University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

CSE Conference and Workshop Papers | Computer Science and Engineering, Department of

1-1-2010

Linear Programming Models For Multi-Channel P2P Streaming Systems

Miao Wang
University of Nebraska - Lincoln, mwangcse@gmail.com

Lisong Xu
University of Nebraska - Lincoln, xu@cse.unl.edu

Byrav Ramamurthy
University of Nebraska - Lincoln, bramamurthy2@unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/cseconfwork

Part of the Computer Sciences Commons

Wang, Miao; Xu, Lisong; and Ramamurthy, Byrav, "Linear Programming Models For Multi-Channel P2P Streaming Systems" (2010). *CSE Conference and Workshop Papers.* Paper 110.
http://digitalcommons.unl.edu/cseconfwork/110

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

Linear Programming Models For Multi-Channel P2P Streaming Systems

Miao Wang, Lisong Xu and Byrav Ramamurthy
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115 USA
{mwang, xu, byrav}@cse.unl.edu

***Abstract*—Most of the commercial P2P video streaming deployments support hundreds of channels and are referred to as multi-channel systems. Measurement studies show that bandwidth resources of different channels are highly unbalanced and thus recent research studies have proposed various protocols to improve the streaming qualities for all channels by enabling cross-channel cooperation among multiple channels. However, there is no general framework for comparing existing and potential designs for multi-channel P2P systems. The goal of this paper is to establish tractable models for answering the fundamental question in multi-channel system designs: Under what circumstances, should a particular design be used to achieve the desired streaming quality with the lowest implementation complexity? To achieve this goal, we first classify existing and potential designs into three categories, namely Naive Bandwidth allocation Approach (NBA), Passive Channel-aware bandwidth allocation Approach (PCA) and Active Channel-aware bandwidth allocation Approach (ACA). Then, we define the bandwidth satisfaction ratio as a performance metric to develop linear programming models for the three designs. The proposed models are independent of implementations and can be efficiently solved due to the linear property, which provides a way of numerically exploring the design space of multi-channel systems and developing closed-form solutions for special systems.**

I. Introduction

Commercial Peer-to-Peer (P2P) live streaming and Video-On-Demand (VoD) systems, such as PPLive [1], UUSee [2], Joost [3], have been successfully supporting tens of thousands users[1]. Besides the large number of users, these systems have a common feature of providing a large number of channels for users to watch, and hence are referred to as multi-channel P2P streaming systems in this paper.

According to recent studies [4] [5], the upload bandwidth is important for P2P streaming systems, which implies that sufficient upload bandwidth (i.e., bandwidth supply is greater than or equal to bandwidth demand) of a specific channel is necessary for maintaining good streaming qualities for users watching that channel. However, measurement studies [6] [7] show that in multi-channel P2P streaming systems, upload bandwidth resources of different channels are highly unbalanced. In other words, some channels suffer bad streaming qualities due to bandwidth deficit in these channels; while some other channels have unused surplus bandwidth.

[1]We use the terms *user* and *peer* inter-changeably in this paper.

One straightforward idea is to utilize surplus bandwidth in the system to improve streaming qualities of channels suffering bandwidth deficit, which is referred to as cross-channel resource sharing. To enable cross-channel resource sharing, some peers should subscribe to[2] more than one channel at a time, which is realistic. For example, PPStream [8], one of the largest P2P streaming service providers in China, supports the Picture-in-Picture (PIP) function, where a user is allowed to watch a channel on the main window and monitor another channel on a smaller window. Wu *et al.* [9] propose a new View-Upload Decoupling (VUD) design for multi-channel P2P streaming systems, where a user might be required to subscribed to unwatched channels to contribute its upload bandwidth to these channels. Moreover, Wang *et al.* [10] and Wu *et al.* [11] study the bandwidth allocation problem of coexisting overlays. Please refer to Section II for detailed discussions on related work.

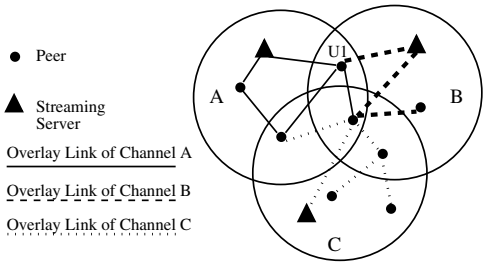

Fig. 1. A Three-Channel P2P Streaming System, where U_1 watches channels A and B.

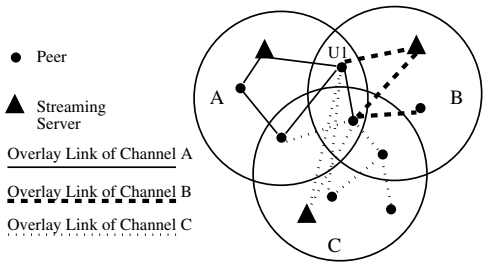

Fig. 2. A Three-Channel P2P Streaming System, where U_1 watches channels A and B but subscribes to the unwatched channel C as a helper.

[2]*Subscribing to a channel* means that a peer joins the overlay of that channel for data dissemination but may or may not watch that channel.

Digital Object Identifier: 10.1109/INFOCOM.2010.5462230

978-1-4244-5837-0/10/$26.00 ©2010 IEEE

We use Figures 1 and 2 to describe three potential designs for multi-channel P2P streaming systems that allow users to watch/subscribe to a *variable* number of channels. Figures 1 and 2 show a P2P streaming system with three channels and we assume that streaming rates of the three channels are the same. The Naive Bandwidth allocation Approach (*NBA*) requires users to subscribe to only their watched channels and allocate their upload bandwidth to watched channels proportional to their streaming rates. As shown in Fig 1, user U_1 subscribes to two channels A and B and evenly allocates its upload bandwidth to channels A and B respectively, if *NBA* is used, since streaming rates of the two channels are equal. With the Passive Channel-aware bandwidth allocation Approach (*PCA*), a user subscribes to only its watched channels and optimally allocates its bandwidth to its watched channels. The bandwidth allocation algorithm [11] for overlapping overlays and the protocol proposed in [10] are examples of *PCA* design. User U_1 in Fig 1 optimally allocates its upload bandwidth to channels A and B considering their bandwidth demands and supplies.

Active Channel-aware bandwidth allocation Approach (*ACA*) differs from *NBA* and *PCA* as follows: a user subscribes to not only its watched channels, but also maybe some other channels as a helper. A user optimally allocates its bandwidth to the watched channels and the subscribed but unwatched channels. For example, as shown in Fig 2, besides watched channels A and B, user U_1 also subscribes to unwatched channel C, since it has surplus bandwidth to help channel C. Note that the main difference between *PCA* and *ACA* is that *ACA* requires a user to subscribe to some channels that it does not watch and to allocate its bandwidth to the unwatched channels. View-Upload-Decoupling proposed in [9] is a special case of *ACA* design, since a peer is restricted to simultaneously watch one channel and might be selected by the system to subscribe to other channels as helpers.

Intuitively, *ACA* should perform better than *PCA*, since *ACA* can efficiently use all of its surplus bandwidth in a system. *PCA* should perform better than *NBA*, since *PCA* is aware of the bandwidth imbalance in a system. However, their implementation complexities also increase in the order of *NBA*, *PCA*, and *ACA* (refer to Section III-A for detailed implementation complexity and performance discussions). Consequently, when designing multi-channel systems, we must decide which design should be used by considering performance and complexity.

Although existing works propose different designs for building multi-channel P2P streaming systems, there are no general methods of comparing these designs. Therefore, important decision problems (i.e., under what circumstances, which design should be used to achieve the desired streaming quality with the lowest complexity?) cannot be answered by existing works. The motivation of this paper is to establish simple models for studying the intrinsic features of designing multi-channel P2P systems, which provides design guidelines.

In this paper, we propose a framework for comparing the three designs using the linear programming models. Specifically, for each of three designs, we define the bandwidth satisfaction ratio of a channel as the total bandwidth allocated by a specific design divided by the total bandwidth demand of that channel. Then, we use the bandwidth satisfaction ratio of a channel as a metric to evaluate the performance of that channel, which is independent of implementations of different designs. Finally, we model the problem of evaluating three designs as solving the corresponding linear programming problems.

The rest of this paper is organized as follows. Section II briefly summarizes the related work. Section III describes the models and insights of the three designs. Finally, we conclude the paper in Section IV.

II. Related Work

Most of the literature about P2P streaming systems focuses on improving the performance within a single channel (referred to as the single-channel P2P streaming systems). Tree-based overlay derived from IP multicast (e.g., Zigzag [12], [13]) is first used to build single-channel systems. However, the tree structure is not resilient to dynamics (e.g., peer joining/leaving the system randomly). Therefore, mesh-based overlays are widely used in commercial systems such as PPLive [1] and UUSee [2]. CoolStreaming [14] first introduces the data-driven design to P2P streaming systems, which has been proven to be powerful in real implementations. Generally speaking, all these designs aim to efficiently utilize peers' upload bandwidth for building scalable and robust single-channel P2P streaming systems.

Recently, P2P streaming systems where a user subscribes to more than one channel have emerged. Wu *et al.* [11] first study the bandwidth allocation problem of coexisting overlays. They propose a game theoretic approach to efficiently allocate bandwidth to different overlays. In our previous work [10], we propose a new divide-and-conquer strategy for building multi-view P2P streaming systems, which solves the inter-channel bandwidth competition and intra-channel streaming separately. The two works can be considered as special cases of *PCA* design.

Moreover, since most commercial P2P streaming systems support hundreds of channels and the bandwidth resources of different channels are unbalanced, recent studies propose cross-channel bandwidth sharing approaches to improve the performances of channels suffering bandwidth deficit. Wu *et al.* [15] study the problem of reducing server bandwidth consumption via cross-channel bandwidth sharing. Wu *et al.* [9] propose the View-Upload Decoupling (VUD) approach to minimize the influence of channel churn on multi-channel P2P streaming systems, where a user can contribute its upload bandwidth to unwatched channels. The same authors [16] also establish queueing network models for analyzing the system performance at the steady state, which only aims to model dynamic features for a specific design and does not provide a general framework for comparing different designs.

Zhang *et al.* [4] prove that the total bandwidth supply influences the performance of a single-channel system, where

Digital Object Identifier: 10.1109/INFCOM.2010.5462230

there is exactly one channel in the system. Kumar *et al.* [5] propose a stochastic model of system bandwidth for studying the performance limitations of a single-channel streaming system. In our previous work [17], we establish a simple processor-sharing queueing model to study how to provide stochastically guaranteed Quality of Service (QoS) for P2P streaming systems. Zhang *et al.* [18] develop an optimal block scheduling algorithm, called min-cost scheduling, which aims to maximally utilize the bandwidth of all overlay links. Massoulie *et al.* [19] also develop a network flow based model to study the decentralized broadcasting problems with bandwidth constraints and propose an optimal broadcasting algorithm. Therefore, the bandwidth satisfaction ratio defined in Section I is an ideal performance metric for each channel in the system, which is independent of different designs.

III. Modeling Multi-Channel P2P Streaming Design

In this section, we first propose three simple models to describe cross-channel resource sharing corresponding to the three designs. These models with the defined feasibility metrics provide us with a way to compare the three designs against various population distributions, upload bandwidth distributions and channel sets.

A. Implementation Complexity and Performance

The motivation of this paper is to establish tractable models for comparing performances of the three designs. Moreover, based on studies [4] [20], the system bandwidth greatly influences streaming qualities of all channels. Therefore, we use bandwidth to evaluate the performance of a specific channel, which is independent of different implementations.

Since cross-channel bandwidth sharing is the key component of three designs, the major difference of implementing these designs is different information required by the cross-channel bandwidth sharing. For example, the *ACA* design requires bandwidth information of all channels (i.e., it needs information of which channel has surplus bandwidth and which channel suffers bandwidth deficit), since it intends to fully utilize the system bandwidth to improve streaming qualities of the system. Therefore, it is the most complex design among the three designs.

B. Model definitions and notation

An important feature of a P2P system is user dynamics; that is, a user may randomly join or leave the system (referred to as peer churn), and change its watched channels (referred to as channel churn). In response to user dynamics, we divide the time axis into a series of short time intervals, and assume that during each interval the system is relatively stable.

The system with peers and their watched channels in each interval is defined as the system configuration for that time interval. Our models study various system configurations that occur in an interval.

Studying the system configurations in an interval is reasonable due to the following reasons. Recent P2P measurement studies [6], [21], [22] show that a P2P system is relatively stable over an interval of minutes, because most users have a lifetime longer than a minute, and at any time instant a significant percentage of users (e.g. $> 70\%$ on average reported in [22]) even have a lifetime on the order of hours. Moreover, we can use the queueing network models [16] to extend our model to capture peer dynamics.

We model the upload bandwidth allocation problem for *NBA*, *PCA* and *ACA* with respect to a given system configuration, since the cross-channel upload bandwidth sharing is the key issue in designing multi-channel P2P streaming systems. *Furthermore, we are interested in comparing the three designs with the same design goal of maximizing the bandwidth obtained by each channel, which is the direct or indirect design goal in most scenarios.* It does not make any sense to compare designs with different design goals (e.g. the designs with different user utilities). Therefore, we define the *bandwidth satisfaction ratio* of a channel as the total obtained upload bandwidth of that channel over the the channel's total bandwidth demand (a formal definition will be introduced below). The goal of all the three designs is to maximize the aggregated bandwidth satisfaction ratio of all channels, in that the upload bandwidth influences the performance of a P2P streaming system [4] [5] [16].

We start our model descriptions for the three designs by introducing the common notation used in all approaches.

- Let Θ be the set of all channels.
- Let $\theta \subseteq \Theta$ be a subset of channels.
- Let S_θ be the group of peers watching just channel set θ. That is, $S_\theta = \{m|\theta(m) = \theta\}$. Note that $S_{\theta_1} \cap S_{\theta_2} = \emptyset$ for $\theta_1 \neq \theta_2$. $\theta(m)$ denotes the channel set watched by peer m.
- Let u_m be the upload bandwidth of peer m.
- Let r_c be the streaming rate of channel $c \in \Theta$.
- Let x_c^θ denote the fraction of upload bandwidth that group S_θ allocates to channel $c \in \theta$.
- Let y_c^θ denote the fraction of upload bandwidth that group S_θ allocates to channel c not in θ. Note that $\sum_{c\in\theta} x_c^\theta + \sum_{c\notin\theta} y_c^\theta = 1$. Also note that for *NBA* and *PCA*, y_c^θ is always 0.
- Let γ_c denote the bandwidth satisfaction ratio of channel c, where γ_c is nonnegative and will be given for each design below.
- Let s_c be the upload bandwidth of the streaming server for channel c.

1) Model for NBA: A peer in *NBA* may watch one or multiple channels, and it subscribes to only its watched channels. It allocates its upload bandwidth among its watched channels proportional to their streaming rates. Therefore, a peer m watching channel set θ, allocates its upload bandwidth u_m to channel $c \in \theta$ with the fraction $\frac{r_c}{\sum_{\forall c' \in \theta} r_{c'}}$. That is $x_c^\theta = \frac{r_c}{\sum_{\forall c' \in \theta} r_{c'}}$

For each channel $c \in \Theta$, the total upload bandwidth demand

Digital Object Identifier: 10.1109/INFCOM.2010.5462230

is

$$D_c = \sum_{\forall \theta : c \in \theta} |S_\theta| r_c \tag{1}$$

The total upload bandwidth supply is

$$S_c = \sum_{\forall \theta : c \in \theta} x_c^\theta (\sum_{\forall m \in S_\theta} u_m) + s_c \tag{2}$$

The bandwidth satisfaction ratio γ_c for channel c is

$$\gamma_c = \frac{D_c}{S_c} \tag{3}$$

.

Definition 1 *Given a system configuration, the multi-channel P2P streaming system is defined as NBA feasible if* $\forall c \in \Theta$, $\gamma_c \geq 1$ holds.

2) Model for PCA: A peer in *PCA* may watch one or multiple channels, and it subscribes to only its watched channels. *PCA* is aware of bandwidth imbalance among different channels. Therefore, it optimally allocates the upload bandwidth of a peer in order to maximize the overall system streaming quality. That is, the goal of *PCA* is to find the optimal x_c^θ for the following optimization problem.

$$\max \sum_{\forall c \in \Theta} \gamma_c \tag{4}$$

subject to

$$\sum_{\forall \theta : c \in \theta} |S_\theta| r_c \leq \sum_{\forall \theta : c \in \theta} x_c^\theta (\sum_{\forall m \in S_\theta} u_m) + s_c, \forall c \in \Theta \tag{5}$$

$$\sum_{\forall c \in \theta} x_c^\theta = 1, \forall \theta \subseteq \Theta \tag{6}$$

$$x_c^\theta \geq 0, \forall c \in \Theta, \theta \subseteq \Theta \tag{7}$$

, where $\gamma_c = \frac{\sum_{\forall \theta : c \in \theta} x_c^\theta (\sum_{\forall m \in S_\theta} u_m) + s_c}{\sum_{\forall \theta : c \in \theta} |S_\theta| r_c}$.

Definition 2 *Given a system configuration, the multi-channel P2P streaming system is defined as PCA feasible if the constraints (5) - (7) are satisfied simultaneously.*

3) Model for ACA: A peer in *ACA* may watch one or multiple channels. In addition to subscribing to the watched channels, a peer may also subscribe to one or multiple other unwatched channels, with the aim of contributing its surplus upload bandwidth to the channels with deficient upload bandwidth.

Note that, in order to forward packets of an unwatched channel, a peer must first download these packets, which in turn consumes the upload bandwidth of that channel. That is, while a peer contributes its bandwidth to an unwatched channel, it at the same time also consumes the bandwidth of the unwatched channel (called *overhead*). Therefore, an efficient *ACA* protocol should minimize its overhead. For example, the View-Upload-Decoupling (VUD) protocol proposed in [9] divides the video stream of a specific channel into multiple substreams (e.g. one substream contains packets with even sequence numbers and the other contains packets with odd sequence numbers), which greatly reduces the overhead due to partial downloading of the video stream. In our paper, we assume that the overhead is zero in order to simplify the analysis. This implies that we consider the best performance of *ACA*.

The goal of *ACA* is to find the optimal x_c^θ and y_c^θ for any c and θ for solving the following optimization problem.

$$\max \sum_{\forall c \in \Theta} \gamma_c \tag{8}$$

subject to

$$\begin{aligned} \sum_{\forall \theta : c \in \theta} |S_\theta| r_c &\leq \sum_{\forall \theta : c \in \theta} x_c^\theta (\sum_{\forall m \in S_\theta} u_m) \\ &+ \sum_{\forall \theta : c \notin \theta} y_c^\theta (\sum_{\forall m \in S_\theta} u_m) \\ &+ s_c, \forall c \in \Theta \end{aligned} \tag{9}$$

$$\sum_{\forall c : c \in \theta} x_c^\theta + \sum_{\forall c : c \notin \theta} y_c^\theta = 1, \forall \theta \subseteq \Theta \tag{10}$$

$$x_c^\theta, y_c^\theta \geq 0, \forall c \in \Theta, \theta \subseteq \Theta \tag{11}$$

, where $\gamma_c =$

$$\frac{\sum_{\forall \theta : c \in \theta} x_c^\theta (\sum_{\forall m \in S_\theta} u_m) + \sum_{\forall \theta : c \notin \theta} y_c^\theta (\sum_{\forall m \in S_\theta} u_m) + s_c}{\sum_{\forall \theta : c \in \theta} |S_\theta| r_c}. \tag{12}$$

Definition 3 *Given a system configuration, the multi-channel P2P streaming system is defined as ACA feasible if the constraints (9) - (11) are satisfied simultaneously.*

In addition to the above three feasibility conditions, we also consider the following general feasibility condition.

Definition 4 *Given a system configuration, the system-wide feasibility for NBA, PCA and ACA is defined such that the following inequality holds*

$$\sum_{\forall c : c \in \Theta} \sum_{\forall \theta : c \in \theta} |S_\theta| r_c \leq \sum_{\forall m \in M} u_m + \sum_{\forall c : c \in \Theta} s_c \tag{13}$$

.

The system-wide feasibility condition is the necessary condition for all channels to stream the video at the source rate. Otherwise, none of the *NBA* feasibility condition, *PCA* feasibility condition or *ACA* feasibility condition can be achieved for the system. Note that the group of constraints (9) - (11) is equivalent to constraint (13), and thus we have the following theorem.

Theorem 1: A system configuration is *ACA* feasible if and only if it is system-wide feasible.

Proof: ($\Rightarrow$): According to Definition 3, if a system configuration is *ACA* feasible, then constraint (9) holds, which implies that for each channel c, the total bandwidth demand of that channel is less than or equal to the total bandwidth supply of channel c. Then, we do summation over all channels. Therefore, the left-hand side of (9) is $\sum_{\forall c : c \in \Theta} \sum_{\forall \theta : c \in \theta} |S_\theta| r_c$ and the right-hand side is $\sum_{\forall c : c \in \Theta} (\sum_{\forall \theta : c \in \theta} x_c^\theta (\sum_{\forall m \in S_\theta} u_m) + \sum_{\forall \theta : c \notin \theta} y_c^\theta (\sum_{\forall m \in S_\theta} u_m)) + \sum_{\forall c : c \in \Theta} s_c$. By equation (10), $\sum_{\forall c : c \in \Theta} (\sum_{\forall \theta : c \in \theta} x_c^\theta (\sum_{\forall m \in S_\theta} u_m) + \sum_{\forall \theta : c \notin \theta} y_c^\theta (\sum_{\forall m \in S_\theta} u_m)) = \sum_{\forall m \in M} u_m$, which implies that inequality (13) holds.

Digital Object Identifier: 10.1109/INFCOM.2010.5462230

($\Leftarrow$): $\forall m \in M$ watching a channel set θ, the peer can allocate x_c^θ of its bandwidth to a channel c, $\forall c : c \in \theta$ and it can also allocate y_c^θ of its bandwidth to a channel c, $\forall c : c \notin \theta$. Furthermore, the relationship of x_c^θ and y_c^θ satisfies $\sum_{\forall c:c\in\theta} x_c^\theta + \sum_{\forall c:c\notin\theta} y_c^\theta = 1$. Therefore, the first term of the right-hand side of inequality (13) is

$$\sum_{\forall m\in M} u_m(\sum_{\forall c:c\in\theta} x_c^\theta + \sum_{\forall c:c\notin\theta} y_c^\theta). \tag{14}$$

Rearranging term (14) based on each channel c, we get the new form of the first term of (13)

$$\sum_{\forall c:c\in\Theta} (\sum_{\forall\theta:c\in\theta} x_c^\theta(\sum_{\forall m\in S_\theta} u_m) + \sum_{\forall\theta:c\notin\theta} y_c^\theta(\sum_{\forall m\in S_\theta} u_m)) \tag{15}$$

Based on inequality (13) and (15), constraint (9) is satisfied by all channels in the system. Therefore, we can conclude that the system-wide feasible condition guarantees the *ACA* feasible. ∎

We use the objective functions of maximizing the aggregated bandwidth satisfaction ratios to establish simple linear programming (LP) models for comparing the three designs. These objective functions might not guarantee fair bandwidth allocation among different channels. However, the optimality of our LP models guarantees that there is at least one feasible solution for a specific design and fair allocations can be achieved by other non-linear objective functions. Therefore, our LP formulation serves well for establishing tractable models and comparing feasibilities of the three designs. According to numerical simulations and closed-form solutions, we have the following conclusions: 1) *PCA* design can achieve similar performance as *ACA* design for general channel structures; 2) for special applications (e.g., all users watch the same popular channel on a large window and monitor another channel on a smaller window), *ACA* should be used; and 3) *NBA* can rarely achieve desired performance. Due to space limitations, we do not elaborate on the simulations and analysis here.

IV. Conclusions

Our proposed models are powerful for studying the intrinsic features of three designs. First, they are independent of specific objective functions corresponding to specific implementations. For a given design, the optimality of the model implies that there exists an implementation to provide satisfactory performances for all channels. Second, the models include channel structures (i.e., channel sets and the fractions of peers subscribed to them), which provide a way of numerically exploring the design space of multi-channel P2P streaming systems via changing channel structures. The channel structures also determine the bandwidth imbalance among different channels. Last but not least, linear models can be efficiently solved and thus large-scale evaluations are feasible. Moreover, we can use the linear property of these models to derive closed-form solutions for special systems (e.g., we can use Farkas Lemma [23] and its variants to develop closed-form solutions for a P2P streaming systems with 2 channels). With these models and large-scale numerical simulations, the following two fundamental questions for designing multi-channel P2P streaming systems can be answered: 1) What are the general characteristics of existing and potential designs? and 2) Under what circumstances, should a particular design be used to achieve the desired streaming quality with the lowest implementation complexity?

References

[1] PPLive, http://www.pplive.com.
[2] UUSee, http://www.uusee.com.
[3] Joost, http://www.joost.com.
[4] M. Zhang, Q. Zhang, and S. Yang, "Understanding the power of pull-based streaming protocol: Can we do better?" *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1678–1694, 2007.
[5] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 919–927.
[6] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1672–1687, December 2007.
[7] X. Hei, Y. Liu, and K. Ross, "Inferring network-wide quality in P2P Live streaming systems," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1640–1654, December 2007.
[8] PPStream, http://www.ppstream.com.
[9] D. Wu, C. Liang, Y. Liu, and K. W. Ross, "View-upload decoupling: A redesign of multi-channel P2P video systems," in *Proceedings of IEEE INFOCOM, Mini-Conference*, Rio de Janeiro, Brazil, 2009, pp. 1–6.
[10] M. Wang, L. Xu, and B. Ramamurthy, "A flexible divide-and-conquer protocol for multi-view peer-to-peer live streaming," in *Proceedings of IEEE P2P*, 2009, pp. 291–300.
[11] C. Wu and B. Li, "Strategies of conflict in coexisting streaming overlays," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 481–489.
[12] D. A. Tran, K. A. Hua, and T. Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming," in *Proceedings of IEEE INFOCOM*, San Francisco, CA, March 2003.
[13] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth multicast in a cooperative environment," in *Proceedings of ACM SOSP*, Lake Bolton, NY, October 2003.
[14] X. Zhang, J. Liu, B. Li, and T. Yum, "DONet/CoolStreaming: A data-driven overlay network for live media streaming," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.
[15] C. Wu, B. Li, and S. Zhao, "Multi-channel live P2P streaming: Refocusing on servers," in *Proceedings of IEEE INFOCOM*, Phoenix, AZ, April 2008.
[16] D. Wu, Y. Liu, and K. W. Ross, "Queuing network models for multi-channel P2P live streaming systems," in *Proceedings of IEEE INFOCOM*, Rio de Janeiro, Brazil, 2009, pp. 73–81.
[17] M. Wang, L. Xu, and B. Ramamurthy, "Providing statistically guaranteed streaming quality for peer-to-peer live streaming," in *Proceedings of NOSSDAV*. New York, NY, USA: ACM, 2009, pp. 127–132.
[18] M. Zhang, Y. Xiong, Q. Zhang, and S. Yang, "Optimizing the throughput of data-driven peer-to-peer streaming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 1, 2009.
[19] L. Massoulié, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized decentralized broadcasting algorithms," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, December 2007, pp. 1073–1081.
[20] N. Magharei and R. Rejaie, "PRIME: Peer-to-peer receiver-driven mesh-based streaming," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.
[21] B. Li, Y. Qu, Y. Keung, S. Xie, C. Lin, J. Liu, and X. Zhang, "Inside the new Coolstreaming: principles, measurements and performance implications," in *Proceedings of IEEE INFOCOM*, Phoenix, AZ, April 2008.
[22] F. Wang, J. Liu, and Y. Xiong, "Stable peers: Existence, importance, and application in peer-to-peer live video streaming," in *Proceedings of IEEE INFOCOM*, Phoenix, AZ, April 2008, pp. 1364–1372.
[23] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific, February 1997.

Digital Object Identifier: 10.1109/INFCOM.2010.5462230