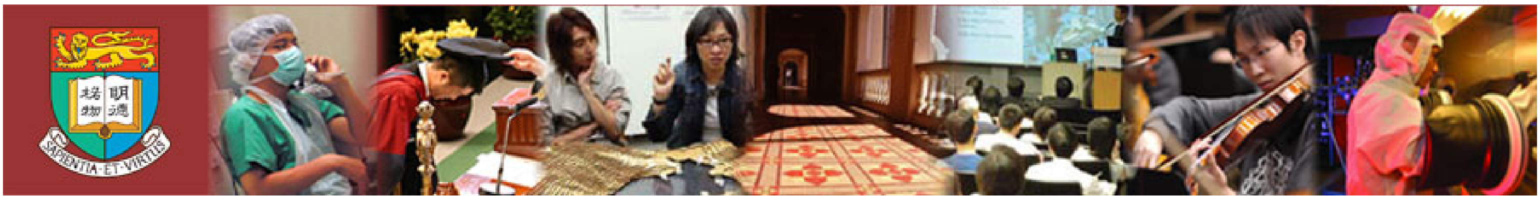


Efficient and Flexible Inter-Overlay Scheduling of Media Streams for Multi-Channel P2P Streaming

Xin Jin and Yu-Kwong Kwok
Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam, Hong Kong SAR

***Abstract*—Existing studies on channel bandwidth imbalance in P2P multi-channel streaming systems have been exclusively focused on inter-overlay bandwidth allocation. However, an efficient inter-overlay scheduling algorithm is still in lack for benefactors. To this end, this paper presents an inter-overlay substream scheduling algorithm compatible with various overlay meshes for active inter-overlay cooperation, through which the outbound bandwidth of benefactors can be efficiently utilized and bandwidth-deprived channels receiving benefactions can attain a better streaming quality.**

***Index Terms*—Peer-to-peer multimedia streaming, active multichannel cooperation, inter-overlay chunk scheduling, unstructured meshes**

I. Introduction

The strength of peer-to-peer (P2P) design, compared with the client-server architecture, is the capacity to efficiently explore distributed client resources and henceforth reduce the economic cost incurred to media service providers. Specifically, peers participating in the same channel formulate a swarm and reciprocate each other by scheduling chunks cooperatively. Hereafter, the terms *overlay* and *channel* are used interchangeably to represent the dissemination of one specific media content.

However, channel heterogeneity, incurred by autonomous peer participation, inevitably leads to disparate service quality experienced by various channels. The situation is further aggravated in relatively unpopular channels because statistically enough bandwidth provision cannot be consistently guaranteed [1]. In particular, measurement study of PPLive reveals significant bandwidth supply variations in diverse channels [2].

Thus, an intuitive idea is that peers within a bandwidth-rich channel $c_r \in \mathcal{C}$ ($\mathcal{C}$ is the set of channels existing in the system) altruistically join a bandwidth-deprived channel $c_d \in \mathcal{C}$ and serve as benefactors (i.e., benefaction peers) by feeding residential peers of c_d (c_d is the secondary overlay of benefactors). To make this happen, the following three open questions should be properly answered for a peer p in c_r:

- *Whether to serve as a benefactor and if so, which channel to benefact?* Such choices depend on the bandwidth availability of peer p. Moreover, considering peer rationality, incentive provision for benefactors is desired to motivate inter-overlay contribution.
- *How to allocate its outbound bandwidth?* Benefactor p has to resolve the bandwidth allocation between c_r and c_d — an essential step to optimally reallocate available outbound bandwidth resources existing in various channels. This dictates the streaming quality of both c_r and c_d.
- *How to efficiently schedule chunks?* That is, how to schedule chunks so that the outbound bandwidth allocated to c_d is fully exploited to maximally enhance the streaming quality of channel c_d? In this paper, we term the chunk scheduling operation of benefactors to help and relay chunks to residential peers *inter-overlay scheduling*. Correspondingly, the chunk scheduling operation of residential peers is referred to as *intra-overlay scheduling*.

Several studies [2], [3] exist in the literature to study the bandwidth allocation problem. However, there is no such an inter-overlay scheduling algorithm with the flexibility to couple with various intra-overlay scheduling designs. To this end, this paper is to design a robust and configurable inter-overlay scheduling algorithm with both such flexibility and the bandwidth utilization efficiency to fully leverage allocated outbound bandwidth for the purpose of inter-overlay benefaction. To the best of our knowledge, this is the *first* attempt to achieve this design objective. The non-trivial difference between inter-overlay scheduling and intra-overlay scheduling implies the design difficulty.

The reminder of this paper proceeds as follows. Section II presents recent advances in P2P multimedia streaming. Section III formulates the inter-overlay scheduling problem and illustrates our proposed inter-overlay substream scheduling protocol. In Section IV, we present performance evaluation methodology. Simulation results are presented in Section V. Finally, Section VI concludes this paper.

II. Related Work

There exists a large body of literature on mesh based P2P overlays and intra-overlay scheduling has been extensively investigated [4], [5]. Specifically, peer i determines how to select neighbor j to request chunk a from the set of neighbors possessing chunk $a \notin C_i$, namely $N_i^a = \{p|(p \in N_i) \wedge (a \in C_p)\}$, where N_i is the neighbor set of peer i and C_p is the set of chunks in p's buffer.

- Random. Peer i randomly selects $j \in N_i^a$.
- Least Load [4]. Peer i selects j with the maximum value of outbound bandwidth over sending queue size. That is,

$$j = \arg\max_{p \in N_i^a} \frac{O_p}{l_p},$$

978-1-4673-0009-4/12/$26.00 ©2012 IEEE

where l_j is the sending queue length of peer j.

- Bandwidth Aware [5]. The probability that i selects j is

$$\pi_i(j) = \frac{O_j}{U_i},\ U_i = \sum_{p \in N_i^a} O_p.$$

However, not until recently, inter-overlay cooperation has become an important topic to improve system performance. Liao *et al.* [6] propose to utilize inter-overlay path optimization by constructing efficient paths using peers from different content overlays. An auction based mechanism [3] is designed by Wu *et al.* to resolve conflicts among coexisting streaming overlays for efficient bandwidth allocation. Wu *et al.* [2] implement the idea of View-Upload-Decoupling (VUD) by dividing media content into independent substreams, whose design resembles multiple tree streaming. Peers in VUD are required to retrieve a substream from a non-residential channel for uploading purposes. A theoretical study based on queueing theory is performed in [1], further demonstrating the strength of inter-overlay design. Recently, Wang *et al.* [7] formulate inter-overlay bandwidth sharing as a utility maximization problem, performed distributively by each peer.

III. Inter-Overlay Scheduling: Designs and Implementations

A. Problem Formulation

The impact of bandwidth provision. Bandwidth provision is critical for channel streaming quality. Thus, we firstly define resource index ρ_c to quantitatively measure the outbound bandwidth provision of channel c:

$$\rho_c = \frac{\sum_{p \in \mathcal{P}_c} O_p + O_s^c}{|\mathcal{P}_c| \cdot s_c},$$

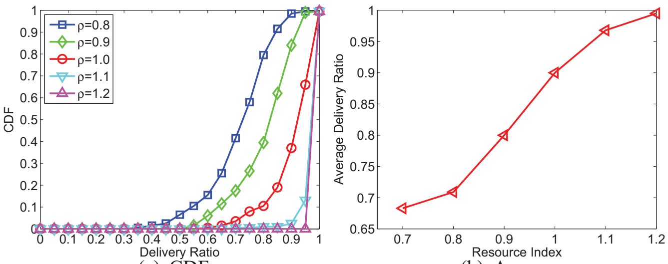
where $\mathcal{P}_c$ is the set of peers participating in channel $c \in \mathcal{C}$, O_p and O_s^c are respectively the outbound bandwidth of peer $p \in \mathcal{P}_c$ and the server of channel c. Denote by $\mathcal{C}$ and s_c respectively the set of coexisting channels and the streaming rate of channel c. Similarly, the resource index after inter-overlay benefaction is

$$\rho_c^b = \frac{\sum_{b \in \mathcal{B}_c} O_b + \sum_{p \in \mathcal{P}_c} O_p + O_s^c}{|\mathcal{P}_c| \cdot s_c},$$

where $\mathcal{B}_c$ is the set of benefactors residing in channel c. Note that we have not counted in the outbound bandwidth overhead incurred by benefactors. We omit the subscript c in the following discussion. Then, the bandwidth benefaction obtained from benefactors can be quantitatively measured by

$$h = \rho^b - \rho.$$

To illustrate the impact of channel bandwidth provision on streaming quality, we implement the least load scheduling algorithm with the capacity to achieve near-optimal streaming quality in systems with enough bandwidth provision [4]. Streaming quality in terms of delivery ratio under different resource index scenarios is shown in Fig. 1(a) and Fig. 1(b) with the simulation environment depicted in Section IV. Obviously, better streaming quality can be attained with the increase of resource index and bandwidth deficit significantly degrades the streaming quality. Consequently, we can learn the importance of inter-overlay bandwidth sharing and the necessity of inter-overlay scheduling.



(a) CDF (b) Average

Fig. 1. The impact of outbound bandwidth provision on the streaming quality in terms of delivery ratio evaluated by a capacity-aware load balancing pull scheduling scheme.

Design objective of inter-overlay scheduling. However, inter-overlay scheduling is different from intra-overlay scheduling in that benefaction peers do not individually need chunks retrieved from the secondary overlay. This inevitably results in *outbound bandwidth overhead* incurred by chunk retrieval before the possibility of bandwidth benefaction. Notice that it is not the signalling overhead for overlay maintenance, but the bandwidth consumption of peers to upload chunks to benefactors in the secondary overlay.

Thus, the optimal inter-overlay scheduling problem can be formally modeled as

$$\max\ \{\sum_{b \in \mathcal{B}} (u_b - d_b)\}, \qquad (1)$$

subject to

$$d_b \le D_b,\ u_b \le O_b,$$

where D_b and O_b are respectively the inbound bandwidth limit and the outbound bandwidth limit allocated to benefactor b for inter-overlay benefactions. d_b and u_b respectively represent the actual inbound and outbound consumption, and $\mathcal{B}$ is the total set of benefactors existing in the system. Simply put, it is desirable for benefactors to download less while contributing more in the SO. If each benefaction peer possesses perfect information about the demand and supply of each chunk, peers can solve the above problem by recursively retrieving chunks with the highest demand. However, network decentralization renders this problem extremely difficult.

In this paper, to evaluate the bandwidth utilization efficiency of inter-overlay scheduling, we define the *benefaction ratio* for each benefactor b

$$R_b = \frac{u_b}{d_b}, \quad \forall b \in \mathcal{B}.$$

Considering peer dynamics and the fact that R_b depends on both chunk availability and chunk demands in the neighborhood of benefactors, we resort to heuristics for inter-overlay scheduling design.

B. Inter-Overlay Substream Scheduling

An intuitive idea is to leverage on substream scheduling, because peers only need to retrieve part of the stream for benefaction purposes. Due to both our assumption that outbound bandwidth constraint is the main bottleneck faced by resource-poor overlays and the fact that pull-scheduling can complement the push mode by retrieving chunks not timely pushed and further enhancing the streaming quality [8], we utilize push-pull inter-overlay substream scheduling to achieve a high benefaction ratio.

To minimize the outbound bandwidth overhead, each benefactor subscribes to one single substream and relays the received chunks to its neighbors. Each benefactor randomly selects a substream when joining the bandwidth-deprived channel and retrieves the corresponding chunks in the very substream. The download sliding windows of benefactors are updated in buffer map exchanges just as residential peers. As discussed above, we utilize the push-pull mode, which not only integrates the push-mode compatible with the assumption that the uplink-constraint is the bottleneck, but also adopts the pull-mode as a fallback strategy.

Substream k is defined as the set of chunks with sequence number S_n satisfying

$$S_n \% K = k,$$

where K is the total number of substreams existing in the system. Upon receiving a chunk from a benefactor, the residential peer will subscribe to the benefactor. Each benefactor will push chunks to all the neighbors subscribing to its substream. And the neighbor degree of benefactor b is determined by

$$Nbr = \alpha \cdot K \cdot \frac{O_b}{s},$$

where Nbr is the neighbor degree, K is the number of substreams, and s is the streaming rate. System parameter α can be utilized to adjust the neighbor degree.

In our design, residential peers randomly select its neighbors from both residential peers and benefactors. However, benefactors' neighborhood only consists of residential peers in that their task is to help residential peers instead of benefactors themselves. Moreover, only if two benefactors share the same substream number, can they reciprocate to each other.

Compared with VUD [2], our design can achieve lower outbound bandwidth overhead due to the fact that VUD requires every peer retrieve one substream from non-residential channels, while our design only desires such action from benefactors. Moreover, our design possesses the flexibility to fit into various chunk-based streaming systems.

IV. Performance Evaluation

We now proceed to evaluate the inter-overlay substream scheduling algorithm. In this section, we respectively describe the simulation setting, simulation methodology and metrics utilized to evaluate system performance.

A. Simulation Setting

We explore and evaluate the inter-overlay substream scheduling algorithm stated above through simulations, based on the P2P media streaming simulator originally developed by Zhang [9] for intra-overlay scheduling. This is an event-driven packet-level C++-based simulator with the capacity to simulate a maximum of $10,000$ peers simultaneously joining a single overlay.

Unless otherwise stated, simulation results correspond to the simulation setting given in Table I. Note that the setup of the three types of nodes is for residential peers. The resource index ρ without bandwidth benefactor is 0.9.

TABLE I
Simulation Setting

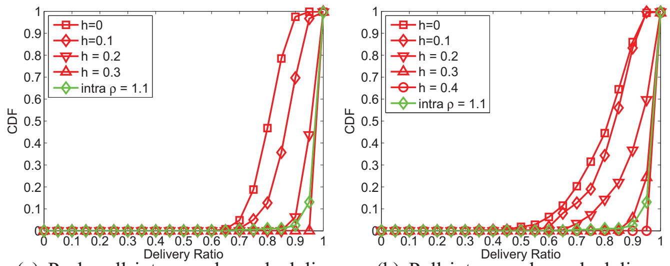
Neighbor Size of Residential Peers	20
Neighbor Size of Benefactors	10
Streaming Rate	300 Kbps
Server Upload Capacity	600 Kbps
Type 1 Node Capacity (Up/Down)	1000 Kbps / 3000 Kbps
Type 2 Node Capacity (Up/Down)	384 Kbps / 1500 Kbps
Type 3 Node Capacity (Up/Down)	128 Kbps / 768 Kbps
Type 1 Node Fraction	0.1
Type 2 Node Fraction	0.214
Type 3 Node Fraction	0.686
Residential Peer Population	300
Benefactor Population	100
Intra-Overlay Scheduling	Random / Push-Pull

B. Simulation Methodology

We modify the simulator by deploying two kinds of peers in the simulator, one group of which represents the residential peers and the other group joins in the system as benefactors. Residential peers follow intra-overlay scheduling method to retrieve chunks they need. On the other hand, benefactors take the proposed inter-overlay scheduling strategy. Initially, we set the number of substreams to be 10 for the study of the algorithm's effectiveness to improve streaming quality and the impact of graph degree, peer dynamics, different intra-overlay scheduling schemes, and benefactor heterogeneity on system performance. Then, we vary the number of substreams to study its influence on system design.

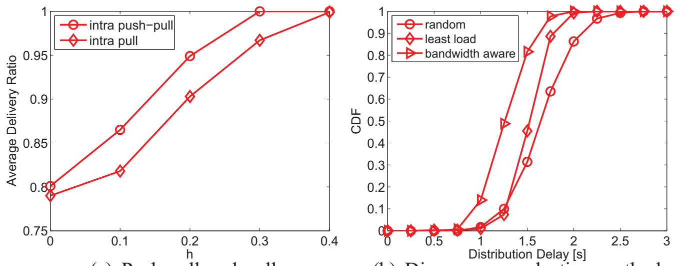
Static and dynamic simulation environments. In a static environment, nodes, including both residential peers and benefactors, join in one by one in the initialization period. After that, they will persist in the life time. Dynamics refer to peer churn (i.e., peer joining/leaving/failure, etc). In the dynamic environment, each node repeatedly joins and departs the system. The online and offline durations of each node per time are exponentially distributed, which incurs a high peer churn rate with an average online duration of 20 minutes.

Benefactor heterogeneity. We utilize the 3-class scenario to evaluate whether the inter-overlay scheduling logic still performs well when the overlay is built upon a heterogeneous network. The 3-class scenario is based on a 3-class bandwidth distribution: low-bandwidth peers with outbound bandwidth $0.5\overline{B}$, mid-bandwidth peers with outbound bandwidth $\overline{B}$, and high-bandwidth peers with outbound bandwidth $2\overline{B}$. The fraction of these 3 classes of peers is respectively $t/3$, $1-t$, and $2t/3$ from low to high, where t is the heterogeneity factor and $0 \leq t \leq 1$; thus, the average bandwidth is $2\overline{B} \cdot t/3 + \overline{B} \cdot (1-t) + 0.5\overline{B} \cdot 2t/3$. This scenario captures the most important properties of heterogeneous networks and has already been widely utilized in the literature [10].



(a) Push-pull intra-overlay scheduling (b) Pull intra-overlay scheduling

Fig. 2. The impact of inter-overlay substream scheduling on streaming quality.



(a) Push-pull and pull (b) Diverse peer selection methods

Fig. 3. The impact of intra-overlay scheduling.

Neighbor degree. We also study the impact of the neighbor degree of benefactors on streaming quality. To focus on the impact of graph degree, the benefactors are homogeneous with outbound bandwidth 270 Kbps and we hold the intra-overlay scheduling algorithm to be push-pull. We also hold the neighbor degree of residential peers constant (i.e., 20).

C. Performance Metrics

We define the following metrics to measure the streaming quality of peers:

- **Distribution delay.** By distribution delay, we mean the elapsed time from the instant a chunk is generated by the source to the moment it is received by a peer.
- **Delivery ratio.** Aside from distribution delay, to quantitatively evaluate the streaming quality, we define delivery ratio to represent the number of chunks arriving at each node before or on playback deadline over the total number of chunks that each node should receive.

V. Simulation Results

In this section, we present our simulation results following the simulation methodology described above.

A. The Effectiveness of Inter-Overlay Substream Scheduling

We study the effectiveness of bandwidth benefaction via the delivery ratio distributions under different values of h (cf. Section III). As a comparison baseline, we also provide the delivery ratio distribution for intra-overlay scheduling with $\rho = 1.1$. Fig. 2(a) shows that after inter-overlay bandwidth benefaction, system performance can be improved.

Fig. 2(b) shows that our substream inter-overlay scheduling scheme can also improve the system performance with pull intra-overlay scheduling. However, to achieve similar streaming performance, the pull scheme requires more benefaction bandwidth than the push-pull scheme. Indeed, the push-pull intra-overlay scheduling scheme can leverage the system bandwidth more efficiently. In Fig 3(a), we explicitly compare the average delivery ratio when we utilize the two intra-overlay schemes under different values of h. It reveals that our inter-overlay scheduling couples better with push-pull intra-overlay scheduling.

B. The Impact of Intra-Overlay Peer Selection Methods.

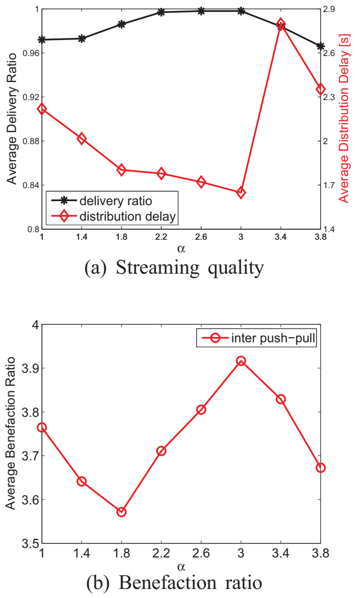
Fig. 3(b) compares three different peer selection schemes used for intra-overlay scheduling via distribution delay: random, least load, and bandwidth aware. The simulation setup is $\alpha = 3.0$ and $h = 0.3$. The average distribution delay of these three schemes are respectively $1649ms$ (random), $1522ms$ (capacity aware), and $1262ms$ (bandwidth aware). The bandwidth aware scheme performs best in terms of distribution delay in that the node heterogeneity is properly considered.

C. The Impact of Graph Degree.

Fig 4(a) illustrates that initially the streaming quality improves in terms of both delivery ratio and distribution delay with the increase of the neighbor degree of benefactors. However, when the neighbor degree is large enough, the streaming quality degrades with the increase of neighbor degree. This may result from the fact that when neighbor degree is small, benefactors can not effectively relay chunks to enough residential peers. Then, the increase in neighbor degree can improve the streaming quality. However, when the neighbor degree is large enough, the proportion of benefactor neighbors of residential peers increases, which may affect effective intra-overlay scheduling, considering the fixed neighbor degree of residential peers. Fig 4(b) illustrates the variation of benefaction ratio with respect to α, which shows that the benefaction ratio of the system is around 3.8.

D. The Impact of Dynamic Environment.

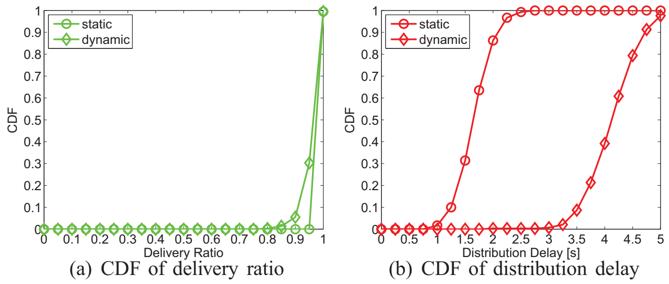
To further study the performance of the inter-overlay scheduling substream algorithm, we compare the scheduling performance under both static and dynamic environment. We setup the following parameters: $\alpha = 3.0$, and $h = 0.3$. Fig. 5(a) shows us the comparison of delivery ratio and Fig. 5(b) shows us the comparison of distribution delay. The average distribution delays in static and dynamic environments are respectively $1649ms$ and $4245ms$. We observe that the inter-overlay substream scheduling algorithm still performs well in the dynamic environment.



(a) Streaming quality

(b) Benefaction ratio

Fig. 4. The impact of the neighbor degree of benefactors on streaming quality.



(a) CDF of delivery ratio

(b) CDF of distribution delay

Fig. 5. Comparison between the static and dynamic environments.

E. The Impact of Benefactor Heterogeneity.

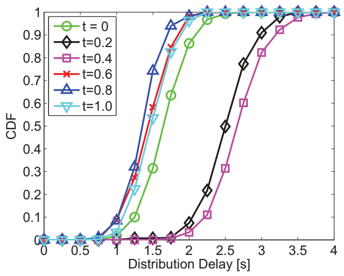
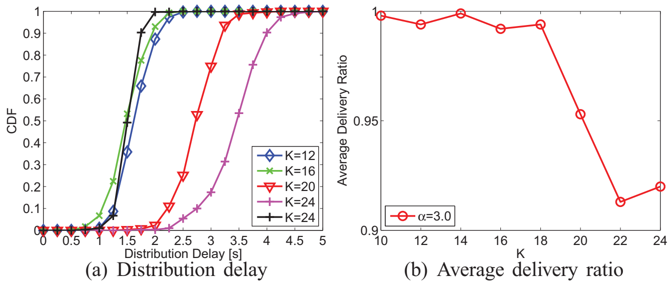


Fig. 6. Benefactor bandwidth heterogeneity study.

Fig. 6 shows the impact of benefactor heterogeneity under differen heterogeneity factors utilizing random scheduling with the setup: $\alpha = 3.0$, $h = 0.3$ and $\overline{B} = 270Kbps$. To focus our study on the effects of benefactor heterogeneity, the bandwidth distribution of residential peers has not been reconfigured in the study of benefactor heterogeneity. It reveals that we have the worst performance when t is 0.2 or 0.4.

F. Impact of the Number of Substreams.

Similarly, we consider the scenario of $\alpha = 3.0$ and $h = 0.3$ with random scheduling. The performance degradation shown in Fig. 7(a) and Fig. 7(b) when K is no smaller than 20 may result from the fact that the neighbor degree of benefactors



(a) Distribution delay

(b) Average delivery ratio

Fig. 7. The impact of the number of substreams on system design.

is too large. That is, α is too large for these values of K. To illustrate this point, the black line in Fig. 7(a) shows the distribution delay when $\alpha = 2.6$ and $K = 24$. In this scenario, the average delivery ratio is 0.992. This means we need to carefully configure α for specific values of K to achieve optimal system performance.

VI. Conclusion

In this paper, we present an inter-overlay scheduling substream algorithm for active benefactors in multichannel P2P multimedia systems. Extensive simulations demonstrates its efficacy to improve the streaming quality of bandwidth-deprived channels. We also study the impact of graph degree, peer dynamics, benefactor heterogeneity and the substream number on system performance.

Our proposed algorithm can be flexibly coupled with existing P2P overlay meshes, as verified via simulation of different intra-overlay scheduling algorithms. This proves its efficacy to assist existing inter-overlay bandwidth sharing schemes to further enhance bandwidth utilization efficiency. Moreover, we propose benefaction ratio to evaluate the bandwidth utilization efficiency. Incentive provision for benefactors and more efficient bandwidth allocation schemes are in prospect.

References

[1] D. Wu, Y. Liu, and K. W. Ross, "Queueing Network Models for Multi-Channel P2P Live Streaming Systems," in *Proc. of IEEE INFOCOM*, April 2009.

[2] D. Wu, C. Liang, Y. Liu, and K. Ross, "View-Upload Decoupling: A Redesign of Multi-Channel P2P Video Systems," in *Proc. of IEEE INFOCOM Mini-Conference*, April 2009.

[3] C. Wu, B. Li, and Z. Li, "Dynamic Bandwidth Auctions in Multi-overlay P2P Streaming with Network Coding," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 6, pp. 806–820, June 2008.

[4] Y. Yang, A. L. Chow, L. Golubchik, and D. Bragg, "Improving QoS in BitTorrent-like VoD Systems," in *Proc. of IEEE INFOCOM*, March 2010.

[5] A. P. C. da Silva, E. Leonardi, M. Mellia, and M. Meo, "A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems," in *Proc. of IEEE P2P*, September 2008.

[6] X. Liao, H. Jin, Y. Liu, and L. M. Ni, "Scalable Live Streaming Service Based on Interoverlay Optimization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1663–1674, December 2007.

[7] M. Wang, L. Xu, and B. Ramamurthy, "A Flexible Divide-And-Conquer Protocol for Multi-View Peer-to-Peer Live Streaming," in *Proc. of IEEE P2P*, September 2009.

[8] M. Zhang, J.-G. Luo, L. Zhao, and S.-Q. Yang.

[9] M. Zhang, "Peer-to-Peer Streaming Simulator," 2011, http://media.cs.tsinghua.edu.cn/∼zhangm/.

[10] L. Abeni, C. Kiraly, and R. L. Cigno, "Robust Scheduling of Video Streams in Network-Aware P2P Applications," in *Proc. of IEEE ICC*, April 2010.