SALSA: Super-Peer Assisted Live Streaming Architecture

Jongtack Kim
School of EECS, INMC
Seoul National University
Email: jkim@netlab.snu.ac.kr

Yugyung Lee
School of Computing and Engineering
University of Missouri at Kansas City
Email: leeyu@umkc.edu

Saewoong Bahk
School of EECS, INMC
Seoul National University
Email: sbahk@netlab.snu.ac.kr

***Abstract*—In P2P live streaming, free-riders which do not upload data but only download them are still present. Since a greater number of viewers can generate more profit, the streaming server wants to serve the free-riders also. In this paper, we have proposed a novel P2P live streaming system, called *super-peer assisted live streaming architecture (SALSA)*. In SALSA, super-peers, which have high upload bandwidth and serve many free-riders, are eligible to receive an incentive reward that is the ability to watch high quality videos. The server places high-quality-view-tickets at auction to make the super-peers serve free-riders more efficiently. We have proposed novel auction mechanisms and a heuristic algorithm. The simulation results showed that the proposed scheme has comparable performance to the optimal form and are able to differentiate the super-peers' video quality commensurate with their contribution level.**

I. INTRODUCTION

In P2P live streaming, peers exchange video data they do not have with neighbors. As the number of peers increases, the upload bandwidth of the system also increases, leading to restriction of the overhead of the server. However, if the participating peers are *strategic*, they are apt to watch the video with the least upload. They usually get the highest payoff, when they do not upload data at all. The *free-riding* is a potential problem in P2P live streaming, as P2P file sharing has been under investigation for a long time.

In this paper, we propose a novel P2P live streaming system called *super-peer assisted live streaming architecture (SALSA)*. In SALSA, super-peers with high upload bandwidth convey the streaming with minimum quality to the free-riders, leading to gaining incentives commensurate with their contribution level from the server. The server's payoff, such as advertising revenue, increases as the number of viewers increases. Therefore, the server wants to accommodate as many free-riders as possible, which usually is limited by the upload bandwidth. The super-peers help the server convey the streaming to the free-riders, and in return, they can watch higher quality videos.

In order to differentiate the video quality, we adopt multiple description coding (MDC) [9]. In MDC, the video with bitrate R is encoded into M descriptions with bitrate R/M. Any $m(1 \leq m \leq M)$ descriptions can be decoded, and a greater number of descriptions generates higher quality videos with less distortion.

In SALSA, the server generates higher quality videos for the super-peer to serve a greater number of free-riders. Since the super-peers can also be strategic, their payoff increases when the video quality gets better and the number of serving free-riders gets smaller. The server puts a high-quality-view-ticket at auction among super-peers whose bid is equivalent to the maximum number of free-riders that the super-peer can serve. The super-peers which lose the bid in the auction can participate in another auction for the next-high-quality-view-ticket. The contributions of this paper can be summarized as follows:

- We design the novel auction mechanisms for a high-quality-view-ticket. The number of tickets is not constant, so the traditional auction mechanisms can be expanded.
- We propose a heuristic algorithm to accommodate as many free-riders as possible using the aforementioned auction mechanisms. The auctions are replicated for each video quality, so the total number of served free-riders depends on the criterion by which winners are selected at each auction. The proposed heuristic algorithm can serve a comparable number of free-riders.
- The overlay multicast tree architecture is adapted for the server to enforce the video quality of super-peers.

The rest of the paper is organized as follows: In section II, we briefly review the related work for free-riding and incentives in P2P streaming systems. Section III describes the behavior of a server, a super-peer, and a free-rider. In section IV, we propose the novel auction mechanisms and a heuristic algorithm to decide the quality of the super-peers and the maximum number of free-riders served. In section V, the performance of SALSA is evaluated. We conclude the paper in section VI.

II. RELATED WORK

The impact of altruism on the performance improvement of P2P streaming systems was previously described in [3]. A few altruistic nodes can significantly enhance the efficacy of the system. In [2], the same authors applied the taxation model to the nodes with high bandwidth for uploading more than what they could download, so the nodes with low bandwidth can also be utilized. Both approaches assume that the server is able to know each peer's accurate bandwidth, but in reality, strategic nodes can easily forge their bandwidth information to optimize their payoff. In SALSA, reporting accurate amount

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

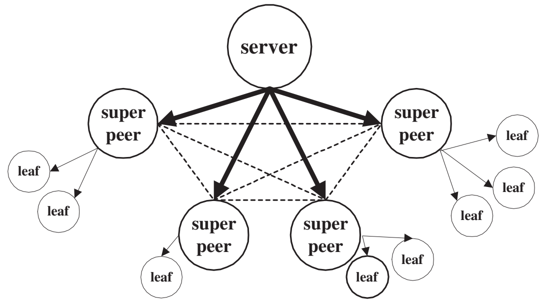

Fig. 1. An example of SALSA architecture. SALSA consists of a server, super-peers, and leaf-peers.

of bandwidth to serve free-riders to the server is one of the optimal strategies for the super-peers.

In [8], a node earns points by uploading data and participates in an auction for selecting good parents based on the points. However, implementing the points needs a payment mechanism. In SALSA, the cost for an auction is the number of free-riders in service, which monitor and report the service status of the serving super-peer.

In [4] and [5], it was reported that a greater number of uploading nodes was provided with a higher quality video using MDC and layered coding. However, the schemes could not utilize the residual bandwidth of high capacity nodes. SALSA is intended to increase the overall uplink utilization by serving free-riders with high bandwidth nodes.

SALSA is distinctively different from previous researches due to the super-peer concept incorporated in P2P live streaming and the new auction mechanisms employed to know the exact number of free-riders super-peers can serve.

III. SALSA Overview

There are three kinds of nodes in SALSA. The first one is a *streaming server*, which exists solely in the system. The server encodes live streaming using MDC and transmits to super-peers. Since the server does not have enough bandwidth to send most descriptions to all super-peers, it transmits 0 or a few descriptions to one super-peer. The second one is a *super-peer*. There are many super-peers in the system. A super-peer receives a few descriptions from the server, and exchanges them with other super-peers to improve video quality. A super-peer should serve at least one free-rider with minimum quality. The third one is a free-rider which is called a *leaf-peer* in this paper. There are many leaf-peers in the system. The leaf-peers do not upload video data but only download them. Each leaf-peer is connected to a super-peer from which it receives the minimum quality video. An example that a server, super-peers, and leaf-peers are connected with each other is shown in Fig. 1.

A. Behavior of a streaming server

The streaming server is intended to serve as many leaf-peers as possible. The number of viewers is connected directly to the server's payoff including advertising revenue. Due to the server's limited bandwidth, the number of leaf-peers that can be served is also limited. If the number of leaf-peers exceeds the upper limit, super-peers begin to help the server. The more leaf-peers a super-peer serves, the higher quality a video gets in return.

The server wants the super-peers to utilize full bandwidth to serve the leaf-peers. If this condition is met, the server in return is able to provide all the super-peers with the highest quality video. Under this guideline, the super-peers try to forge their bandwidth as small as possible to increase their payoff. Subsequently, the server puts high-quality-view-tickets at auction among the super-peers. A super-peer's bid for the auction depends on the number of leaf-peers that it can serve.

TABLE I
Bidders' valuation and bids for an object

Bidder	1	2	3	4	5	6	7	8	9	10
Valuation	100	90	80	70	60	50	40	30	20	10
Bid	95	89	80	70	55	40	40	15	20	5

B. Behavior of a super-peer

A super-peer's payoff increases, as the quality of the video improves. When the video quality is the same, payoff increases as the number of leaf-peers which a super-peer serves decreases. The video quality is denoted as $Q_j(1 \leq j \leq M)$, where the number of received descriptions is j. Then the super-peer participates in the auction game for quality Q_j with other super-peers. For each Q_j, super-peer i decides the maximum number of leaf-peers it can serve, K_{ij}. Depending on the status of super-peer i, such as bandwidth usage of other software or satisfaction level for current video quality, K_{ij} can be changed on occasion. Super-peer i tries to conceal K_{ij} to maximize its payoff. In order to cope with this process, the server periodically puts high-quality-view-tickets at auction.

C. Behavior of a leaf-peer

A leaf-peer receives a list of available super-peers from the streaming server. It requests a connection to the super-peer having the lowest RTT. The super-peer accepts it if the number of currently serving leaf-peers is smaller than those guaranteed by the server. Then, the leaf-peer sends the super-peer's ID to the server and obtains the minimum quality video from the super-peer. If the super-peer's service does not secure the minimum quality, it reports to the server and requests a new connection to another super-peer.

IV. SALSA Detail

A. Auction design

1) First-price and second-price auction: An auction can be classified into a first-price and a second-price auction. In the first-price auction, the bidder offering the highest bid wins an object A at the price of 'a'. On the contrary, in the second-price auction, the bidder offering the highest bid obtains the object and pays price 'b' which is equal to the second highest bid. As shown in the example described in Table I, where the bidders, the bidders' valuation for the object, and the bidders' bids are presented, bidder 1 is always the winner in both auctions, in which the price is 95 in the first-price auction and 89 in the second-price auction.

From the auctioneer's point of view, the first-price auction seems to be more profitable. However, in the first-price auction, the winner still obtains the object even if the bid was slightly lowered. In other words, if bidder 1 decreases her bid to 90, she still obtains the object while saving a cost of 5. Therefore, the bidding profile in Table I is not a Nash equilibrium. In the second-price auction, the bid equivalent to her valuation for the object makes not only a Nash equilibrium but also a weakly dominating strategy. It is due to the fact that the highest bid decides the winner, but the winning price can be decided by the rest of bids.

2) n-winner $(n+1)^{th}$-price auction: The auction needed in SALSA is different from the above ones. There are sufficient tickets for quality Q_j, and even all the super-peers can obtain the tickets. So we consider a new auction where the number of winners is n and the price is $(n+1)^{th}$ bid. We name it *n-winner $(n+1)^{th}$-price auction*. The bidders whose bid is equal to the winners' also become winners. Now we find the weakly dominating strategy in this auction following the similar way to the second-price auction.

Proposition 1: In an n-winner $(n+1)^{th}$-price auction, for super-peer i and quality Q_j, a bid equal to its valuation K_{ij} weakly dominates all its other bids.

Proof: For any action profile $(b_1, \cdots, K_{ij}, \cdots, b_N)$, denote the price by p.
i) If super-peer i is a winner, its payoff is $K_{ij} - p$.
i-a) Increasing its bid does not change winners or the price, so its payoff is the same.
i-b) Decreasing its bid until p does not change winners or the price, so its payoff is the same. If its bid becomes equal to or less than p, it loses in the auction, so the payoff decreases to 0.
ii) If super-peer i is a loser, its payoff is 0.
ii-a) Increasing its bid to become a winner makes the price larger than K_{ij}, so the payoff becomes smaller than 0.
ii-b) Decreasing its bid does not make it a winner, so the payoff is the same. ■

By Proposition 1, super-peer i bids K_{ij} for quality Q_j in an n-winner $(n+1)^{th}$-price auction. Therefore, the server comes to know how many leaf-peers super-peer i can afford to serve to watch the video of quality Q_j.

When the number of leaf-peers increases, the auction outcome for quality Q_j may not be enough to serve all the leaf-peers. So the server subsequently puts quality Q_{j-1} at auction among the losers to serve more leaf-peers. In these repeated auctions, the criterion to decide the number of winners at each auction results in a different total number of served leaf-peers. The more leaf-peers are served, the higher profit the server gets. Denote by w_j and p_j the number of winners and the price of quality Q_j, respectively. Then, the solution $(w_M, \cdots, w_1)$ to the following problem is an optimal allocation of winners.

Algorithm 1 Heuristic algorithm

$i \leftarrow$ Maximum video quality
$j \leftarrow i - 1$
$L_i \leftarrow$ All super-peers
sum $\leftarrow 0$
repeat
  $(W_i, L_i, p_i) \leftarrow$ (Winners, Losers, Price) from m-auction for Q_i
  tempSum $\leftarrow$ sum + $|W_i| \times p_i$
  if tempSum + $|L_i| < K$ **then**
    $(W_r, L_r, p_r) \leftarrow$ (Winners, Losers, Price) from r-auction for Q_i
    $(W_j, L_j, p_j) \leftarrow$ (Winners, Losers, Price) from m-auction for Q_j
    if $|W_i| \times p_i < |W_r| \times p_r + |W_j| \times p_j$ **then**
      tempSum $\leftarrow$ sum + $|W_r| \times p_r$
      $W_i \leftarrow W_r$, $L_i \leftarrow L_r$, $p_i \leftarrow p_r$
    end if
  end if
  sum $\leftarrow$ tempSum
  $i \leftarrow i - 1$
until sum + $|L_i| < K$ && $L_i \neq \phi$ && $i > 2$
$s \in W_i$ serves p_i leaf-peers
$s \in L_i$ serves 1 leaf-peer

$$\begin{aligned} &\text{maximize} && \sum_{j=1}^{M} w_j p_j \\ &\text{subject to} && \sum_{j=1}^{M} w_j = N, \quad w_j \geq 1. \end{aligned} \tag{1}$$

This is an integer programming problem that is not easy to solve. In Section V, we use the exhaustive search method to get the solution, which is subsequently compared to the results of our heuristic algorithm.

B. Heuristic algorithm

We propose two criteria to decide the winners of the n-winner $(n+1)^{th}$-price auction. The first criterion is to maximize the number of served leaf-peers $w_j p_j$ through the quality Q_j auction. In other words, it is a greedy algorithm that does not consider the next auction. It is due to the fact that the valuation of super-peers decreases with the video quality worse. We call the auction with this criterion *m-auction*.

The second one is that the fixed ratio r of super-peers become winners. If valuation decrease for quality degradation is relatively small, the performance of m-auction is quite bad. It is because the price for high quality is set at a relatively low level. In this case, a portion r of bidders become the winners of the auction. We call this *r-auction*.

Our proposed heuristic algorithm applies both of the criteria to maximize the total number of served leaf-peers. The server receives the valuation information for quality Q_{j-1} as well as Q_j. Then it chooses either m-auction or r-auction for quality Q_j depending on the auction outcome for quality Q_{j-1}. This process is repeated until the view-tickets for all quality levels are auctioned off or all the leaf-peers are served. The heuristic algorithm is generalized as shown in algorithm 1 whose complexity is $O(N^2M)$.

Example 1: Let's consider an example where N is 10 and M is 5. The valuation for each video quality by each super

TABLE II
TEN SUPER-PEERS' VALUATION FOR FIVE VIDEO QUALITY LEVELS

Bidder	1	2	3	4	5	6	7	8	9	10
Q_5	100	90	80	70	60	50	40	30	20	10
Q_4	95	85	75	65	55	45	35	25	15	5
Q_3	30	27	24	21	18	15	12	9	6	3
Q_2	1	1	1	1	1	1	1	1	1	1
Q_1	0	0	0	0	0	0	0	0	0	0

peer is shown in Table II. If we apply m-auction for quality Q_5, 5 super-peers become winners at the price of 50. So 255 (=5×50+5×1) leaf-peers can be served. If we apply r-auction instead, 1 super-peer becomes a winner at the price of 90. In this case, m-auction for quality Q_4 supports 180 (=4×45) leaf-peers, which results in the total of 275 (=1×90+4×45+5×1) leaf-peers served. So the server applies r-auction for quality Q_5. Subsequently, m-auction for Q_4 gives 185 (= 4×45+5×1) and r-auction gives 127 (=$1 \times 75 + 4 \times 12 + 4 \times 1$). Then the server applies m-auction for quality Q_4.

C. Quality differentiation method

Once the video quality of super-peers is determined, the server needs a method to enforce them not to watch the higher quality video than what is assigned. In the previous P2P streaming systems, peers exchange missing descriptions with each other to increase video quality [4], [5]. In SALSA, the server generates overlay multicast trees among super-peers to enforce video quality. Overlay multicast trees make it feasible for video delivery to meet the fairness [1] and not incur the control overhead, such as update and request message exchanges needed in the conventional systems. Therefore, adhering to the trees is one of the most viable strategies adopted for super-peers[1].

We implement CoopNet [7] for the overlay multicast tree generation in SALSA. CoopNet is originally designed for the resilient P2P streaming. Video is encoded using MDC and each description is transmitted through a different tree. A peer becomes an interior node only in one tree, and becomes a leaf node in the other trees. Thus, even though a peer does not work properly, the number of missing descriptions of the other peers is less than one.

In SALSA, if the number of descriptions is M, the server generates M overlay multicast trees, through which each description is transmitted. Here a super-peer having a quality Q_m ticket becomes an interior node in a tree, and the number of its children is limited by m. It also becomes a leaf-node in other $(m-1)$ trees. Then the super-peer uploads at most m times to download m descriptions. Fig. 2 shows the example of the overlay multicast trees where video is encoded into 4 descriptions, and super-peer 1∼5 gets quality Q_4, 6∼8 gets Q_3, and 9∼10 gets Q_2.

The server generates and manages the trees. The centralized approach is more efficient than the distributed one but contains a single point of failure. But in SALSA, if the server fails, the

[1]The collusion between super-peers is out of the scope in this paper.

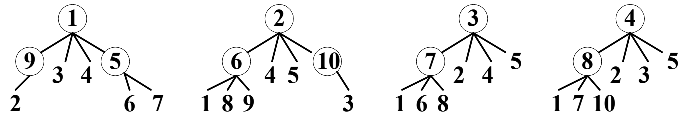

Fig. 2. Video is encoded into 4 descriptions, and super-peer 1∼5 gets quality Q_4, 6∼8 gets Q_3, and 9∼10 gets Q_2. The circle indicates the interior nodes.

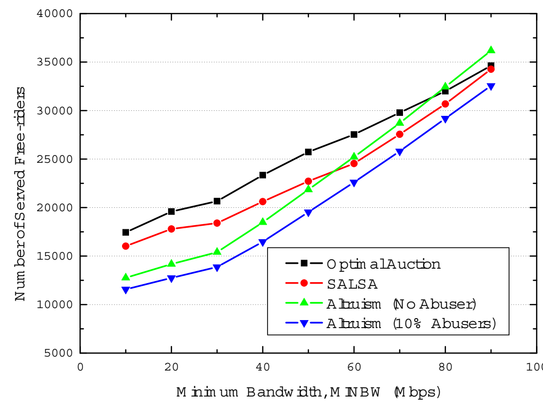

Fig. 3. Number of served free-riders vs. $MINBW$.

streaming service cannot be provided. So the tree management failure is not a major concern in that case. The overhead of the server increases as the number of super-peers gets larger. Let's consider the case with 10,000 super-peers and 20 descriptions. We assume that all the super-peers watch a quality Q_{20} video. Even if we include an 8 byte parent node pointer and 24 byte auxiliary data for every node of trees, only $(8+24) \times 10,000 \times 20$ = 6.4MB memories are needed. Since bandwidth and CPU usage are also very minimal for any modern machine [7], the centralized approach has scalability.

V. Performance Evaluation

In simulations, the video is divided into 30 groups of frames (GOFs) for 1 second, and a frame is encoded into 20 descriptions, and each description has the bitrate of 50kbps. The server bandwidth is 1Mbps, which is fully utilized to serve super-peers. Bandwidth B_i for super-peer i is uniformly distributed in the interval $[MINBW, MAXBW]$. It is assumed that in order to get quality Q_{20}, each super-peer fully utilizes its bandwidth except for 1Mbps which is left for the overlay multicast tree. We also assume that the number of super-peers is fixed.

Super-peer i decides the valuation K_{ij} for each quality $Q_j (1 \le j \le 20)$. At least quality Q_2 is guaranteed for a super-peer which serves only 1 leaf-peer. The valuation K_{ij} is modeled by the following concave function.

$$K_{ij} = K_{i20} \frac{1 - e^{-aj}}{1 - e^{-a \cdot 20}} = 20(B_i - 1) \frac{1 - e^{-aj}}{1 - e^{-a \cdot 20}},$$

where a is a parameter indicating the concave level.

Fig. 3 depicts the maximum number of served free-riders (leaf-peers) for $MINBW$ when there are 20 super-peers. $MAXBW$ is set at 100Mbps. *Optimal auction* represents the solution of (1) obtained by the exhaustive search method. The

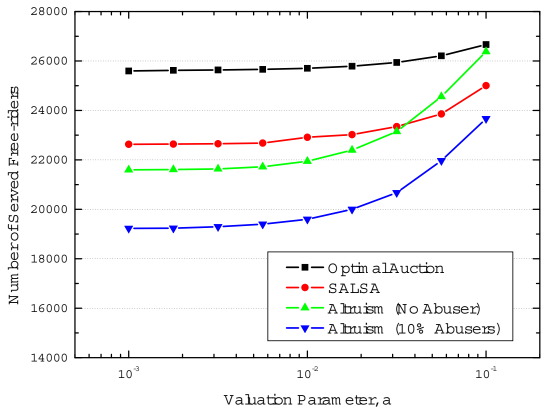

Fig. 4. Number of served free-riders vs. parameter a.

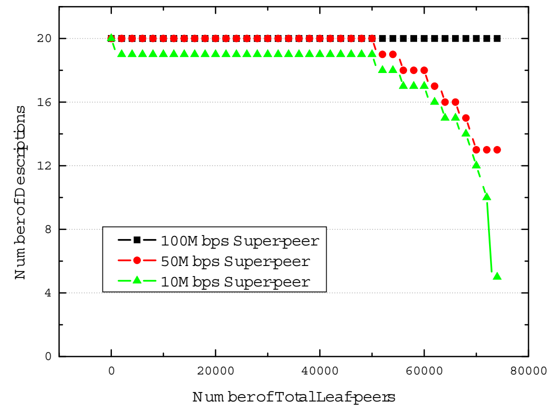

Fig. 5. Video quality of super-peers vs. number of total leaf-peers.

performance of SALSA is comparable to the optimum one with the proximity of at least 88%. *Altruism* [3] accommodates free-riders using surplus bandwidth of altruistic peers. Super-peers in SALSA can be regarded as a kind of altruistic peers. The server enforces the level of altruism K, which is the ratio of upload to download rate of altruistic peers. When the bandwidth gap between altruistic peers is large, fixing K cannot reflect their heterogeneity, leading to the degraded efficacy. It performs better than ours when altruistic peers are homogeneous in their bandwidth. But if some altruistic peers are nor obedient to the protocol, performance becomes different. For instance, if 10% of the altruistic peers are abusers, the number of served free-riders decreases by 10%, resulting in worse performance compared to SALSA. On the contrary, in SALSA, adhering to the protocol is the best strategy to super-peers.

Fig. 4 depicts the maximum number of served free-riders with respect to parameter a. When a gets larger, the valuation decrease for quality degradation gets smaller. $MINBW$ is set at 50Mbps. The results show that SALSA supports at least 88% of leaf-peers that can be supported by the optimum allocation. Altruism performs better than ours when a is large. But, if 10% of the altruistic peers are abusers, the maximum number of served free-riders in Altruistic protocol is smaller than that in SALSA.

In Fig. 5, the video quality of super-peers is described according to the number of total leaf-peers. Here we consider 91 super-peers with bandwidth 10, 11, $\cdots$, 100Mbps, respectively. a is set at 0.01. Under this condition, 73,302 leaf-peers can be served at maximum. When there is no leaf-peer, all super-peers watch quality Q_{20}. If some leaf-peers are added, each super-peer's video quality is determined according to the number of its serving leaf-peers. Therefore, the 100Mbps super-peer can always watch the video of quality Q_{20}. However, the quality of the 10Mbps super-peer decreases to Q_{19} when there are 91 leaf-peers. With the number of leaf-peers, the competition among super-peers becomes stronger, so the video quality of super-peer with low bandwidth degrades. As shown in Fig. 5, SALSA encourages super-peers to contribute to reducing the server load by uploading more to obtain a higher quality of video.

VI. Conclusions

In this paper, we proposed a novel P2P live streaming system called SALSA. In SALSA, the streaming server can accommodate leaf-peers with the aid of super-peers. In return, the super-peers can obtain high-quality-view-tickets as incentives. Using novel auction mechanisms and a heuristic algorithm, the server can support a comparable number of leaf-peers to the optimum level. Through simulations, we confirmed that a super-peer's video quality can be defined by the number of its serving leaf-peers, which becomes its contribution to the streaming server.

Acknowledgment

This research is supported by the Ubiquitous Computing and Network (UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy (MKE) in Korea as a result of UCN's subproject 09C1-C1-20S.

References

[1] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and improving a BitTorrent networks's performance mechanisms," *Proceedings of IEEE INFOCOM*, April, 2006.
[2] Y. Chu, J. Chuang, and H. Zhang, "A case for taxation in peer-to-peer streaming broadcast," *Proceedings of ACM SIGCOMM PINS*, August, 2004.
[3] Y. Chu and H. Zhang, "Considering altruism in peer-to-peer internet streaming broadcast," *Proceedings of ACM NOSSDAV* June, 2004.
[4] Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang, "P2P video live streaming with MDC: Providing incentives for redistribution," *Proceedings of IEEE ICME*, July, 2007.
[5] Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang, "Using layered video to provide incentives in P2P live streaming," *Proceedings of ACM SIGCOMM P2P-TV*, August, 2007.
[6] M. J. Osborne, "An introduction to game theory," *Oxford University Press*, 2004.
[7] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," *Proceedings of IEEE ICNP*, November, 2003.
[8] G. Tan and S. A. Jarvis, "A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast," *Proceedings of IEEE IWQoS*, June, 2006.
[9] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proceedings of the IEEE*, pp. 57-70, Jan. 2005.