is a challenging task to design an appropriate bandwidth allocation protocol, such that these overlays efficiently share the available upload bandwidth on peers, media content is efficiently distributed to achieve the required streaming rate, as well as the streaming costs are minimized. In this paper, we seek to design simple, effective and decentralized strategies to resolve conflicts among coexisting streaming overlays in their bandwidth competition, and combine such strategies with network coding based media distribution to achieve efficient multi-overlay streaming. Since such strategies of conflict are game theoretic in nature, we characterize them as a decentralized collection of dynamic auction games, in which downstream peers bid for upload bandwidth at the upstream peers for the delivery of coded media blocks. With extensive theoretical analysis and performance evaluation, we show that these local games converge to an optimal topology for each overlay in realistic asynchronous environments. Together with network coding based media dissemination, these streaming overlays adapt to peer dynamics, fairly share peer upload bandwidth to achieve satisfactory streaming rates, and can be prioritized.**

***Index Terms*—Distributed networks, distributed applications, peer-to-peer streaming, bandwidth auction, multiple overlays**

I. Introduction

Peer-to-peer streaming applications have recently become a reality in the Internet [1], [2], [3], in which large numbers of peers self-organize into streaming overlays. It is natural to consider multiple coexisting streaming overlays (sessions) in such applications, each of which corresponds to a channel of television programming or live events. Generated with a modern codec such as H.264, each overlay distributes a live media stream with a specific streaming rate, such as 800 Kbps for a Standard-Definition stream and 1700 Kbps for a 480p (848×480 pixels) High-Definition stream. To meet such exacting *demands* of bandwidth that have to be satisfied at all participating peers, a streaming overlay relies on available upload bandwidth *supplies* of both dedicated streaming servers and regular participating peers. Smooth streaming playback is not possible unless such *supplies* meet the demand for streaming bandwidth, and efficient media distribution scheme is applied.

This work was supported in part by Bell Canada through its Bell University Laboratories R&D program.

It only becomes more challenging when coexisting streaming overlays are considered, sharing the available upload bandwidth in the peer-to-peer network. Consider a typical scenario where multiple peers from different overlays are in conflict with one another, competing for limited upload bandwidth at the same streaming server or upstream peer in the network. Apparently, the allocation of such upload bandwidth needs to be meticulously mediated with appropriate strategies, and media content needs to be efficiently distributed, such that the streaming rate requirement of each overlay is satisfied at all participating peers. It would be best if, at the same time, fairness can be achieved across different overlays, and costs of streaming (*e.g.,* latencies) can be minimized. It goes without saying that if such tactical strategies are not implemented, the conflict among streaming overlays may not be resolved satisfactorily.

In this paper, we seek to design simple, decentralized, but nonetheless effective tactical strategies to resolve inherent bandwidth conflicts among coexisting streaming overlays, and utilize network coding to achieve efficient media content distribution. For this purpose, we characterize the bandwidth conflicts in a game theoretic setting, with *dynamic auction games*. Such games evolve over time, and involve repeated *auctions* in which competing downstream peers from different overlays *bid* for upload bandwidth at the same upstream peer, for the streaming of media blocks coded with network coding. In these dynamic auction games, an upstream peer allocates its upload bandwidth based on bids from downstream peers, and a downstream peer may optimize and place its bids to multiple upstream peers that have innovative coded blocks it desires, and subsequently compete in multiple auctions. Each of these auctions is locally administered, and leads to cleanly decentralized strategies.

With extensive theoretical analysis and performance evaluation using simulations, we show that these decentralized game-theoretic strategies not only converge to a Nash equilibrium, but also lead to favorable outcomes, in realistic asynchronous environments: we are able to obtain an optimal topology for each coexisting streaming overlay, in the sense that streaming rates are satisfied, and streaming costs are minimized. These topologies of coexisting overlays evolve and adapt to peer dynamics, fairly share peer upload bandwidth, and can be prioritized. In contrast to existing game theoretic approaches that are largely theoretical in nature, we show that our proposed strategies can be practically implemented in realistic

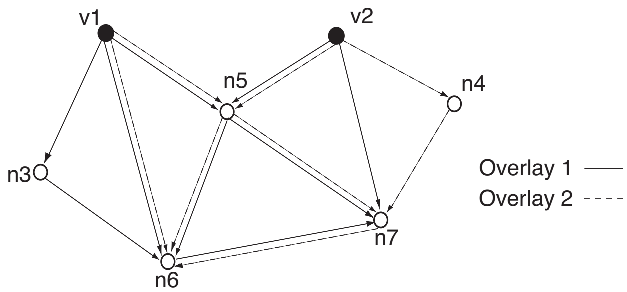


Fig. 1. Two concurrent P2P streaming overlays: an example.

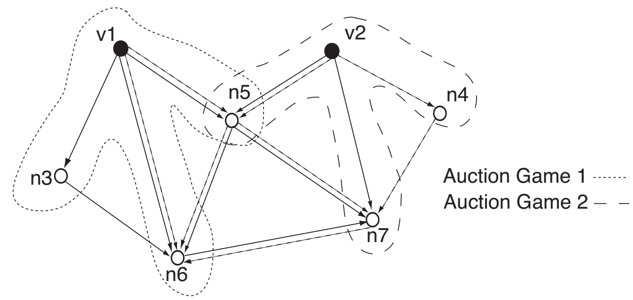


Fig. 2. Decentralized auction games in the example overlays.

streaming overlays, and can be seamlessly integrated with efficient media distribution based on network coding to produce a complete conflict-resolving multi-overlay streaming protocol design. Indeed, our focus in this paper is not on reasoning about the rationality and selfishness of peers, nor on incentive engineering to encourage contribution. We seek to devise practical strategies that may be realistically implemented, and use game theoretic tools only to facilitate the design of such conflict-resolving strategies.

The remainder of this paper is organized as follows. In Sec. II, we present our system model and motivate the design of distributed auction games for resolving bandwidth conflicts. In Sec. III, we discuss the bidding and allocation strategies, and establish their equilibrium using game theory techniques. In Sec. IV, their convergence to the optimal bandwidth allocation is analyzed in both asynchronous and dynamic environments, and their practical implementation with network coding based media distribution is discussed. Sec. V is dedicated to an in-depth study of the proposed strategies in realistic settings, with respect to interactions of multiple dynamic streaming overlays. We then discuss related work and conclude the paper in Sec. VI and Sec. VII, respectively.

II. Multi-overlay Streaming Model

A. Network model and assumptions

This paper considers a P2P live streaming network including multiple coexisting streaming overlays, each consisting of streaming servers and participating peers. Each server may serve more than one overlay, while each peer may also participate in multiple overlays [1]. Fig. 1 shows an example of two coexisting streaming overlays, each with two streaming servers and four participating peers.

In each streaming overlay, participating servers and peers form a mesh topology, in which any peer is served by its *upstream* peers (servers can be deemed as special upstream peers), and may serve one or more *downstream* peers at the same time. The peers server each other by exchanging segments of media data in the streaming channel, which are received and cached in their local playback buffers. The playback buffer at each peer represents a sliding window of the streaming channel, and contains segments to be played in the immediate future.

With respect to the construction of each overlay, we consider there exists a standalone neighbor list maintenance mechanism in the network, consisting of one or multiple bootstrapping servers. The mechanism assigns a certain number of existing peers in an overlay as neighbors to each new peer upon its joining, and maintains the number of neighbors for each peer during streaming upon peer dynamics. The application-layer links between peers in each overlay are established based on their media content availability during streaming.

Let $\mathcal{S}$ denote the set of all coexisting streaming overlays in the network. The topology of each overlay $s \in \mathcal{S}$ can be modeled as a directed graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{N}_s, \mathcal{A}_s)$, where $\mathcal{V}_s$ is the set of servers serving overlay s, $\mathcal{N}_s$ represents the set of participating peers, and $\mathcal{A}_s$ denotes the set of application-layer links in overlay s. Let R_s be the required streaming rate of the media stream distributed in overlay s. Let $\mathcal{V}$ be the set of all streaming servers in the network, *i.e.*, $\mathcal{V} = \cup_{s\in\mathcal{S}}\mathcal{V}_s$, and $\mathcal{N}$ be the set of all existing peers, *i.e.*, $\mathcal{N} = \cup_{s\in\mathcal{S}}\mathcal{N}_s$. Let U_i denotes the upload bandwidth at peer i, $\forall i \in \mathcal{V} \cup \mathcal{N}$.

Realistically, we assume that the last-mile upload bandwidth on each peer (including servers) constitutes the "supply" of bandwidth in the overlays, *i.e.*, bandwidth bottlenecks lie at the peers rather than at the core of the overlays. In addition, we assume that the download bandwidth of each peer is sufficient to support the required streaming rate(s) of the overlay(s) it participates in. This represents a practical scenario, as peers with insufficient download bandwidth for an overlay will soon quit the overlay and may join another overlay with lower streaming rate requirement.

B. Auction game model

To resolve the bandwidth conflict on its upload link, each upstream peer i in the network, $\forall i \in \mathcal{V} \cup \mathcal{N}$, organizes a *dynamic bandwidth auction game*, referred to as auction i. In auction i, the "goods" for sale is the upload bandwidth of peer i with a total quantity of U_i, and the players are all the downstream peers of peer i in all overlays it participates in. Let j^s represent peer j in overlay s. The set of players in auction i can be expressed as $\{j^s, \forall j : (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}\}$ [2]. As each peer in an overlay may stream from multiple upstream peers in the overlay, a player j^s may concurrently bid for upload bandwidth in multiple auction games, each hosted by one upstream peer.

The auction games at the peers are dynamically carried out in a repeated fashion to resolve bandwidth conflicts over time. In each *bidding round* of auction i, each player submits its bid to peer i, declaring its requested share of upload bandwidth, as well as the unit price it is willing to pay. The upstream peer i

[1]A practical scenario of this case is that users behind a same gateway may watch different channels, while they appear as the same identity (peer) in the Internet with one same external IP address.

[2]Note that in case a downstream peer j participates in multiple overlays, it is viewed as multiple players, each for one overlay.

then allocates shares of its upload capacity, U_i, to the players based on their bids. Let x_{ij}^s denote the upload bandwidth that player j^s requests from peer i, and p_{ij}^s denote the unit price it is willing to pay to peer i. The bid from player j^s in auction i can be represented as a 2-tuple $b_{ij}^s = (p_{ij}^s, x_{ij}^s)$.

Such a distributed game model can be illustrated with the example in Fig. 2. In the example, there exist 7 auction games, two of which are marked: auction 1 at v_1 with 5 players 3^1 (peer $n3$ in overlay 1), 5^1, 5^2, 6^1 and 6^2, auction 2 at v_2, with 4 players 4^2, 5^1, 5^2 and 7^1, respectively.

Seamlessly integrated with the distributed bandwidth auctions, a dissemination scheme based on the state-of-the-art framework of streaming with network coding ([4], [5], [6]) is employed to distribute coded media content among the peers in each overlay. Based on network coding, a downstream peer in each overlay bids in the auction at one upstream peer only when the upstream peer can supply it with innovative coded blocks for the overlay, and the allocated bandwidth in the auctions is efficiently utilized to deliver such coded blocks. We leave the related detailed discussions to Sec. IV-C.

III. The Bandwidth Auction Game

In this section, we present the auction strategies to resolve bandwidth conflicts in multi-overlay streaming, including the allocation strategy taken by an upstream peer and the bidding strategy by downstream peers, and establish the equilibrium of the distributed auctions from the game theoretical point of view.

A. Allocation strategy

In auction i, the seller, upstream peer i, aims to maximize its revenue by selling its upload bandwidth U_i at the best prices. Given bids $b_{ij}^s = (p_{ij}^s, x_{ij}^s)$'s from all the players j^s ($\forall j : (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$), upstream peer i's allocation strategy can be represented by the following revenue maximization problem. Here, a_{ij}^s ($\forall j : (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$) is the bandwidth share to be allocated to each downstream peer j in each competing overlay s.

Allocation i:

$$\max \sum_{s\in\mathcal{S}} \sum_{j:(i,j)\in\mathcal{A}_s} p_{ij}^s a_{ij}^s \qquad (1)$$

subject to

$$\sum_{s\in\mathcal{S}} \sum_{j:(i,j)\in\mathcal{A}_s} a_{ij}^s \leq U_i,$$
$$0 \leq a_{ij}^s \leq x_{ij}^s, \qquad \forall j : (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}.$$

Such an allocation strategy can be achieved in the following fashion:

Upstream peer i selects the highest bid price, e.g., p_{ij}^s *from player j^s, and allocates bandwidth $a_{ij}^s = \min(U_i, x_{ij}^s)$ to it. Then if it still has remaining bandwidth, it selects the second highest bid price and assigns the requested bandwidth to the corresponding player. This process repeats until peer i has allocated all its upload capacity, or bandwidth requests from all the players have been satisfied.* □

The above allocation strategy can be formally stated in the following formula:

$$a_{ij}^s = \min(x_{ij}^s, U_i - \sum_{p_{ik}^{s'} \geq p_{ij}^s, k^{s'} \neq j^s} a_{ik}^{s'}),$$
$$\forall j : (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}. \qquad (2)$$

B. Bidding strategy

In each overlay $s \in \mathcal{S}$, a peer j may place its bids to multiple upstream peers, that can supply innovative coded blocks to it. As a common objective, it wishes to achieve the required streaming rate for the overlay, and experiences minimum costs. We consider two parts of costs when peer j streams from peer i in overlay s: streaming cost — denoted by streaming cost function $D_{ij}^s(x_{ij}^s)$ — represents the streaming latency actually experienced by j; bidding cost — calculated by $p_{ij}^s x_{ij}^s$ — represents the bid peer j submits to peer i in overlay s. The bidding cost reflects the degree of *competition* and *demand* for bandwidth in the auctions at upstream peers. The overall cost at player j^s is the sum of the two parts from all its upstream peers, $\forall i : (i,j) \in \mathcal{A}_s$.

In this way, the preference for player j^s in deciding its bids in the auctions can be expressed by the following cost minimization problem. Practically, each cost function D_{ij}^s should be non-decreasing and its value increases more rapidly when the requested bandwidth x_{ij}^s is larger (*i.e.* the property of convexity). Therefore, without loss of generality, we assume the cost functions are non-decreasing, twice differentiable and strictly convex.

Bidding $\mathbf{j^s}$:

$$\min \sum_{i:(i,j)\in\mathcal{A}_s} (D_{ij}^s(x_{ij}^s) + p_{ij}^s x_{ij}^s) \qquad (3)$$

subject to

$$\sum_{i:(i,j)\in\mathcal{A}_s} x_{ij}^s \geq R_s, \qquad (4)$$
$$x_{ij}^s \geq 0, \qquad \forall i : (i,j) \in \mathcal{A}_s. \qquad (5)$$

The bidding strategy of player j^s consists of two main components: bandwidth requests and price adjustments.

1) Bandwidth requests: If the bid prices p_{ij}^s's are given, the requested bandwidths at player j^s towards each of its upstream peers in overlay s, *i.e.*, $x_{ij}^s, \forall i : (i,j) \in \mathcal{A}_s$, can be optimally decided by solving the problem **Bidding $\mathbf{j^s}$**. This can be done efficiently with a water-filling approach, in which player j^s acquires the required streaming rate R_s by requesting from upstream peers that incur minimum marginal costs:

Let $f_j^s(x)$ denote the overall cost at player j^s, i.e., $f_j^s(x) = \sum_{i:(i,j)\in\mathcal{A}_s}(D_{ij}^s(x_{ij}^s) + p_{ij}^s x_{ij}^s)$. *The marginal cost with respect to x_{ij}^s is $\frac{df_j^s(x)}{dx_{ij}^s} = D_{ij}^{'s}(x_{ij}^s) + p_{ij}^s$. Beginning with $x_{ij}^s = 0$ ($\forall i : (i,j) \in \mathcal{A}_s$), the player identifies one x_{ij}^s that achieves the smallest marginal cost and increases the value of this x_{ij}^s. As $D_{ij}^s(x_{ij}^s)$ is strictly convex, $D_{ij}^{'s}(x_{ij}^s)$ increases with the increase of x_{ij}^s. The player increases this x_{ij}^s until its marginal cost is no longer the smallest. Then it finds a new x_{ij}^s with the current smallest marginal cost and increases its value. This process repeats until the sum of all x_{ij}^s's ($\forall i : (i,j) \in \mathcal{A}_s$) reaches R_s.* □

The water-filling approach can be illustrated in Fig. 3, in which the height of each bin represents the marginal cost for

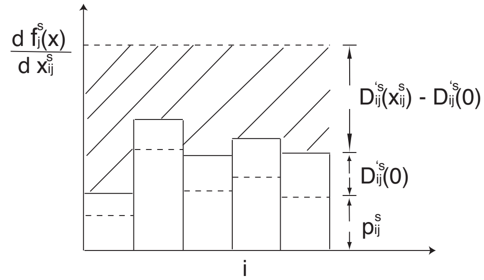


Fig. 3. Bandwidth requesting strategy at player j^s: an illustration of the water-filling approach.

player j^s to stream from each upstream peer i. To fill water at a total quantity of R_s into these bins, the bins with the lowest heights are flooded first, until all bins reach the same water level. Then the same water level keeps increasing until all the water has been filled in.

Theorem 1. *Given bid prices $p_{ij}^s, \forall i : (i,j) \in \mathcal{A}_s$, the water-filling approach obtains a unique optimal requested bandwidth assignment at player j^s,* i.e., *$(x_{ij}^{s*}, \forall i : (i,j) \in \mathcal{A}_s)$, which is the unique optimal solution to the problem* ***Bidding*** $\mathbf{j^s}$.

We postpone the proof of Theorem 1 to Appendix A.

2) Price adjustments: We next address how each player is to determine the bid price to each of its desirable upstream peers. A price adjustment scheme is designed for this purpose, by which each player tactically adjusts its prices in participating auctions based on whether its bandwidth requirement is achieved in the previous bidding round.

When a player j^s first joins an overlay, it initiates bid prices p_{ij}^s's towards all desired upstream peers to 0. Then it calculates the current optimal requested bandwidth assignment with the *water-filling approach*, and sends its bids to the upstream peers. After upstream peer i allocates its upload capacity with the *allocation strategy*, it sends allocated bandwidth values to corresponding players. Upon receiving an allocated bandwidth, player j^s increases the corresponding bid price if its "demand" is higher than the "supply" from the upstream peer, and otherwise decreases the price. Meanwhile, it recomputes its requested bandwidth assignment for all its upstream peers with the water-filling approach. Such price adjustment is carried out in an iterative fashion, until the player's bandwidth requests may all be granted at respective upstream peers if it is to bid the new prices in the next round.

Using the water-filling approach as a building block, the price adjustment scheme is summarized in the *bidding strategy* to be carried out by player j^s in each round of its participating auctions, as presented in Table I.

The intuition behind the bidding strategy is that, each player places different bid prices to different upstream peers, considering both the streaming cost and the overall demand at each upstream peer. If the streaming cost is low from an upstream peer, the player is willing to pay a higher price and strives to acquire more upload bandwidth from this peer. On the other hand, if the bandwidth competition at an upstream peer is intense such that the bidding cost becomes excessive, the player will forgo its price increases and request more bandwidths from other peers. At all times, the marginal cost of streaming from each upstream peer is kept the same, as achieved by the water-filling process.

TABLE I
BIDDING STRATEGY AT PLAYER j^s

Input
–(p_{ij}, x_{ij}): bids submitted in previous bidding round
–allocated bandwidth a_{ij}^s in previous bidding round from all upstream peers $i, \forall i : (i,j) \in \mathcal{A}_s$.

Adjust prices and bandwidth requests
Repeat
(a) For each upstream peer i
– If $x_{ij}^s > a_{ij}^s$, increase price p_{ij}^s by a small amount δ;
– If $x_{ij}^s \leq a_{ij}^s$ and $p_{ij}^s > 0$, decrease price p_{ij}^s by δ.
(b) Adjust requested bandwidth assignment $(x_{ij}^s, \forall i : (i,j) \in \mathcal{A}_s)$ with the water-filling approach.
(c) For each upstream peer i
– Calculate new allocation a_{ij}^s that can be acquired from i if the current price p_{ij}^s is bid, based on Eqn. (2), with queried bids of some other players in the previous round of auction i.
Until: all requested bandwidths x_{ij}^s's, are to be achieved with current prices p_{ij}^s's, *i.e.*, $x_{ij}^s \leq a_{ij}^s, \forall i : (i,j) \in \mathcal{A}_s$, and prices p_{ij}^s's are the lowest possible to achieve it.

Submit new bids
Send new bids $b_{ij}^s = (p_{ij}^s, x_{ij}^s), \forall i : (i,j) \in \mathcal{A}_s$, to respective upstream peers.

We note that to calculate the new achievable allocation a_{ij}^s, player j^s needs to know bids placed by some of its opponents in the previous bidding round in auction i. Instead of asking upstream peer i to send all received bids, player j^s can query such information gradually only when necessary. If p_{ij}^s is to be increased, it asks for the bid of opponent $m^{s'}$ whose price $p_{im}^{s'}$ is immediately higher than p_{ij}^s in auction i. While p_{ij}^s is still below $p_{im}^{s'}$, player j^s's achievable bandwidth is unchanged; only when p_{ij}^s exceeds $p_{im}^{s'}$, its achievable bandwidth is increased by $a_{im}^{s'}$, and player j^s queries upstream peer i again for the bid containing the immediately higher price than the current value of p_{ij}^s. Similar bid inquiries can be implemented for the case that p_{ij}^s is to be reduced. In this way, the price adjustments can be achieved practically with little messaging overhead.

C. Game Theoretical Analysis

The distributed auction games in the coexisting streaming overlays are carried out in a repeated fashion, as these are *dynamic games*. They are correlated with each other as each player optimally places its bids in multiple auctions. A critical question is: *Does there exist a stable "operating point" of the decentralized games, that achieves efficient partition of network upload bandwidths?* We now seek to answer this question with game theoretical analysis.

We consider upload bandwidth competition in the entire network as one *extended* dynamic non-cooperative strategic game (referred to as G_{ext}), containing all the distributed correlated auctions. The set of players in the extended game can be represented as

$$\mathcal{I} = \{j^s, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S}\}. \quad (6)$$

The action profile taken by player j^s is a vector of bids, in which each component is the bid to place to one upstream peer. Formally, the set of action profiles for player j^s is defined as

$$\Gamma_j^s = \{B_j^s | B_j^s = (b_{ij}^s, \forall i : (i,j) \in \mathcal{A}_s),$$
$$b_{ij}^s = (p_{ij}^s, x_{ij}^s) \in [0, +\infty) \times [0, R_s], \sum_{i:(i,j)\in\mathcal{A}_s} x_{ij}^s \geq R_s\}. \quad (7)$$

Then, let B denote the bid profile in the entire network, *i.e.*, $B = (B_j^s, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S}) \in \times_{j,s}\Gamma_j^s$. The preference relation $\succsim_j^s$ for player j^s can be defined by the following overall cost function, which is the objective function in the problem **Bidding j^s** in (3)

$$\text{Cost}_j^s(B) = \sum_{i:(i,j)\in\mathcal{A}_s} (D_{ij}^s(x_{ij}^s) + p_{ij}^s x_{ij}^s). \quad (8)$$

Therefore, we say two bid profiles $B \succsim_j^s B'$ if $\text{Cost}_j^s(B) \leq \text{Cost}_j^s(B')$.

Definition 1. *A bid profile B in the network, $B = (B_j^s, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S}) \in \times_{j,s}\Gamma_j^s$, is feasible if its bandwidth requests further satisfy upload capacity constraints at all the upstream peers, i.e., $\sum_{s\in\mathcal{S}}\sum_{j:(i,j)\in\mathcal{A}_s} x_{ij}^s \leq U_i, \forall i \in \mathcal{V}\cup\mathcal{N}$.*

When a bid profile is feasible, from the allocation strategy discussed in Sec. III-A, we can see the upload bandwidth allocations will be equal to the requested bandwidths.

Using $\widetilde{B_j^s}$ to represent action profiles of all players other than player j^s in $\mathcal{I}$, *i.e.*, $\widetilde{B_j^s} = (B_m^k, \forall m^k \in \mathcal{I} \setminus \{j^s\})$, we have the following definition of Nash equilibrium.

Definition 2. *A feasible bid profile $B^* = (B_j^{s*}, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S})$ is a Nash equilibrium of the extended game $G_{ext}\langle\mathcal{I}, (\Gamma_j^s), (\succsim_j^s)\rangle$ if for every player $j^s \in \mathcal{I}$, we have $\text{Cost}_j^s(B_j^{s*}, \widetilde{B_j^{s*}}) \leq \text{Cost}_j^s(B_j^{'s}, \widetilde{B_j^{s*}})$ for any other feasible bid profile $B' = (B_j^{'s}, \widetilde{B_j^{s*}})$.*

We next show the existence of a Nash equilibrium for the extended game. We focus on feasible streaming scenarios as stated in the following assumption:

Assumption 1. *The total upload bandwidth in the P2P network is sufficient to support all the peers in all overlays to stream at required rates,* i.e., *there exists a feasible bid profile in the P2P network.*

Theorem 2. *In the extended game $G_{ext}\langle\mathcal{I}, (\Gamma_j^s), (\succsim_j^s)\rangle$ in which distributed auctions are dynamically carried out with the allocation strategy in (2) and the bidding strategy in Table I, there exists a Nash equilibrium under Assumption 1.*

We postpone the proof of Theorem 2 to Appendix B.

The next theorem shows that at equilibrium, the upload bandwidth allocation in the network achieves the minimization of the global streaming cost.

Theorem 3. *At Nash equilibrium of the extended game $G_{ext}\langle\mathcal{I}, (\Gamma_j^s), (\succsim_j^s)\rangle$, upload bandwidth allocation in the network achieves streaming cost minimization, as achieved by the following global streaming cost minimization problem:*

$$\min \sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{N}_s}\sum_{i:(i,j)\in\mathcal{A}_s} D_{ij}^s(y_{ij}^s) \quad (9)$$

subject to

$$\sum_{s\in\mathcal{S}}\sum_{j:(i,j)\in\mathcal{A}_s} y_{ij}^s \leq U_i, \quad \forall i \in \mathcal{V}\cup\mathcal{N}, \quad (10)$$
$$\sum_{i:(i,j)\in\mathcal{A}_s} y_{ij}^s \geq R_s, \quad \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S}, \quad (11)$$
$$y_{ij}^s \geq 0, \quad \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S} \quad (12)$$

Theorem 3 can be proven by showing that the set of KKT conditions for the global streaming cost minimization problem is the same as that satisfied by the equilibrium bid profile $B^* = ((p_{ij}^{s*}, x_{ij}^{s*}), \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S})$, and the equilibrium bid prices at each upstream peer i (*i.e.*, $p_{ij}^{s*}, \forall j : (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$) have the same value as the Lagrangian multiplier associated with the upload capacity constraint (10) at peer i. We postpone the detailed proof of Theorem 3 to Appendix C. From the proof of Theorem 3, we can derive the following corollary:

Corollary. *At Nash equilibrium, the bid prices to each upstream peer i from all competing players that are allocated non-zero bandwidths are the same,* i.e., *$\exists t_i^*, p_{ij}^{s*} = t_i^*$ if $x_{ij}^{s*} > 0$, $\forall j : (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$.*

This corollary can also be intuitively illustrated: If a player in auction i is paying a price higher than some other player who is also allocated non-zero bandwidth, the former can always acquire more bandwidth from the latter with a price lower than its current price. Thus at equilibrium, when no one can unilaterally alter its price, all players must be paying the same price.

IV. Practical Application of Proposed Strategies

We next discuss the practical deployment of the auction strategies in realistic asynchronous and dynamic P2P streaming networks, show the convergence of the dynamic auctions to the Nash equilibrium, and discuss how such dynamic bandwidth allocation can be seamlessly integrated with network coding based media distribution.

A. Asynchronous play

In a practical P2P network, peers are inherently asynchronous with different processing speeds. Besides, with various message passing latencies, bids and allocated bandwidth updates may arrive at each upstream or downstream peer at different times. All these make each auction completely asynchronous. A practical deployment of the game theoretical strategies should be able to practically handle such asynchronous game play.

In our design, bids and allocation updates are passed by messages sent over TCP, such that their arrival is guaranteed. The strategies are to be carried out practically in each auction in the following fashion:

Allocation. At each upstream peer, starting from the last time it sends out allocation updates, the upstream peer collects new bids until it has received them from all its existing downstream peers in all the overlays it participates in, or a timeout value, T, has passed since the previous allocation, whichever is shorter. Then the upstream peer allocates its upload bandwidth again with the allocation strategy discussed in (2): for those downstream peers whose bids it has received, it uses the new bids; for those slow ones that it has not heard from in this round, it uses the most recent bids from them. Then the upstream peer sends allocation updates to all the downstream peers whose allocation has been changed, and starts a new round of execution.

Bidding. At each downstream peer in each streaming overlay, since its last bidding round, it waits for bandwidth allocation until all allocated bandwidth updates have arrived from all its requested upstream peers, or time T has passed since the last time it placed all the bids, whichever is shorter. Then it adjusts prices towards those upstream peers from which it has received allocation updates, retains its previous prices towards those it has not heard from in this round, and recalculates its new bids to all the upstream peers, using the bidding strategy in Table I. It then submits new bids, that are different from the ones submitted previously, to the respective upstream peers.

While we have established the existence of Nash equilibrium of the distributed auctions in Theorem 2, we have not yet addressed another critical question: *can the Nash equilibrium,* i.e., *the stable operating point that achieves optimal bandwidth allocation among peers in different overlays, be actually reached with such dynamic asynchronous play of the auction games?* We now seek to justify such a convergence, based on the following assumptions:

Assumption 2. *a) Each downstream peer in each overlay will communicate its new bids to its upstream peers within finite time (until it has acquired the required streaming bandwidth for the overlay at the lowest possible prices); b) Each upstream peer will communicate latest allocation updates to its downstream peers within finite time.*

Theorem 4. *Under Assumption 1 and 2, the asynchronous distributed auctions converge to the Nash equilibrium, whose existence is established in Theorem 2.*

Proof: We first note the following property of the extended auction game, as modeled in Sec. III-C:

Claim 1. The total allocated upload bandwidth in the entire network, $U_{alloc} = \sum_{s\in\mathcal{S}}\sum_{(i,j)\in\mathcal{A}_s} a_{ij}^s$, grows monotonically during the asynchronous play of the extended auction game.

The truth of the above claim lies in the fact that once a unit of upload bandwidth is allocated during the auction, it remains allocated throughout the rest of the game; *i.e.*, a unit of allocated bandwidth may switch its ownership from a downstream peer i in overlay s_1 to another downstream peer j in overlay s_2, but will never become idling again.

We further know U_{alloc} is bounded above by the total upload bandwidth in the network, $U_{all} = \sum_{i\in\mathcal{V}\cup\mathcal{N}} U_i$. Since U_{alloc} is increasing and upper-bounded, it therefore converges. Let U^*_{alloc} be its convergence value.

We now prove the theorem by contradiction. We assume that the extended auction game does not converge and runs for an infinitely long time. By Assumption 2, we know that there must exist peers that do not obtain sufficient bandwidth and thus bid infinitely often with updated prices. In this case, U^*_{alloc} must be smaller than the aggregated bandwidth demand at all peers in all the overlays, as otherwise peers stop bidding and the auction terminates.

When the total allocated bandwidth U^*_{alloc} is not sufficient to satisfy the bandwidth requirement at all the peers, based on our price adjustment strategy, we know the bid prices at all upstream peers will be growing unbounded. In this case, there must not exist an upstream peer that still has spare upload bandwidth, *i.e.*, all upload bandwidth in the network has been allocated. We thus derive $U^*_{alloc} = U_{all}$. Therefore, U_{all} is smaller than the total bandwidth demand at all the peers in all the overlays, which contradicts with assumption 1. Thus the extended auction game must converge. In addition, the extended auction game must converge to its Nash Equilibrium (since peers would otherwise continue bidding), which achieves streaming cost minimization based on Theorem 3. □

B. Peer dynamics

Peer asynchrony aside, the inherent dynamics of realistic P2P network further leads to dynamics of auction participants. The players in each auction may change dynamically, due to new peers joining the network, existing peers joining another overlay or switching upstream peers due to content availability, or peer failures and departures; a distributed auction may start or close due to the arrival or departure of an upstream peer. With slightly more extra effort, our asynchronous deployment can readily adapt to such dynamics.

At the downstream side, when a peer newly joins an auction at an upstream peer, *e.g.*, in the cases of arrival of a new downstream or upstream peer, it initializes its bid price to 0, computes requested bandwidth together with its prices to other upstream peers, and then forwards its bid to the upstream peer. In the case that one of its upstream peers fails or departs, the downstream peer can detect it based on the broken or closed connections. Then it may exclude the upstream peer from its bandwidth request calculation.

At the upstream side, when an upstream peer receives the bid from a new peer, it immediately incorporates the downstream peer into its bandwidth allocation. When it detects the failure or departure of a downstream peer based on the broken or closed connections, the upstream peer allocates upload bandwidth to the remaining peers only in a new round, excluding the departed peer from the auction game.

When peer dynamics are present in the network, the dynamic auction game progresses and strives to pursuit the optimal bandwidth allocation in the latest overlay topology. When peers continue to join and leave the overlays, the optimal bandwidth allocation naturally becomes a moving target. When such dynamics stop and overlay topology stabilizes, if the overall bandwidth supply is sufficient in the current topology, by results in Theorem 2, 3 and 4, we know there exists a Nash Equilibrium which achieves global streaming cost minimization, and the auction game converges to such an optimal streaming topology.

C. Combining with network coding

So far we have presented and analyzed the bandwidth auction strategies by assuming an existing mesh topology for each overlay as part of the input. We now discuss how such overlay meshes can be constructed and maintained in a P2P system, and how the bandwidths allocated during the auctions are utilized to stream media content. We present such input construction and output utilization within the state-of-the-art framework of streaming with network coding [4], [5], [6].

Network coding is a recent technique originated from information theory that allows encoding/decoding at every node across the network [7], [8]. It has proven to increase data diversity in a content distribution system, which facilitates peer reconciliation, enhances failure resilience, and therefore improves the overall system efficiency [4], [9]. Within the context of multi-overlay media streaming using network coding, for each overlay s, its media bitstream M_s is separated into segments $g_s^1, g_s^2, \ldots$, each corresponding to n playback seconds. Each segment g_s^i further consists of k equal-sized blocks, and bits in each block are viewed as a vector of q-bit symbols over the Galois Field $GF(2^q)$. Encoding operations are performed at both the servers and the upstream peers in each overlay s, by linearly combining multiple blocks of a segment in M_s in symbol-wise fashion over $GF(2^q)$. A downstream peer in the overlay may recover a segment by decoding from any k innovative (linearly independent) blocks received for that segment [4].

As presented in Sec. II, a bootstrapping server initializes and maintains the neighbor list at each peer in each overlay. Based on such initial connectivity, each peer periodically advertises block availability to its neighboring peers in the overlay. An advertisement consists of metadata that describes the segment number and linear coefficients of coded blocks available in that segment. Based on such block availability, each peer computes whether a neighbor can provide it with innovative coded blocks for a segment it desires, and how many new coded blocks it can serve [9]. Then the peer bids for bandwidth at the possible serving neighbors. In this way, the sets of overlay links involved in the auctions, *i.e.*, $\mathcal{A}_s, \forall s \in \mathcal{S}$, are clearly defined.

The upload bandwidth allocated to each downstream peer in each overlay during an auction is utilized to deliver new coded blocks produced by the upstream peer for the requested segment in the respective overlay. Once the transmission of a segment finishes, the allocated bandwidth can be used for sending innovative coded blocks of another segment desired by the downstream peer, if there exist. Otherwise the situation is similar to a peer (upstream) departure, and the downstream peer needs to bid elsewhere for additional bandwidth. Thanks to the data diversity generated by network coding, content scheduling policies such as *rarest first* are of less concern here, as network coding essentially eliminates the *last block problem* [9] and maximally saturates the allocated bandwidth with useful blocks.

V. Performance Evaluation

In this section, we conduct in-depth investigations of the proposed auction strategies in practical scenarios. Using simulations under real-world asynchronous settings, the focus of our investigation is to show that, as an outcome of our proposed strategies, coexisting overlay topologies can fairly share network bandwidth, evolve under various network dynamics, efficiently saturate the allocated bandwidth by using network coding, and can be prioritized.

The general realistic settings for our forthcoming experiments are as follows: Each network includes two classes of peers, 30% Ethernet peers with 10 Mbps upload capacities and 70% ADSL/Cable modem peers with heterogeneous upload capacities in the range of $0.4 - 0.8$ Mbps. Streaming servers in a network are Ethernet peers as well. We use delay-bandwidth products to represent streaming costs (M/M/1 delays), with streaming cost functions in the form of $D_{ij}^s = x_{ij}^s/(C_{ij} - x_{ij}^s)$. Here, C_{ij} is the available overlay link bandwidth, chosen from the distribution of measured capacities between PlanetLab nodes [10]. In asynchronous play of the auction games, the timeout value for an upstream/downstream peer to start a new round of bandwidth allocation/requested bandwidth calculation, T, is set to 1 second. The media stream to be distributed in each overlay is partitioned into 2-second segments, and each segment is further divided into 64 blocks.

A. Limited visibility of neighbor peers

In Assumption 1 of our game theoretical analysis in Sec. III-C, we assume that upload capacities in the network are sufficient to support all the peers to stream at required rates. This is generally achievable when the neighbor list maintained at each peer contains a lot of other peers in each overlay it participates in. However, in practical scenarios, each peer only has knowledge of a limited number of other peers, much smaller than the size of the network. We first study the convergence and optimality of the proposed strategies in such practical cases with asynchronous play of the auctions. As known neighbors constitute possible upstream peers in the streaming, the neighbor list at a peer is henceforth referred to as the *upstream vicinity* of the peer.

In our experiments, peers in upstream vicinity of each peer are randomly selected by a bootstrapping server from the set of all peers in the same overlay. The actual upstream peers at which each peer bids for bandwidth are decided by their content availability during streaming. In this set of experiments, we focus on allocated streaming bandwidth at each peer from the auctions, and will investigate the difference between allocated bandwidth and actually achieved streaming rate of media delivery with network coding in Sec. V-C. Specifically, we seek to answer the following questions: First, what is the appropriate size of the upstream vicinity, such that the auctions converge and the required streaming bandwidth can be achieved at all peers in an overlay? Second, if the upstream vicinity is smaller, do the peers need to bid longer before the auction games converge? Finally, how different is the resulting optimal topologies when auction strategies are used with upstream vicinities of various sizes, with respect to streaming cost?

Evaluation. We investigate by experimenting in networks with 100 to $10,000$ peers with various sizes of upstream vicinities. In each network, we now consider a single overlay, with one server serving a 1 Mbps media stream to all peers.

Fig. 4 illustrates the outcome of our distributed auction strategies, either when they converge, or when a maximum bidding time, 2 seconds, has been reached. In the latter case, we assume the games have failed to converge, as there exist peers that cannot achieve the required streaming bandwidth with their current size of upstream vicinities. Decreasing the size of upstream vicinities from $n-1$ where n is the total number of peers in each network, we discover that with $20-30$ peers in the upstream vicinity, the games can still converge and the required streaming bandwidth can still be achieved at all peers in most networks, as shown in Fig. 4(A) and (B). Fig. 4(B) further reveals that convergence is always achieved rapidly (in a few seconds) in all networks with different sizes

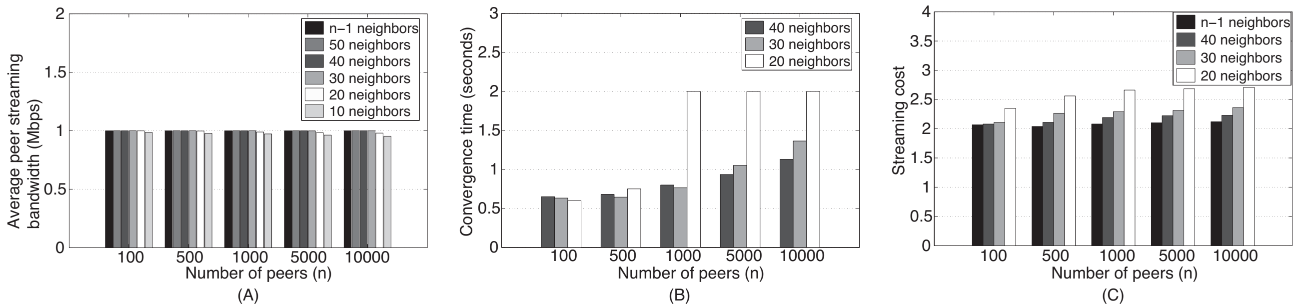


Fig. 4. Outcomes of distributed auctions in networks of different sizes, and with various sizes of upstream vicinities.

of upstream vicinities, as long as these games converge at all with a particular upstream vicinity size. A careful observation exhibits that the auction games take slightly longer to converge in larger networks. For a fixed network size, with larger upstream vicinity, a peer may spend more time in receiving bandwidth allocation and computing bandwidth requests, but carry out fewer bidding rounds before it acquires the required streaming bandwidth. Therefore, the total time to convergence remains similar with different upstream vicinity sizes when the auctions converge, and only distinguishes itself in larger networks when they fail to converge.

Fig. 4(C) compares the optimality of resulting topologies in terms of their global streaming costs computed with the allocated bandwidths. Although each resulting topology achieves streaming cost minimization with respect to its own input mesh topology, the global streaming cost is less when the input topology is denser with larger upstream vicinities. However, compared to the ultimate minimum streaming cost achieved when upstream vicinities contain all other peers in the overlay, the cost experienced by using upstream vicinities of a much smaller size (30) is only 10% higher.

Summary. From these observations, it appears that the appropriate size of upstream vicinities is relatively independent of network sizes, and the bandwidth allocation converges quickly in most cases. Both are good news when our auction strategies are to be applied in realistic large-scale networks. Based on results in this section, in our following experiments, the upstream vicinity size at each peer is set to 30.

B. The case of multiple coexisting overlays

We now proceed to study how our game strategies resolve the bandwidth competition among multiple coexisting streaming overlays. In particular, how does the topology of each overlay evolve, if coexisting overlays are started in the network? Do multiple coexisting overlays fairly share network bandwidth, and experience similar streaming costs?

Evaluation 1. We introduce more and more streaming overlays onto a 1000-peer network: At the beginning, all peers participate in one overlay and start to bid for their streaming bandwidths. Then every 10 seconds, the peers join one more new streaming overlay. To clearly show the effects of an increasing number of coexisting overlays on the allocated streaming bandwidth of each existing overlay, the required streaming rates for all overlays are set to the same 1 Mbps.

Fig. 5 illustrates the evolution of the average allocated peer streaming bandwidth in each overlay over time, when 5 overlays are sequentially formed in the network. We can see when there are up to 3 overlays coexisting in the network, the upload capacities in the network are sufficient for each overlay to achieve their required streaming bandwidth. When there are 4 or 5 overlays, the capacities become insufficient to support all the overlays.

In the former case with $1-3$ overlays, every time a new overlay is formed, the previous equilibrium bandwidth allocation across overlays is disturbed, and the games quickly converge to a new equilibrium, in which each overlay achieves the required streaming bandwidth again. In addition, the costs experienced by coexisting overlays when their topologies stabilize are shown in Fig. 6. We observe both streaming and bidding costs are very similar across the multiple coexisting overlays.

In the latter case with $4-5$ overlays in the network, Fig. 5 shows that the games fail to converge, and the streaming bandwidth obtained by each overlay fluctuates over time. We observed during the experiment that peers in each overlay bid higher and higher prices at their upstream peers, but were nevertheless unable to acquire the required streaming bandwidth. Similar bandwidth deficits can be observed in all coexisting overlays from Fig. 5.

In practical P2P applications, some streaming overlays might expect to receive better service quality than others. For example, live streaming of premium television channels should enjoy a higher priority and better quality than regular ones. Since our game strategies can achieve fairness among various overlays (as observed from Fig. 5 and Fig. 6), we wonder if it is further possible to introduce a practical prioritization strategy in our games, such that differentiated service qualities can be provided to different overlays.

In our previous experiment, we have observed that overlays fairly share bandwidth for a simple reason: peers in different overlays are not constrained by a bidding *budget*, and they can all raise bid prices at will to acquire more bandwidth from their desired upstream peers, which leads to relative fair bandwidth allocation at the upstream peers. Motivated by such insights, we introduce a budget-based strategy to achieve service differentiation, by offering higher budgets to peers in higher priority overlays. To introduce such budgets, we only need to make the following minor modification to the bidding strategy proposed in Sec. III-B:

When a peer j joins a streaming overlay s, it obtains a bidding budget W_s from its bootstrapping server. Such a budget represents the "funds" peer j can use to acquire bandwidth in overlay s, and its total bidding costs to all upstream peers

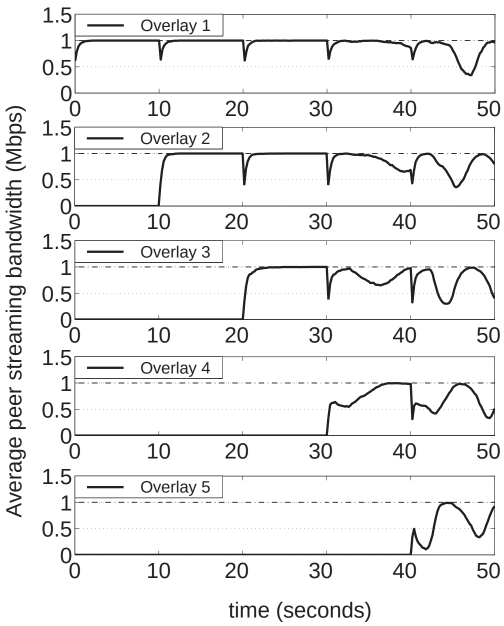


Fig. 5. The evolution of peer streaming bandwidth in multiple coexisting overlays with an increasing number of overlays over time.

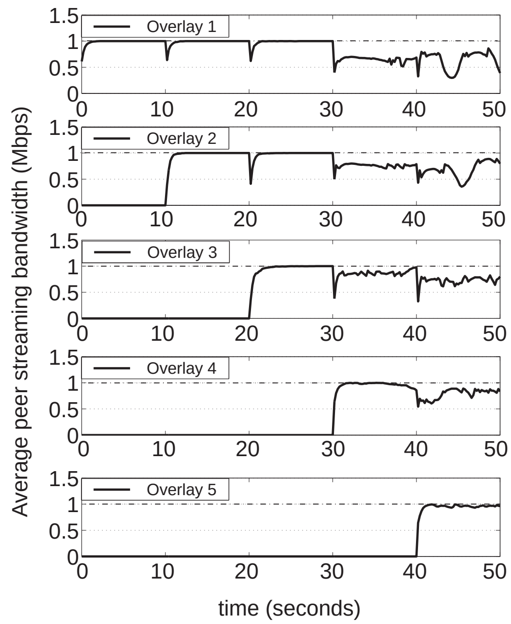


Fig. 7. The evolution of peer streaming bandwidth for multiple coexisting overlays with different budgets, and with an increasing number of overlays over time.

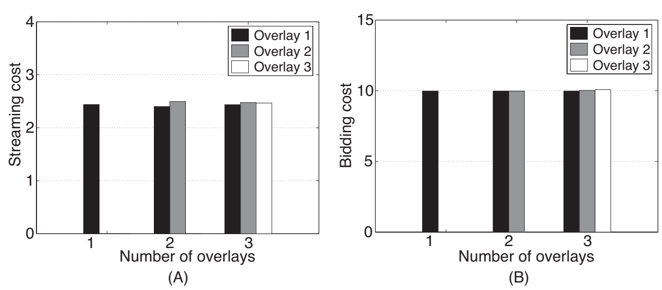


Fig. 6. A comparison of costs among multiple coexisting overlays.

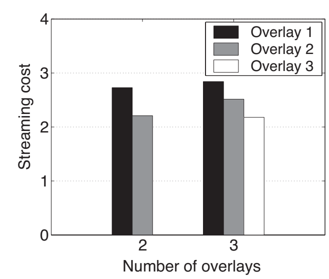


Fig. 8. A comparison of streaming costs among multiple coexisting overlays with different budgets.

cannot exceed this budget, i.e., $\sum_{i:(i,j)\in\mathcal{A}_s} p^s_{ij}x^s_{ij} \leq W_s$. *All peers in the same overlay receive the same budget, and the bootstrapping server assigns different levels of budgets to different overlays based on their priorities. During its price adjustments in overlay s, peer j may only increase its bid prices if the incurred total bidding cost does not exceed W_s.*

Evaluation 2. Applying the budget-based bidding strategy, we perform the previous experiment again and show our new results in Fig. 7 and Fig. 8. The budgets assigned to peers in overlay 1 to 5 range from low to high.

Comparing Fig. 7 with Fig. 5 in the cases when 1 to 3 overlays coexist, we see that overlays can still achieve their required streaming bandwidth within their budgets. However, when comparing Fig. 8 to Fig. 6(A), we observe that the streaming costs are differentiated across overlays, *i.e.*, overlays with larger budgets are able to achieve lower streaming cost than those with smaller budgets. This is because the former can afford to pay higher prices and thus eclipse the latter in auctions at their commonly desired upstream peers.

A further comparison between Fig. 7 and Fig. 5 (when 4 or 5 overlays coexist) shows that, when upload capacities become insufficient, the overlay with the highest budget, overlay 4 or overlay 5 in respective phases, always achieves the highest and most stable streaming rates, while those for overlays with smaller budgets become less sufficient and less stable.

Summary. We have observed that, no matter if upload capacities are sufficient or not, our game strategies achieve fair bandwidth sharing among multiple coexisting overlays. When overlays are able to achieve their required streaming bandwidths, they also experience similarly costs, which further reveal their fair share of lower latency paths. Further, we show that by introducing budgets to our bidding strategy, we are able to differentiate service qualities among coexisting overlays.

C. Overlay interaction under peer dynamics

In the following set of experiments, we study how coexisting streaming overlays evolve with peer arrivals and departures, with respect to how the allocated bandwidth and actually achieved streaming rate in each overlay vary in such dynamics. We investigate both cases that the overlays have or do not have differentiated budgets.

Evaluation. We simulate a dynamic P2P streaming network, in which 2 servers concurrently broadcast 4 different 60-minute live streaming sessions, at the streaming rate of 300 Kbps, 500 Kbps, 800 Kbps and 1 Mbps, respectively. Starting from the beginning of the live broadcasts, 1000 peers join the network following a Poisson process. The inter-arrival times follow an exponential distribution with an expected length of *INTARRIV* seconds. Upon arrival, each peer randomly selects 2 broadcast sessions and joins the respective overlays; then the peer stays in the network for a certain period of time, following an exponential lifetime distribution with an expected length of *LIFETIME* seconds. In this way, we simulate 4 dynamically evolving streaming overlays with approximately the same number of participating peers at any time. Network coding is implemented on each peer in each overlay, which codes the blocks in each segment of the media stream over $GF(2^{16})$. Similar to the previous experiments, each peer maintains about 30 neighbors in each overlay it participates in, and bids at upstream peers that have innovative coded blocks to serve it. All other settings of the experiments are identical to those in previous experiments. We monitor both the allocated bandwidths from the dynamic auctions and the actually achieved streaming rates of receiving coded blocks at existing peers in each dynamic overlay during the 60-minute broadcasts.

Fig. 9 and Fig. 10 show the results achieved when the budget-based strategy is not applied. Setting *INTARRIV* and *LIFETIME* to different values, we have repeated the experiment, and made the following observations: With expected inter-arrival time of 1 second, 1000 peers have all joined the network in the first 10 minutes; peer arrivals last for 45 minutes when *INTARRIV* is 3 seconds. With an expected lifetime of 10 minutes, most peers have left the network before the end of streaming; when *LIFETIME* is 30 minutes, approximately half of all the peers remain till the end.

Therefore, the most severe peer dynamics occurs when 1000 peers keep joining for 45 minutes, but have almost all left before 60 minutes, *i.e.*, the cases shown in Fig. 9(C) and Fig. 10(C), which represent the highest level of fluctuations for both the allocated bandwidth and achieved streaming rate. A longer peer lifetime brings better overlay stability, which is illustrated by the smaller rate fluctuation in (B) and (D) of Fig. 9 and Fig. 10. A careful comparison of the fluctuation of the allocated bandwidth across different overlays in Fig. 9 reveals slightly larger fluctuations for overlays with larger streaming rate requirements. This is because with our auction strategies, different overlays fairly share upload capacities at common upstream peers, and the larger the required bandwidth is, the harder it is to achieve.

Comparing Fig. 10 to Fig. 9, we can see that the actually achieved peer streaming rates in all overlays under all settings are close to the bandwidths allocated to the respective overlays in the dynamic auctions. This exhibits that the allocated bandwidths can be efficiently utilized to deliver useful blocks, based on the media distribution scheme using network coding.

On the other hand, when overlays with higher rate requirement are prioritized with higher budgets, Fig. 11 shows a different outcome from that in Fig. 9. In Fig. 11, under all four interval settings, the prioritized high-rate overlays are always guaranteed more stable bandwidths, while low-rate overlays experience more severe bandwidth fluctuations. Similar to the case without overlay prioritization, we have further observed that the actually achieved streaming rates in all the overlays are close to their allocated bandwidths, and thus omit the related illustrations here.

Summary. We have clearly demonstrated the effectiveness of our auction strategies under high degrees of peer dynamics, which guarantee stable streaming bandwidth allocation for all overlays at all times during such dynamics. We have also shown that together with network coding based media distribution, the allocated bandwidth can be maximally utilized to actually achieve satisfactory streaming rates. Using the budget-based bidding strategy, better streaming quality can be further provided for prioritized overlays.

VI. Related Work

There exists little literature that studies interactions and competitions among multiple coexisting overlays in a same P2P network. Recent work from Jiang *et al.* [11] and Keralapura *et al.* [12] are the most related, focusing on multiple overlay routing. In Jiang *et al.* [11], interactions among multiple selfish routing overlays are studied with a game theoretic model, where each overlay splits its traffic onto multiple paths and seeks to minimize its weighted average delay. In Keralapura *et al.* [12], route oscillations are investigated when multiple routing overlays inadvertently schedule their own traffic without knowledge of one another. Comparably, our work is significantly different, as we consider multiple P2P streaming overlays featuring many-to-many traffic, instead of point-to-point traffic in routing overlays.

Mesh-based P2P streaming solutions have been designed and successfully implemented in practice in recent years [1], [13], [14], [2], [3]. While multi-overlay streaming is the norm in these state-of-the-art mesh-based solutions, we are only aware of two pieces of work that touch upon the topic of coexisting live streaming overlays [15], [16]. These work propose to encourage peers in different overlays to help each other by relaying media belonging to other overlays. While it is beneficial to improve network resource utilization at a specific time, there are questions remaining to be answered: How should each peer carefully allocate its upload capacity among concurrently requesting peers from different overlays? If new requests from peers in the same overlay come later, should the bandwidth allocated to other overlays be deprived? From a more practical perspective, our work considers the case that each overlay consists of only receiving peers but each peer may participate in multiple overlays, and investigates bandwidth competition among the overlays at their common upstream peers.

Auction-based approaches have been proposed to allocate network bandwidth based on the demand and willingness to pay from competing users [17], [18], [19], [20]. A majority of such work are based on Progressive Second Price auctions, in which competitors decide their bids based on their true valuation. Aiming to solve the congestion problem on a single link or path, such existing work deals with elastic traffic, and competitors bid for their bandwidth share to maximize their utilities. In comparison, we design bandwidth auctions in a more complicated and practical scenario of constructing multiple streaming overlay topologies. Demanding an inelastic streaming rate at a lowest possible cost, each peer bids in multiple auctions, and adjusts its bid prices and requested

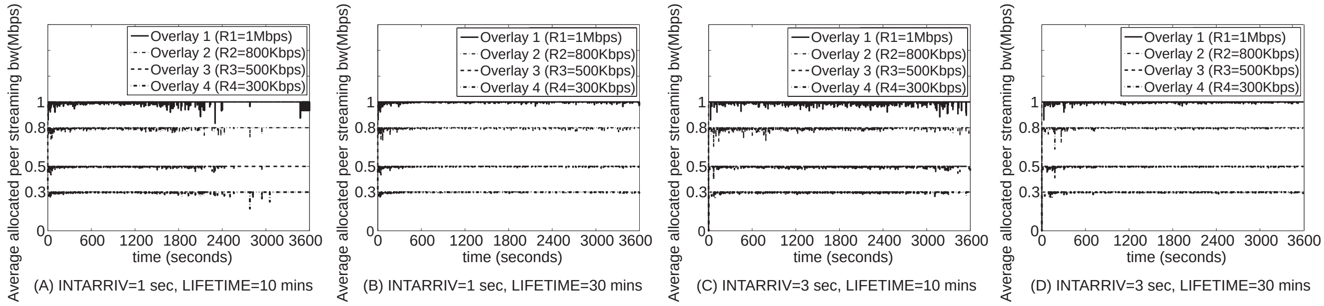


Fig. 9. Allocated streaming bandwidths for 4 coexisting overlays: under peer dynamics without budget

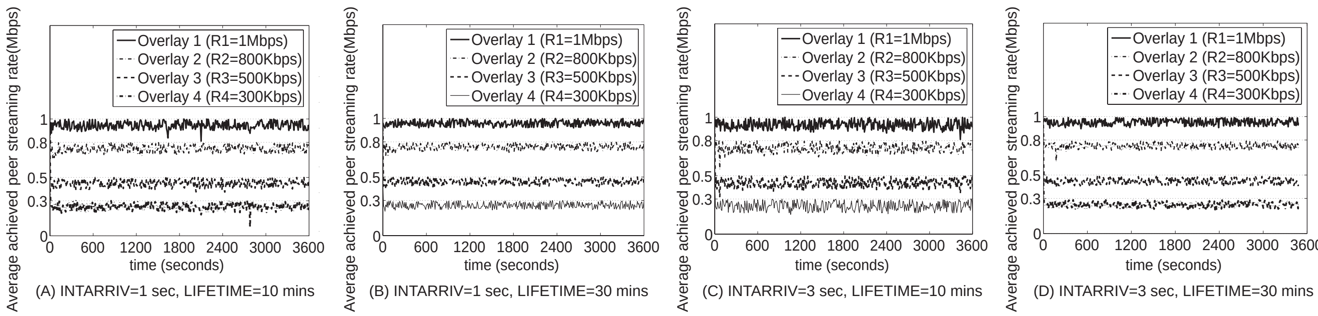


Fig. 10. Actually achieved streaming rates for 4 coexisting overlays: under peer dynamics without budget.

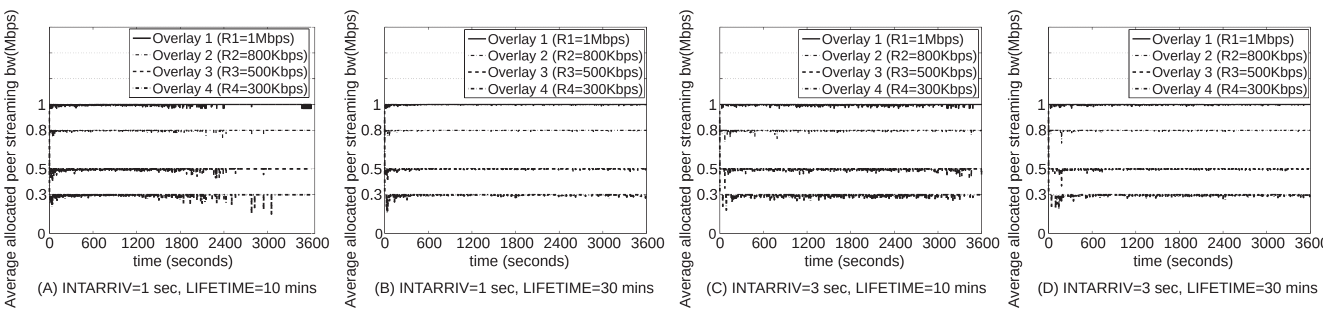


Fig. 11. Allocated streaming bandwidths for 4 coexisting overlays: under peer dynamics with different budgets.

bandwidths judiciously based on the current marginal cost of streaming from different upstream peers.

In P2P content distribution, game theory has been widely used to characterize peer selfishness and to provide incentives for peers to contribute their upload capacities (*e.g.*, [21], [22], [23], [24]). Different from previous work, incentive engineering, selfishness and strategyproofness are not parts of our focus in this paper. Instead, our work utilizes the distributed and dynamical nature of auction games to design effective mechanisms for demand-driven dynamic bandwidth allocation, in which local games achieve globally optimal topology construction.

In addition, pricing mechanisms [25], [26], [27] are proposed for a bandwidth provider to establish bandwidth prices to charge users, in order to regulate the behavior of selfish users and achieve social welfare maximization. Such pricing schemes are different from our auction games, in the sense that bandwidth prices are determined solely by the provider, rather than from bid prices placed by users.

Finally, network coding has been first proposed to achieve the maximum capacity of a multicast network [7], [8]. Due to its benefits of enhanced block diversity and bandwidth efficiency, in recent years, network coding has been applied to P2P content distribution applications [6], [9] and P2P streaming [4], [5]. Our focus in this work is not to propose a new P2P streaming protocol using network coding, but to show the seamless integration of our bandwidth auction strategies with the state-of-the-art framework of streaming with network coding, in achieving maximal bandwidth utilization across multiple streaming overlays.

VII. Concluding Remarks

This paper considers conflict-resolving strategies among multiple coexisting overlays for streaming in peer-to-peer networks. Our objective is crystal clear: we wish to devise practical and completely decentralized strategies to allocate peer upload capacities and efficiently utilize the allocated bandwidth, such that (1) the streaming rate requirement can be satisfied in each overlay; (2) streaming costs can be globally minimized; and (3) overlays fairly share available upload bandwidths in the network. Most importantly, we wish to achieve global optimality using localized algorithms. We use dynamic auction games to facilitate our bandwidth allocation, use game theory in our analysis to characterize the conflict among coexisting overlays, and discuss the integration of our bandwidth auctions with network coding based media distribution to achieve most efficient utilization of network bandwidth. Finally, we show that our proposed algorithm adapts well to the dynamics in P2P networks, the optimally allocated bandwidths can be fully utilized to deliver useful blocks at all times, and service can be differentiated across overlays with different bidding budgets.

APPENDIX A
PROOF OF THEOREM 1

Proof: Let $x_{ij}^{s*}, \forall i : (i,j) \in \mathcal{A}_s$ be an optimal solution to the problem **Bidding $\mathbf{j^s}$** in (3). Introducing Lagrangian multiplier λ for the constraint in (4) and $\nu = (\nu_i, \forall i : (i,j) \in \mathcal{A}_s)$ for the constraints in (5), we obtain the KKT conditions for the problem **Bidding $\mathbf{j^s}$** as follows (pp. 244, [28]):

$$\sum_{i:(i,j)\in\mathcal{A}_s} x_{ij}^{s*} \geq R_s,$$

$$\lambda^* \geq 0,$$

$$x_{ij}^{s*} \geq 0, \nu_i^* \geq 0, \forall i : (i,j) \in \mathcal{A}_s,$$

$$\lambda^*(R_s - \sum_{i:(i,j)\in\mathcal{A}_s} x_{ij}^{s*}) = 0, \tag{13}$$

$$x_{ij}^{s*}\nu_i^* = 0, \forall i : (i,j) \in \mathcal{A}_s, \tag{14}$$

$$D_{ij}^{'s}(x_{ij}^{s*}) + p_{ij}^s - \lambda^* - \nu_i^* = 0,$$
$$\forall i : (i,j) \in \mathcal{A}_s. \tag{15}$$

For $x_{ij}^{s*} > 0$, we have $\nu_i^* = 0$ from (14). Then from (15), we derive the marginal cost with respect to x_{ij}^{s*}, $\frac{df_j^s(x^*)}{dx_{ij}^s} = D_{ij}^{'s}(x_{ij}^{s*}) + p_{ij}^s = \lambda^*$. Since D_{ij}^s is strictly convex and twice differentiable, the inverse function of $D_{ij}^{'s}$, i.e., $D_{ij}^{'s-1}$, exists and is continuous and one-to-one. Then we have $\forall i : (i,j) \in \mathcal{A}_s)$

$$x_{ij}^{s*} = \begin{cases} 0 & \text{if } \lambda^* < D_{ij}^{'s}(0) + p_{ij}^s, \\ D_{ij}^{'s-1}(\lambda^* - p_{ij}^s) & \text{if } \lambda^* \geq D_{ij}^{'s}(0) + p_{ij}^s. \end{cases} \tag{16}$$

In deriving the optimal solution which achieves a same marginal cost value λ^* for all the positive x_{ij}^{s*}'s, we always increase the smallest marginal cost $D_{ij}^{'s}(x_{ij}^s) + p_{ij}^s$ by increasing the corresponding x_{ij}^s. In this way, we are increasing the marginal costs towards the same value of λ^*. As $\lambda^* > 0$, we derive that x_{ij}^{s*}'s satisfy $R_s - \sum_{i:(i,j)\in\mathcal{A}_s} x_{ij}^{s*} = 0$ based on (13). Therefore, this water-filling process continues until R_s is used up, i.e., $\sum_{i:(i,j)\in\mathcal{A}_s} x_{ij}^{s*} = R_s$, by which time we obtain the unique optimal solution defined by (16). □

APPENDIX B
PROOF OF THEOREM 2

Proof: Define $\mathcal{B}$ to be the set of all possible bid profiles in the entire network, *i.e.*, $B \in \mathcal{B}$ and $\mathcal{B} \subset \times_{j,s}\Gamma_j^s$. From the definition of Γ_j^s in (7), we know $\mathcal{B}$ is convex and x_{ij}^s's are bounded. In addition, under Assumption 1, all peers in all overlays can obtain enough streaming bandwidths, and thus their bid prices to upstream peers will not be increased infinitely in the respective auctions. Therefore, all prices p_{ij}^s's in a possible bid profile are bounded, *i.e.*, $\exists \bar{p} > 0, p_{ij}^s \in [0, \bar{p}], \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$. Altogether, we derive that $\mathcal{B}$ is a convex compact set.

The action profile for each player j^s — $B_j^s = (b_{ij}^s, \forall i : (i,j) \in \mathcal{A}_s)$ — can also be represented as $B_j^s = (P_j^s, X_j^s)$, where $P_j^s = (p_{ij}^s, \forall i : (i,j) \in \mathcal{A}_s)$ is the vector of bid prices toward all upstream peers of player j^s, and $X_j^s = (x_{ij}^s, \forall i : (i,j) \in \mathcal{A}_s)$ is the vector of requested bandwidths towards all upstream peers at player j^s.

The price adjustment strategy described in Table I defines a mapping function θ_j^s, from bid profile B in the previous bidding round, to new prices to bid by player j^s, *i.e.*, $P_j^s = \theta_j^s(B)$.

Given price vector P_j^s, Theorem 1 gives that the water-filling approach uniquely decides the best requested bandwidth assignment X_j^s at player j^s. This mapping from price vector P_j^s to requested bandwidth vector X_j^s can be defined as function $X_j^s = \varphi_j^s(P_j^s)$.

Let $g_j^s(B) = (\theta_j^s(B), \varphi_j^s(\theta_j^s(B))$ be the mapping function from bid profile B in the previous bidding round to a new action profile at player j^s. Let $g(B) = (g_j^s(B), \forall j \in \mathcal{N}_s, s \in \mathcal{S})$. Therefore, $g(B)$ is a point-to-point mapping from $\mathcal{B}$ to $\mathcal{B}$. The Nash equilibrium is a fixed-point of this mapping. We next show the existence of such a fixed-point.

We first show φ_j^s is a continuous mapping. Given P_j^s, X_j^s is the optimal solution of **Bidding $\mathbf{j^s}$**. Therefore, φ_j^s is defined by (16) in Appendix A. Since D_{ij}^s is strict convex and twice differentiable, we know $D_{ij}^{'s}$ is continuous. Thus based on the water-filling process, we know the optimal marginal cost λ^* is continuous on P_j^s. Furthermore, as $D_{ij}^{'s-1}$ is continuous too, from (16), we derive X_j^s is continuous on P_j^s, *i.e.*, φ_j^s is continuous.

We next show θ_j^s is a continuous mapping from $\mathcal{B}$ to vector space of bid prices P_j^s at player j^s. Let p_{ij}^s be the bid price player j^s places to upstream peer i in the previous bidding round, and q_{ij}^s be the new bid price after the price adjustment defined by θ_j^s. Without loss of generality, we simplify our proof by showing that a small disturbance of the previous price p_{ij}^s to $p_{ij}^{'s} = p_{ij}^s + \epsilon, \epsilon > 0, \epsilon \to 0$ results in little disturbance at new price q_{ij}^s, *i.e.*, letting $q_{ij}^{'s}$ denote the new price corresponding to $p_{ij}^{'s}$, we have $q_{ij}^{'s} \to q_{ij}^s$. We divide our discussions into 2 cases, and first give a result to be used in the discussions: If p_{ij}^s is increased to $p_{ij}^{'s}$ and all other bid prices at player j^s remain unchanged, the corresponding requested bandwidth to upstream peer i is decreased, *i.e.*, $x_{ij}^{'s} \leq x_{ij}^s$. This can be directly obtained from the water-filling process used to solve **Bidding $\mathbf{j^s}$**.

We now investigate the two cases:

A) At upstream peer i, there is no bid price from other players right between p_{ij}^s and $p_{ij}^{'s}$, *i.e.*, there does not exist p_{im}^k, such that $p_{ij}^s \leq p_{im}^k \leq p_{ij}^{'s}$.

Starting the price adjustment described in Table I from p_{ij}^s and $p_{ij}^{'s}$ respectively, we consider two sub cases: (i) If p_{ij}^s is to be reduced as $x_{ij}^s \leq a_{ij}^s$, we know $p_{ij}^{'s}$ is to be reduced too, since $x_{ij}^{'s} \leq x_{ij}^s$ but $a_{ij}^{'s} \geq a_{ij}^s$ (due to $p_{ij}^{'s} > p_{ij}^s$). When $p_{ij}^{'s}$ is reduced, it will soon reach p_{ij}^s, and its following adjustments will be the same as those for p_{ij}^s. (ii) Similarly, if p_{ij}^s is to be increased, it will soon reach $p_{ij}^{'s}$ and their following adjustments will be the same. In both cases, we have for the new prices $q_{ij}^{'s} \to q_{ij}^s$.

B) At upstream peer i, there is a bid price from another player which lies right between p^s_{ij} and $p^{'s}_{ij}$, *i.e.*, $\exists p^k_{im}$, such that $p^s_{ij} \leq p^k_{im} \leq p^{'s}_{ij}$.

We again discuss three sub cases: (i) If p^s_{ij} is to be reduced due to $x^s_{ij} \leq a^s_{ij}$, $p^{'s}_{ij}$ is to be reduced too, since $x^{'s}_{ij} \leq x^s_{ij}$ but $a^{'s}_{ij} \geq a^s_{ij}$ (due to $p^{'s}_{ij} \geq p^k_{im} \geq p^s_{ij}$). During $p^{'s}_{ij}$'s adjustments, its value is continuously decreased, passing p^k_{im} and reaching p^s_{ij}. Then its following adjustments will be the same as those for p^s_{ij}. (ii) If p^s_{ij} is to be increased due to $x^s_{ij} > a^s_{ij}$ and $p^{'s}_{ij}$ is to be reduced as $x^{'s}_{ij} \leq a^{'s}_{ij}$, they will both stop at a same value near p^k_{im}. (iii) If both p^s_{ij} and $p^{'s}_{ij}$ are to be increased, p^s_{ij}'s value will be continuously increased to pass p^k_{im} and reach p^s_{ij}. Then their following adjustments will be the same. In all cases, the new prices $q^{'s}_{ij} \rightarrow q^s_{ij}$.

Therefore, based on the continuity of θ^s_j and φ^s_j, we derive that the mapping $g^s_j(B) = (\theta^s_j(B), \varphi^s_j(\theta^s_j(B))$ is continuous. Thus, $g(B) = (g^s_j(B), \forall j \in \mathcal{N}_s, s \in \mathcal{S})$ is a continuous mapping from $\mathcal{B}$ to itself. Based on Brouwer Fixed Point Theorem, any continuous mapping of a convex compact set into itself has at least one fixed point, *i.e.*, $\exists B^* = g(B^*) \in \mathcal{B}$. In addition, the fixed point B^* must be a feasible profile, as otherwise the adjustments of prices and requested bandwidths do not converge. Therefore, the fixed point B^* is a Nash equilibrium of the extended game. □

Appendix C
Proof of Theorem 3

Proof: We prove by showing that at equilibrium, the KKT conditions satisfied by the equilibrium bid profile $B^* = ((p^{s*}_{ij}, x^{s*}_{ij}), \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S})$ are the same as KKT conditions for the global streaming cost minimization problem in (9).

At equilibrium, given p^{s*}_{ij}'s, the requested bandwidths at each player j^s, $x^{s*}_{ij}, \forall i : (i,j) \in \mathcal{A}_s$, are the optimal solution to the problem **Bidding j^s**, and are also the same as allocated bandwidths from respective upstream peers. Therefore, altogether, we know the bandwidth allocations in the entire network, $x^{s*}_{ij}, \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$, solve the following optimization problem:

$$\min \sum_{s\in\mathcal{S}} \sum_{j\in\mathcal{N}_s} \sum_{i:(i,j)\in\mathcal{A}_s} (D^s_{ij}(x^s_{ij}) + p^{s*}_{ij} x^s_{ij}) \tag{17}$$

subject to

$$\sum\nolimits_{i:(i,j)\in\mathcal{A}_s} x^s_{ij} \geq R_s, \quad \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S}, \tag{18}$$

$$x^s_{ij} \geq 0, \qquad \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}, \tag{19}$$

and also satisfy upload capacity constraints at all the upstream peers:

$$\sum_{s\in\mathcal{S}} \sum_{j:(i,j)\in\mathcal{A}_s} x^{s*}_{ij} \leq U_i, \quad \forall i \in \mathcal{V} \cup \mathcal{N}. \tag{20}$$

Introducing Lagrangian multiplier $\lambda = (\lambda^s_j, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S})$ for the constraints in (18) and $\nu = (\nu^s_{ij}, \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S})$ for constraints in (19), we obtain the KKT conditions for the optimization problem in (17) as follows:

$$\begin{aligned}
\sum_{s\in\mathcal{S}} \sum_{j:(i,j)\in\mathcal{A}_s} x^{s*}_{ij} &\leq U_i, \forall i \in \mathcal{V} \cup \mathcal{N}, \\
\sum_{i:(i,j)\in\mathcal{A}_s} x^{s*}_{ij} &\geq R_s, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S}, \\
x^* \geq 0, \lambda^* &\geq 0, \nu^* \geq 0, \\
\lambda^{s*}_j (R_s - \sum_{i:(i,j)\in\mathcal{A}_s} x^{s*}_{ij}) &= 0, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S}, \\
x^{s*}_{ij} \nu^{s*}_{ij} &= 0, \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}, \qquad (21) \\
D^{'s}_{ij}(x^{s*}_{ij}) + p^{s*}_{ij} - \lambda^{s*}_j - \nu^{s*}_{ij} &= 0, \\
&\forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}. \qquad (22)
\end{aligned}$$

For $x^{s*}_{ij} > 0$, we have $\nu^{s*}_{ij} = 0$ from (21), and $D^{'s}_{ij}(x^{s*}_{ij}) + p^{s*}_{ij} = \lambda^{s*}_j$ from (22). Since D^s_{ij} is strictly convex and twice differentiable, the inverse function of $D^{'s}_{ij}$, i.e., $D^{'s-1}_{ij}$, exists. Then we have $\forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$

$$x^{s*}_{ij} = \begin{cases} 0 & \text{if } \lambda^{s*}_j < D^{'s}_{ij}(0) + p^{s*}_{ij}, \\ D^{'s-1}_{ij}(\lambda^{s*}_j - p^{s*}_{ij}) & \text{if } \lambda^{s*}_j \geq D^{'s}_{ij}(0) + p^{s*}_{ij}. \end{cases} \tag{23}$$

Similarly, for the global streaming cost minimization problem in (9), introducing Lagrangian multiplier $q = (q_i, \forall i \in \mathcal{V} \cup \mathcal{N})$ for the constraints in (10), $\lambda = (\lambda^s_j, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S})$ for the constraint in (11) and $\nu = (\nu^s_{ij}, \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S})$ for constraints in (12), we obtain the following KKT conditions:

$$\begin{aligned}
\sum_{s\in\mathcal{S}} \sum_{j:(i,j)\in\mathcal{A}_s} y^{s*}_{ij} &\leq U_i, \forall i \in \mathcal{V} \cup \mathcal{N}, \\
\sum_{i:(i,j)\in\mathcal{A}_s} y^{s*}_{ij} &\geq R_s, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S}, \\
y^* \geq 0, q \geq 0, \lambda^* &\geq 0, \nu^* \geq 0, \\
q^*_i (\sum_{s\in\mathcal{S}} \sum_{j:(i,j)\in\mathcal{A}_s} y^{s*}_{ij} - U_i) &= 0, \forall i \in \mathcal{V} \cup \mathcal{N}, \qquad (24) \\
\lambda^{s*}_j (R_s - \sum_{i:(i,j)\in\mathcal{A}_s} y^{s*}_{ij}) &= 0, \forall j \in \mathcal{N}_s, \forall s \in \mathcal{S}, \\
y^{s*}_{ij} \nu^{s*}_{ij} &= 0, \forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}, \\
D^{'s}_{ij}(y^{s*}_{ij}) + q^*_i - \lambda^{s*}_j - \nu^{s*}_{ij} &= 0, \\
&\forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}. \qquad (25)
\end{aligned}$$

And similarly, we can obtain $\forall (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$

$$y^{s*}_{ij} = \begin{cases} 0 & \text{if } \lambda^{s*}_j < D^{'s}_{ij}(0) + q^*_i, \\ D^{'s-1}_{ij}(\lambda^{s*}_j - q^*_i) & \text{if } \lambda^{s*}_j \geq D^{'s}_{ij}(0) + q^*_i. \end{cases} \tag{26}$$

To show the two sets of KKT conditions are actually the same, we first demonstrate that at each upstream peer i, the equilibrium bid prices from all competing players that are allocated non-zero bandwidths are the same, *i.e.*, $\exists t^*_i, p^{s*}_{ij} = t^*_i$ if $x^{s*}_{ij} > 0$, $\forall j : (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$. This can be illustrated as follows: If a player in auction i is paying a price higher than some other player who is also allocated non-zero bandwidth, the former can always acquire more bandwidth from the later with a price lower than its current price. Thus at equilibrium, when no one can unilaterally alter its price, all players must be paying the same price.

In addition, we know from the price adjustment described in Table I that if upstream peer i's upload capacity is large enough to satisfy all bandwidth requests, the corresponding bid prices in auction i can all be lowered down to 0. Therefore, at equilibrium, if $U_i > \sum_{s\in\mathcal{S}}\sum_{j:(i,j)\in\mathcal{A}_s} x_{ij}^{s*}$, we have $p_{ij}^{s*} = t_i^* = 0, \forall j : (i,j) \in \mathcal{A}_s, \forall s \in \mathcal{S}$. Thus we know t_i^* satisfies

$$t_i^*(\sum_{s\in\mathcal{S}}\sum_{j:(i,j)\in\mathcal{A}_s} x_{ij}^{s*} - U_i) = 0, \forall i \in \mathcal{V}\cup\mathcal{N}. \tag{27}$$

From these results and comparing (27)(22) with (24)(25) respectively, we can derive $t_i^* = p_i^*, \forall i \in \mathcal{V}\cup\mathcal{N}$, and the two sets of KKT conditions are actually the same. Therefore, the equilibrium solution in (23) is the same as the optimal solution to the global streaming cost minimization problem in (26). Thus Theorem 3 is proven. □

References

[1] X. Zhang, J. Liu, B. Li, and T. P. Yum, "CoolStreaming/DONet: A Data-Driven Overlay Network for P2P Live Media Streaming," in *Proc. of IEEE INFOCOM, Miami, FL*, 2005.

[2] *PPLive*, http://www.pplive.com/.

[3] *UUSee*, http://www.uusee.com/.

[4] M. Wang and B. Li, "Lava: A Reality Check of Network Coding in P2P Live Streaming," in *Proc. of IEEE INFOCOM, Anchorage, Alaska*, 2007.

[5] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proc. of the 51st Allerton Conference on Communication, Control and Computing*, October 2005.

[6] P. Maymounkov, N. Harvey, and D. Lun, "Methods for Efficient Network Coding," in *Proc. of The 44th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL*, 2006.

[7] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[8] T. Ho, M. Médard, J. Shi, M. Effros, and D. Karger, "On Randomized Network Coding," in *Proc. of 41st Annual Allerton Conference on Communication, Control, and Computing*, October 2003.

[9] C. Gkantsidis, J. Miller, and P. R. Rodriguez, "Comprehensive View of a Live Network Coding P2P System," in *Proc. of ACM Internet Measurement Conference (IMC), Rio de Janeiro, Brazil*, 2006.

[10] "PlanetLab IPerf," http://jabber.services.planet-lab.org/php/iperf/.

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

[12] R. Keralapura, C.-N. Chuah, N. Taft, and G. Iannaccone, "Can Co-existing Overlays Inadvertently Step on Each Other," in *Proc. of 13th International Conference on Network Protocols (ICNP 2005)*, November 2005.

[13] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. E. Mohr, "Chainsaw: Eliminating Trees from Overlay Multicast," in *Proc. of the 4th International Workshop on Peer-to-Peer Systems*, February 2005.

[14] N. Magharei and R. Rejaie, "PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming," in *Proc. of IEEE INFOCOM 2007*, May 2007.

[15] X. Liao, H. Jin, Y. Liu, L. M. Ni, and D. Deng, "AnySee: Peer-to-Peer Live Streaming," in *Proc. of IEEE INFOCOM 2006*, March 2006.

[16] G. Tan and S. A. Jarvis, "Inter-Overlay Cooperation in High-Bandwidth Overlay Multicast," in *Proc. of the 35th International Conference on Parallel Processing (ICPP 06)*, August 2006.

[17] A. Lazar and N. Semret, "Design and Analysis of the Progressive Second Price Auction for Network Bandwidth Sharing," *Telecommunications Systems: Special Issue on Network Economics*, 1999.

[18] N. Semret, R. R.-F. Liao, A. T. Campbell, and A. A. Lazar, "Pricing, Provisioning and Peering: Dynamic Markets for Differentiated Internet Services and Implications for Network Interconnections," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2499–2513, December 2000.

[19] P. Maillé and B. Tuffin, "Multi-Bid Auctions for Bandwidth Allocation in Communication Networks," in *Proc. of IEEE INFOCOM 2004*, March 2004.

[20] M. Bitsaki, G. D. Stamoulis, and C. Courcobetis, "A New Strategy for Bidding in the Network-Wide Progressive Second Price Auction for Bandwidth," in *Proc. of the 1st Conference on Future Networking Technologies (CoNEXT 2005)*, 2005.

[21] R. T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau, "A Game Theoretic Approach to Provide Incentive and Service Differentiation in P2P Networks," in *Proc. of ACM SIGMETRICS/Performance'04*, June 2004.

[22] G. Tan and S. A. Jarvis, "A Payment-based Incentive and Service Differentiation Mechanism for Peer-to-Peer Streaming Broadcast," in *Proc. of the 14th International Workshop on Quality of Service (IWQoS 2006)*, June 2006.

[23] P. Golle, K. L. Brown, I. Mironov, and M. Lillibridge, "Incentives for Sharing in Peer-to-Peer Networks," in *Proc. of the 2nd International Workshop on Electronic Commerce*, 2001.

[24] K. Lai, M. Feldman, I. Stoica, and J. Chuang, "Incentives for Cooperation in Peer-to-Peer Networks," in *Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[25] F. P. Kelly, A. Maulloo, and D. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, March 1998.

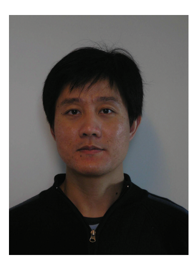
[26] T. Basar and R. Srikant, "Revenue-Maximizing Pricing and Capacity Expansion in a Multi-User Regime," in *Proc. of IEEE INFOCOM 2002*, June 2002.

[27] Y. Qiu and P. Marbach, "Bandwidth Allocation in Ad Hoc Networks: A Price-Based Approach," in *Proc. of IEEE INFOCOM 2003*, April 2003.

[28] S. Boyd, *Convex Optimization*. Cambridge University Press, 2004.

Chuan Wu. Chuan Wu received her B.Engr. and M.Engr. degrees from Department of Computer Science and Technology, Tsinghua University, China, in 2000 and 2002, respectively. She is currently a Ph.D. candidate at the Department of Electrical and Computer Engineering, University of Toronto, Canada. Her research interest is distributed algorithm design for improving Quality of Service of overlay multicast applications. She is particularly interested in applying optimization and game theory to guide practical protocol design.

Baochun Li. Baochun Li received his B.Engr. degree in 1995 from Department of Computer Science and Technology, Tsinghua University, China, and his M.S. and Ph.D. degrees in 1997 and 2000 from the Department of Computer Science, University of Illinois at Urbana-Champaign. Since 2000, he has been with the Department of Electrical and Computer Engineering at the University of Toronto, where he is currently an Associate Professor. He holds the Nortel Networks Junior Chair in Network Architecture and Services since October 2003, and the Bell University Laboratories Chair in Computer Engineering since July 2005. His research interests include application-level Quality of Service provisioning, wireless and overlay networks.



Zongpeng Li. Zongpeng Li received his B.Engr. degree in Computer Science and Technology from Tsinghua University (Beijing) in 1999, his M.S. degree in Computer Science from University of Toronto in 2001, and his Ph.D. degree in Electrical and Computer Engineering from University of Toronto in 2005. He has been working as an Assistant Professor at the Department of Computer Science in the University of Calgary since August 2005, and won the Alberta Ingenuity New Faculty Award in 2007. His research interests are in computer networks and network algorithms.