

2013

Exploring the Design Space of Multichannel Peer-to-Peer Live Video 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/csearticles>

Wang, Miao; Xu, Lisong; and Ramamurthy, Byrav, "Exploring the Design Space of Multichannel Peer-to-Peer Live Video Streaming Systems" (2013). *CSE Journal Articles*. 119.

<http://digitalcommons.unl.edu/csearticles/119>

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 Journal Articles by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Exploring the Design Space of Multichannel Peer-to-Peer Live Video Streaming Systems

Miao Wang, *Member, IEEE*, Lisong Xu, *Member, IEEE*, and Byrav Ramamurthy, *Member, IEEE*

Abstract—Most of the commercial peer-to-peer (P2P) video streaming deployments support hundreds of channels and are referred to as multichannel systems. Recent research studies have proposed specific protocols to improve the streaming quality for all channels by enabling cross-channel cooperation among multiple channels. In this paper, we focus on the following fundamental problems in designing cooperating multichannel systems: 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? To answer the first question, we propose simple models based on linear programming and network-flow graphs for three general designs, namely Naive Bandwidth allocation Approach (NBA), Passive Channel-aware bandwidth allocation Approach (PCA), and Active Channel-aware bandwidth allocation Approach (ACA), which provide insight into understanding the key characteristics of cross-channel resource sharing. For the second question, we first develop closed-form results for two-channel systems. Then, we use extensive numerical simulations to compare the three designs for various peer population distributions, upload bandwidth distributions, and channel structures. Our analytical and simulation results show that: 1) the NBA design can rarely achieve the desired streaming quality in general cases; 2) the PCA design can achieve the same performance as the ACA design in general cases; and 3) the ACA design should be used for special applications.

Index Terms—Design comparison, multichannel, peer-to-peer (P2P) video streaming.

I. INTRODUCTION

PEER-TO-PEER (P2P) video streaming systems, including both live streaming and video-on-demand (VOD) applications, have been hugely successful in providing multimedia streaming services with hundreds of channels (e.g., UUsee claims to provide about 10 000 channels [1]). Other similar large-scale industry deployments including PPStream [2], CoolStreaming [3], and PPLive [4] support hundreds of channels with tens of thousands of concurrent users¹ [5]. All of these

systems are referred to as multichannel P2P video streaming systems.

Current measurement studies [5], [6] show that the distribution of resources such as the upload bandwidth is unbalanced among different channels, which implies that some channels have satisfactory streaming qualities with surplus resources, while others suffer poor streaming qualities due to resource deficit. Letting the channels with surplus bandwidth help those with deficit bandwidth is the common intuition behind several potential designs since the upload bandwidth is the most precious resource that greatly influences the streaming qualities of all channels [7], [8]. In this paper, cross-channel cooperation means sharing upload bandwidth among different channels.

In a multichannel system with cross-channel cooperation, a user may subscribe² to a variable number of channels and simultaneously watch either all or some of the subscribed channels. It is realistic for a user to simultaneously watch multiple channels since commercial P2P streaming systems allow their users to watch programs in customized manners. For example, PPStream [2] provides picture-in-picture (PIP) functionality, which allows a user to watch two channels at a time. In addition, a user can simultaneously watch one channel on the main screen and monitor multiple channels on the bottom subscreens with Microsoft Silverlight Rich Media Player [10]. Note that, a user may not watch all of its subscribed channels. For example, Wu *et al.* [9] propose a view–upload–decoupling (VUD) approach for building multichannel P2P streaming systems that requires a user to subscribe to other channels as a helper to alleviate the impact of channel switching, even though the user does not watch these subscribed channels.

There are three potential designs for multichannel systems that allow users to watch/subscribe to a variable number of channels.

- Naive Bandwidth allocation Approach (*NBA*), where a user subscribes to only its watched channels and allocates its upload bandwidth to its watched channels proportional to the channel streaming rates (e.g., if all channels have the same streaming rate, the user allocates its bandwidth equally to all watched channels). Most of the current multichannel systems use *NBA* due to its simplicity.
- Passive Channel-aware bandwidth allocation Approach (*PCA*), where a user subscribes to only its watched channels and optimally allocates its bandwidth to its watched channels. The bandwidth allocation algorithm [11] for overlapped overlays and the protocol proposed in [12] are examples of *PCA* design.

Manuscript received August 18, 2010; revised July 15, 2011 and January 10, 2012; accepted March 11, 2012; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. C. S. Lui. Date of publication April 25, 2012; date of current version February 12, 2013.

M. Wang was with the Department of Computer Science and Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588-0115 USA. He is now with A10 Networks, Inc., San Jose, CA 95134 USA (e-mail: mwang@cse.unl.edu).

L. Xu and B. Ramamurthy are with the Department of Computer Science and Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588-0115 USA (e-mail: xu@cse.unl.edu; byrav@cse.unl.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNET.2012.2194165

¹We use the term *user* and *peer* interchangeably in this paper.

²Subscribing to a channel means that a peer participates in video dissemination for that channel, but may not watch that channel. If the peer does not watch that channel, it serves as a helper [9] for that channel.

- Active Channel-aware bandwidth allocation Approach (*ACA*), where a user subscribes not only to its watched channels, but also maybe to some other channels as a helper. A user optimally allocates its bandwidth to the watched channels and the subscribed but unwatched channels. 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. VUD proposed in [9] is a special case of *ACA* design since a peer is restricted to watch only one channel and might be selected by the system to join other channels as a helper.

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

Based on the above definitions, the three designs represent the major design schemes of a multichannel P2P streaming system, which is referred to as the design space in this paper. However, all the existing works focus on proposing and evaluating specific protocols instead of studying the intrinsic features of designing multichannel systems. The goal of this paper is to generalize and analyze the designs of multichannel systems, and thus shed insights into choosing the proper design, in terms of complexity and effectiveness, which actually explores the design space of a multichannel P2P streaming system. Specifically, we answer the following two questions: 1) what are the general characteristics of existing and potential designs? 2) under what circumstances should a specific design be used to achieve the desired streaming quality with the lowest complexity?

The contributions of this paper are as follows.

- 1) We identify three designs—namely *NBA*, *PCA*, and *ACA*—for building multichannel P2P streaming systems and develop simple models based on linear programming and network flow graphs for the three designs, which capture their main characteristics.
- 2) With established models, we further prove that finding optimal *ACA* design with overhead is NP-complete and provide qualitative discussion of relative implementation complexities.
- 3) We derive closed-form results for a two-channel system. Our results show that for this special case there is no need to use *ACA* design, and the *NBA* design can either only provide low-quality streaming or consume higher bandwidth to provide the same level of streaming quality as *PCA*. The channel structure (refer to the last paragraph of Sections IV and V-B) greatly influences the performance.
- 4) We conduct extensive numerical simulations to compare the three designs in general cases. Our results show that for general multichannel P2P streaming systems, *PCA* can achieve the same performance as *ACA*, while for special applications, *ACA* is required.

The rest of this paper is organized as follows. Section II briefly summarizes the related work. Section III describes our simple models based on network-flow graphs and insights

on the three designs. Section IV discusses the homogenous two-channel systems, which are special cases of models proposed in Section III. Section V describes the simulation settings and results. Finally, we conclude the paper in Section VI.

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 [13], [14]) 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 [4] and UUSee [1]. CoolStreaming [3] 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 investigate the case when a peer joins multiple overlays in a P2P live streaming system and propose an auction-based bandwidth allocation algorithm to improve the streaming quality for all channels. In our previous work [12], we propose a flexible protocol for multiview P2P live streaming systems, based on the divide-and-conquer strategy, which solves the interchannel competition and intrachannel streaming separately.

In the area of multichannel systems, there are three closely related papers. In [15], the authors study the problem of provisioning the server bandwidth consumption in multichannel systems. Wu *et al.* [9] propose the view-upload-decoupling approach to minimize the influence of channel churn among multiple channels. Moreover, in [16], the same authors establish queueing network models to analytically study the performance of multichannel systems by considering channel churn, peer churn, bandwidth heterogeneity, etc. Their analytical model differs from ours due to the following reasons: 1) they focus on the multichannel system with the restriction that a peer can watch exactly one channel, which is a special case of the *ACA*; we study more general cases; 2) they analyze the dynamic features for multichannel systems, in which a specific approach is used; we focus on fundamental problems of whether a complex design should be used and which design is better.

In the area of theoretical analysis of P2P streaming, there are some studies on single-channel streaming systems, where there is exactly one channel in the system. Kumar *et al.* [8] study the performance limitations of a single-channel streaming system with a stochastic fluid model. Liu *et al.* [17] derive the performance bound of single-channel systems in terms of server load, streaming rate, and tree depth. Massoulie *et al.* [18] develop a network-flow based model to study the decentralized broadcasting problems and propose an optimal broadcasting algorithm. In [19], Liu *et al.* study the flash crowd problem in P2P live streaming systems. References [20] and [21] focus on the chunk scheduling problem and propose optimal algorithms. They assume that the total upload bandwidth supply of all channels are sufficient to satisfy the total bandwidth demand to guarantee the algorithm convergence, and the algorithm is evaluated with a small number of channels and channel combinations.

In our previous works [22]–[24], we propose a framework for comparing multichannel P2P streaming systems with linear programming models. In this paper, we first extend the framework with detailed discussions and proofs and apply the framework for analyzing the three designs as well. Moreover, we discuss the model of *ACA* design with overhead and use extensive numerical simulations to compare the three designs in general scenarios.

III. LINEAR PROGRAMMING MODELS, NETWORK-FLOW GRAPHS, AND INSIGHTS FOR MULTICHANNEL P2P STREAMING DESIGNS

In Section III-A, we introduce the linear programming models for the three designs with feasibility definitions. Section III-B uses network-flow graphs to provide insights on the three designs, which expose intrinsic characteristics of models defined in Section III-A and provide intuitive explanations of their usages. Then, in Section III-C, we prove that the model for *ACA* design with overhead is NP-complete. Finally, we discuss the implementation complexities of the three designs in Section III-D.

A. Linear Programming Models for the Three Designs

As described above, designs of a multichannel P2P streaming system are different in their approaches for solving the bandwidth allocation problem among all channels in the system, which is referred to as the interchannel bandwidth allocation problem. To compare different designs, we need to establish models, which are able to capture the main characteristics of approaches for solving the interchannel bandwidth allocation problem and can be efficiently solved. In the remaining part of this section, we introduce the linear programming models, which can satisfy the above requirements and are used for comparing different designs in numerical simulations.

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 the system is relatively stable during each interval.

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 [5], [25], [26] 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 [26]) 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 multichannel P2P streaming systems. Furthermore, we are interested in comparing the three designs to

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 channel's total bandwidth demand (a formal definition will be introduced below). Note that the primary goal of this paper is to compare the feasible solution regions of *NBA*, *PCA*, and *ACA* designs, which are independent of corresponding optimization objectives of the three designs and only determined by constraints of bandwidth allocation models defined in the following sections. The reason for choosing the maximization of the aggregated bandwidth satisfaction ratio of all channels as the bandwidth allocation objective is mainly due to its linear property. The linear property makes large-scale numerical simulations possible.

We introduce the common notation used in this section and Section III-B as follows.

- 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, $S_\theta = \{m | \theta(m) = \theta\}$. $\theta(m)$ denotes the channel set watched by peer m .
- Let M be the total number of peers in the system.
- 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 .
- Let U_θ be the total upload bandwidth supply of user set S_θ .
- Let D_c be the total upload bandwidth demand of channel c .
- Let N be the total number of peers.
- Let P_θ be the fraction of peers watching channel set θ .

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 $r_c / \sum_{c' \in \theta} r_{c'}$. That is $x_c^\theta = r_c \times (\sum_{c' \in \theta} r_{c'})^{-1}$.

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

$$D_c = \sum_{\forall \theta: c \in \theta} |S_\theta| r_c. \quad (1)$$

The total upload bandwidth supply is

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

The bandwidth satisfaction ratio γ_c for channel c is

$$\gamma_c = \frac{S_c}{D_c}. \quad (3)$$

Definition 1: Given a system configuration, the multichannel 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:

$$\max \sum_{c \in \Theta} \gamma_c \quad (4)$$

subject to

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

$$\sum_{c \in \theta} x_c^\theta = 1 \quad \forall \theta \subseteq \Theta \quad (6)$$

$$x_c^\theta \geq 0 \quad \forall c \in \Theta, \theta \subseteq \Theta \quad (7)$$

where $\gamma_c = \frac{(\sum_{\theta: c \in \theta} x_c^\theta (\sum_{m \in S_\theta} u_m) + s_c)}{(\sum_{\theta: c \in \theta} |S_\theta| r_c)}^{-1}$.

Definition 2: Given a system configuration, the multichannel 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 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 this section, we assume that the overhead is zero in order to simplify the analysis. This implies that we consider the best performance of *ACA*. We will discuss the *ACA* with overhead in Section III-C.

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:

$$\max \sum_{c \in \Theta} \gamma_c \quad (8)$$

subject to

$$\sum_{\theta: c \in \theta} |S_\theta| r_c \leq \sum_{\theta: c \in \theta} x_c^\theta \left(\sum_{m \in S_\theta} u_m \right) + \sum_{\theta: c \notin \theta} y_c^\theta \left(\sum_{m \in S_\theta} u_m \right) + s_c \quad \forall c \in \Theta \quad (9)$$

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

$$x_c^\theta, y_c^\theta \geq 0 \quad \forall c \in \Theta, \theta \subseteq \Theta \quad (11)$$

where

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

Definition 3: Given a system configuration, the multichannel 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:

$$\sum_{c: c \in \Theta} \sum_{\theta: c \in \theta} |S_\theta| r_c \leq \sum_{m \in M} u_m + \sum_{c: c \in \Theta} s_c. \quad (13)$$

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: Please find a detailed proof in Appendix A. ■

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 nonlinear objective functions. Therefore, our LP formulation serves well for establishing tractable models and comparing feasibilities of the three designs.

B. Network-Flow Graphs for the Three Designs

In Section III-A, we establish linear programming models for *NBA*, *PCA*, and *ACA* designs, which are powerful for theoretical analysis and numerical simulations. In order to understand the intrinsic characteristics of the three designs better (i.e., differences and relationships), we interpret the three designs along with their corresponding models with generalized network-flow graphs. With network graphs, we are able to obtain

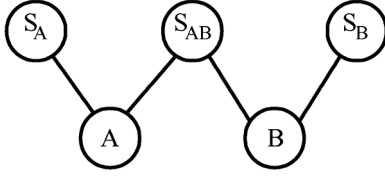


Fig. 1. Resource allocation graph for a multichannel P2P streaming system with two channels A and B .

a clear intuition of cross-channel bandwidth sharing among different channels and the power of *ACA* design. Note that we can also apply network-flow-based algorithms in numerical simulations to solve the bandwidth allocation problems defined by linear programming models above. We first describe the way of constructing network-flow graphs from linear programming models below.

We first consider the resource allocation graph and network-flow graph for *PCA*. For each user set S_θ , *PCA* considers how to allocate the total upload bandwidth of all users watching just channel set θ to all channels $c \in \theta$. Intuitively, a user set S_θ provides its upload bandwidth to a set of channels, and a channel c requests upload bandwidth from some user sets. Therefore, we call a user set a *bandwidth supplier* and a channel a *bandwidth consumer*. The relationship between suppliers and consumers can be described by a bipartite resource allocation graph $G = (S, D, E)$, where vertex set S is the set of all suppliers (i.e., S contains S_θ for any $\theta \subseteq \Theta$), vertex set D is the set of all consumers (i.e., D contains c for any $c \in \Theta$), and edge set E represents the supplier-consumer relationship (i.e., $e = (S_\theta, c) \in E$ iff $c \in \theta$). Fig. 1 illustrates the bipartite graph with three suppliers and two consumers for a multichannel P2P streaming system with two channels: A and B . For example, supplier S_{AB} allocates its upload bandwidth to consumers A and B .

Based on the resource allocation graphs, we can construct the network-flow graph for *PCA* to visualize the model for *PCA*. We introduce two artificial vertices s and t to denote the source and sink of the network-flow graph, respectively. We add edges to connect the source s with all consumer vertices in D , whose capacities are the bandwidth demands of the correspondingly connected consumer vertices. The capacities of edges in the original resource allocation graph are set to $+\infty$. Then, we divide each supplier vertex into vertices S_θ and S'_θ connected by a single edge, whose capacity is determined as follows. 1) If $|\theta| = 1$, which implies that the users in S_θ watch a single channel, then the capacity $U_\theta = s_c + \sum_{m \in \theta} u_m$ for $c \in \theta$. 2) If $|\theta| > 1$, then the capacity $U_\theta = \sum_{m \in \theta} u_m$. Finally, we connect the vertices S'_θ to the sink t , with $+\infty$ edge capacities. Fig. 2 shows the network-flow graph with three suppliers and two consumers.

The network-flow graph for *ACA* differs from the graph of *PCA* since a user m who does not watch a channel, say c , is still able to help channel c . Therefore, we add new virtual edges to resource allocation graph to construct the network-flow graph for *ACA* as follows. For any pair of a bandwidth supplier vertex S_θ and a bandwidth consumer vertex c , $c \in D$, there is a virtual edge connecting them. Then, we apply the same rules of constructing network-flow graphs for *PCA*. An example is shown in Fig. 3.

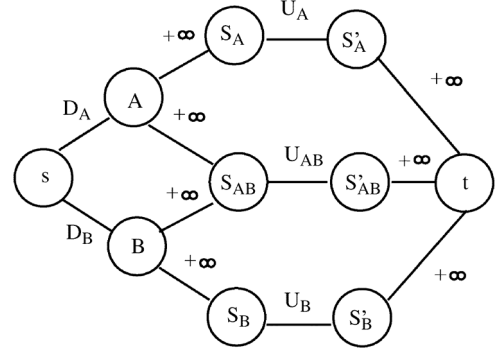


Fig. 2. Network-flow graph for two-channel *PCA* model. The notations on edges denote the edge capacities.

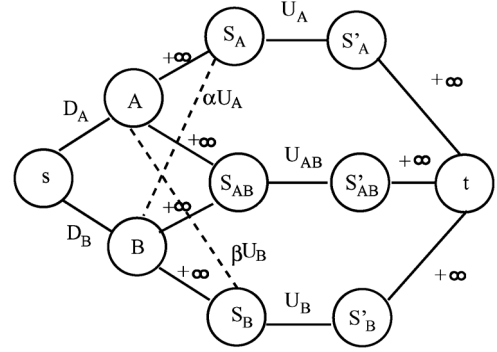


Fig. 3. Network-flow graph for two-channel *ACA* model. The notations on edges denote the edge capacities. The dashed lines denote virtual edges.

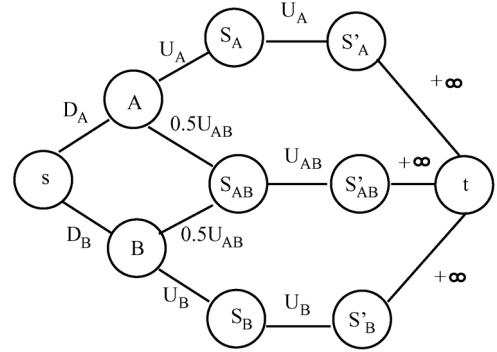


Fig. 4. Network-flow graph for two-channel *NBA* model. The notations on edges denote the edge capacities. We assume that $r_A = r_B$.

Finally, the network-flow graph for *NBA* differs from that of *PCA* in the capacities of edges between the supplier and consumer vertices, which are set based on the bandwidth allocation strategy of *NBA*, instead of $+\infty$. Fig. 4 shows the network-flow graph for the *NBA* design with three suppliers and two consumers, where the two channels have the same streaming rate.

With the network-flow graphs of *NBA*, *PCA*, and *ACA*, we can interpret the task of determining whether a specific design has a feasible solution in the sense that there exists a bandwidth allocation strategy using which all channels' bandwidth demands are satisfied. As shown in Figs. 2–4, the capacities of edges connecting the source vertex s and the consumers denote the bandwidth demands.³ Therefore, if all these edges are saturated

³In the network-flow graphs, we let the capacity of the edge connecting the source vertex s and a consumer be the exact bandwidth demand, which is equivalent to the case where the equality holds in constraints (5) and (9).

in the maximum flow of the graphs corresponding to *NBA*, *PCA*, and *ACA* designs (the maximum flow is equal to the total bandwidth demand in the system), we say that the system is *NBA feasible*, *PCA feasible*, and *ACA feasible*, respectively (please refer to Definitions 1–3). Obviously, the system-wide feasibility is a necessary condition for feasibilities of all three designs. Furthermore, from Fig. 3, we can see that there is no edge capacity limit for the edge connecting a bandwidth consumer, and a bandwidth supplier and a bandwidth consumer can request bandwidth from any bandwidth supplier with the *ACA* design. Therefore, the system-wide feasible condition is also the sufficient condition for *ACA* feasibility, which is stated in Theorem 1. Note that the reason why system-wide feasibility implies *ACA* feasibility is that in our current *ACA* network-flow graph, we do not consider the overhead caused by *ACA*, which will be discussed in Section III-C.

C. ACA Model With Overhead

As mentioned in Section III-B, the *ACA* model in this paper does not consider the overhead caused by cross-channel cooperation. In real P2P streaming systems, any *ACA* design will introduce a certain amount of overhead since when a peer with surplus bandwidth intends to help peers watching its unwatched channels, it first needs to download the useful data (i.e., it increases the bandwidth demand of that channel). We use *ACA-O* to denote *ACA* with overhead. In this section, we show that omitting the overhead is not unrealistic for real P2P streaming systems and does not change the relative order of the three designs in terms of performance. In addition, we also prove that omitting the overhead greatly simplifies the analysis of *ACA* design and makes large-scale numerical simulations possible in that the model for *ACA-O* is NP-complete.

In fact, current P2P streaming deployments (e.g., [2]) divide video streams into multiple substreams with lower streaming rates compared to the original streaming rate. A peer with surplus bandwidth can subscribe to an unwatched channel as a helper (bandwidth amplifier) [9] as long as the extra bandwidth is higher than the substream rate (i.e., the overhead caused by subscribing to an unwatched channel is lower than the bandwidth increase in that channel). Intuitively, when omitting the overhead introduced by *ACA* design, assuming that *ACA* has a feasible solution is equivalent to assuming that the total bandwidth supply of all channels is higher than the total bandwidth demand. Note that the power of *ACA* design is to balance bandwidth among all channels in a P2P streaming system and the overhead is bounded (i.e., limited by streaming rate of substreams). To efficiently utilize peers' bandwidth, we should divide the original video stream into substreams as many as possible. The overhead caused by *ACA* design asymptotically approaches to zero if the number of substreams approaches infinity. However, there is a tradeoff between the number of substreams and the substream scheduling complexity. Wu *et al.* [16] establish queueing models for analyzing the relationship between streaming quality and methods of dividing substreams. Finding the optimal number of substreams is beyond the scope of this paper.

Determining the feasibility of *ACA-O* could be NP-hard. Intuitively, a consumer determines from which supplier to

request bandwidth and how much bandwidth should be requested from that supplier, where the number of choices of both decisions is exponentially large. Although searching over exponentially large space does not necessarily imply that the problem is NP-hard, we prove below that even a simplified *ACA-O* is NP-hard. Since with the *ACA-O* model, the peer with surplus bandwidth determines which peer should be helped and how much bandwidth should be allocated to that user while considering the overhead caused by such a cooperation. The simplified *ACA-O* in this paper refers to the case that the peer with surplus bandwidth has the information about the bandwidth that can be allocated to different peers and the overhead (cost) for helping these peers. Therefore, the simplified *ACA-O* only needs to find out a set of peers that the peer with surplus bandwidth could help. A decision version of the simplified *ACA-O* is given below.

To illustrate the simplified *ACA-O*, we use Fig. 3 as an example. Since *ACA* design causes overhead, it changes the edge capacity of edges between the source s and consumers. For example, in Fig. 3, consumer A requests bandwidth from supplier S_B , which increases the edge capacity of edge (S, A) . Therefore, for simplified *ACA-O*, we assume that for each consumer the amount of bandwidth requested from each supplier is set by some scheme, which is reflected by the edge capacity of edges between consumers and suppliers. For example, in Fig. 3, the edge capacity of (A, S_B) is some fixed number instead of $+\infty$. Then, determining the feasibility of the simplified *ACA-O* is to find out a feasible network-flow in the flow graph with a set of saturated edges between consumers and suppliers, whose starting vertices cover all consumers. The decision version of this problem is shown below.

Simplified ACA-O Decision: Given a network-flow graph of the simplified *ACA-O*, $G = (\{S, D, D', s, t\}, E)$, where S and D denote the consumers and suppliers, respectively; D' denotes the mirror vertices of suppliers (e.g., S'_B in Fig. 3); s and t denote the source and sink vertices; $\{S, D, D', s, t\}$ denotes the vertex set of the graph; E denotes the set of edges with capacities.

Question: Is there a network-flow containing a set of saturated edges between S and D , whose starting vertices cover the vertex set S ?

Theorem 2: Simplified *ACA-O* Decision is NP-complete.

Proof: Please find a detailed proof in Appendix B. ■

To sum up, the *ACA* model without overhead used in studying the relative performance of the three designs is reasonable and will provide meaningful comparison results for the three designs.

D. Discussions of Implementation Complexity

In this section, we briefly discuss what are the main factors of implementation complexities for designing multichannel P2P streaming systems with cross-channel bandwidth sharing and how they influence the implementation complexities of the three designs.

The *NBA* design is not aware of the bandwidth imbalance of different channels and therefore does not require any extra system bandwidth information, and there is no optimal bandwidth allocation algorithm either. By contrast, the *ACA* not only

requires extra system bandwidth information for optimal bandwidth allocation, but also it needs some algorithm to find a proper/optimal way for peers to determine whether should subscribe to unwatched channels and which channels should be subscribed, which results in the most complex design among the three. The implementation complexity of *PCA* design falls between *NBA* and *PCA* since it requires extra system bandwidth information for optimal bandwidth allocation, which is similar to *ACA*, but does not need to determine whether and how peers subscribe to unwatched channels. Note that the qualitative comparisons are based on the assumption that the three designs are implemented with reasonable mechanisms. A poorly designed implementation of *PCA* is probably more complex than a well-designed implementation of *ACA*.

IV. TWO-CHANNEL P2P STREAMING SYSTEMS

In this section, we compare the three designs in a P2P streaming system with two channels using the closed-form feasibility discriminant.

A. Closed-Form Discriminant for Homogenous Two-Channel System With *PCA* Design

The simplest multichannel system is the two-channel system. To obtain a closed-form discriminant, we first assume that all peers have the same upload bandwidth and the streaming rates of the two channels are the same as well. This simplified system is referred to as homogenous two-channel system (HOMO-2 for short). In addition, to state the problem compactly and show the procedure of obtaining the results clearly, we represent the problem in the linear programming format.

Before presenting the results of HOMO-2, we first introduce a theorem that will be used for establishing the closed-form results for HOMO-2 and is a variant of Farkas's lemma [27]. To state the theorem compactly, we use the matrix notation to prove the theorem. Since constraints for *NBA*, *PCA*, and *ACA* models can be rearranged into the form $ax \leq b$, the matrix notation of the constraints can be written in the following format:

$$\begin{aligned} \mathbf{A}\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned} \quad (14)$$

where \mathbf{A} denotes the coefficient matrix for the rearranged constraints and \mathbf{b} denotes the right-hand-side values of the constraints.

Theorem 3: Given a matrix \mathbf{A} of dimensions $m \times n$ and a vector $\mathbf{b} \in \mathbf{R}^m$, the feasibility set⁴ determined by the system of inequalities (14) is either nonempty or $\exists \mathbf{p} \in \mathbf{R}^m$ satisfies $\mathbf{p} \geq \mathbf{0}$, $\mathbf{p}^T \mathbf{b} < \mathbf{0}$, $\mathbf{p}^T \mathbf{A} \geq \mathbf{0}^T$, but not both.

For the simplified system, there are two channels, Channels 1 and 2⁵ with streaming rate r and three channel sets $\theta_1 = \{1\}$, $\theta_2 = \{2\}$, $\theta_3 = \{1, 2\}$. All peers have upload bandwidth u . Other notations are the same with Section III. The following theorem shows the discriminant for *PCA* design.

⁴Informally, the feasibility set is a set of solutions that satisfy all constraints of a linear programming model.

⁵We use 1 and 2 to represent channels instead of A and B in this and the next sections.

Theorem 4: For homogenous two-channel systems with *PCA* design (*HPCA-2*), the feasibility of the bandwidth allocation problem represented by the network-flow graph is determined by $\Delta = (s_1 + s_2/uN) + P_{\theta_1} + P_{\theta_2} + P_{\theta_3} - (P_{\theta_1} + P_{\theta_2} + 2P_{\theta_3})(r/u)$. If $\Delta < 0$, the bandwidth allocation problem is infeasible; otherwise it is feasible.

Proof: Following Theorem 3, we derive the primal and dual problem for *HPCA-2* below (we rearranged the constraints to be in the same format as those in Theorem 3). Note that the goal of Primal is to find a feasible vector $\mathbf{p} = (p_1, p_2)$, where the elements of matrix \mathbf{A} are the right-hand side of constraints (16) and (17) and elements of vector \mathbf{b} are coefficients of the left-hand side of constraints (16) and (17).

Dual:

$$\max \mathbf{0}^T \mathbf{x} \quad (15)$$

subject to

$$-P_{\theta_3}x_1^{\theta_3} \leq \frac{s_1}{uN} + P_{\theta_1} - (P_{\theta_1} + P_{\theta_3})\frac{r}{u} \quad (16)$$

$$\begin{aligned} P_{\theta_3}x_1^{\theta_3} &\leq \frac{s_2}{uN} + P_{\theta_2} + P_{\theta_3} - (P_{\theta_2} + P_{\theta_3})\frac{r}{u} \quad (17) \\ x_1^{\theta_3} &\geq 0 \end{aligned}$$

Primal:

$$\begin{aligned} \min p_1 &\left(\frac{s_1}{uN} + P_{\theta_1} - (P_{\theta_1} + P_{\theta_3})\frac{r}{u} \right) \\ &+ p_2 \left(\frac{s_2}{u} + (P_{\theta_2} + P_{\theta_3}) \left(1 - \frac{r}{u} \right) \right) \end{aligned} \quad (18)$$

subject to

$$\begin{aligned} -p_1P_{\theta_3} + p_2P_{\theta_3} &\geq 0 \\ p_1, p_2 &\geq 0. \end{aligned} \quad (19)$$

Based on $P_{\theta_3} \neq 0$ and the constraint (19), we conclude that $p_2 \geq p_1$. Therefore, the objective function (18) has an upper bound $p_2 \times \Delta$ and a lower bound $p_1 \times \Delta$. Because $p_1, p_2 \geq 0$, the sign of Δ determines the sign of the objective function (18). If $\Delta < 0$, the Primal problem is unbounded since its upper bound is negative and any pair of $p_1, p_2 > 0$ is a certificate of infeasibility for *HPCA-2* based on Theorem 3. If $\Delta \geq 0$, we cannot find a pair $p_1, p_2 \geq 0$ that makes the primal objective function negative since its lower bound is always nonnegative. ■

When $P_{\theta_3} = 0$, it becomes a two-channel system with two isolated channels, which is a special case of *NBA* design with no cross-channel resource sharing. $\Delta = (s_1 + s_2/uN) + P_{\theta_1} + P_{\theta_2} + 0 - (P_{\theta_1} + P_{\theta_2} + 0)(r/u) = (s_1 + s_2/uN) + P_{\theta_1} + P_{\theta_2} - (P_{\theta_1} + P_{\theta_2})(r/u)$. $\Delta \geq 0$ indicates that $s_1 + s_2 + uN(P_{\theta_1} + P_{\theta_2}) \geq rN(P_{\theta_1} + P_{\theta_2})$, which is exactly the same as the condition for system-wide feasibility. For this system, only if $u > r$ can it achieve the required streaming rate.

Corollary 1: For *HPCA-2*, when $N \rightarrow \infty$ (i.e., the system size approaches infinity), there exists a critical point P^* for the fraction of peers watching both channels, where $P^* = u - r/r$. If $P_{\theta_3} \leq P^*$ bandwidth allocation problem is feasible, otherwise it is infeasible.

Proof: For *HPCA-2*, the equation

$$P_{\theta_1} + P_{\theta_2} + P_{\theta_3} = 1 \quad (20)$$

holds. Substitute $P_{\theta_1} + P_{\theta_2} + P_{\theta_3}$ in Δ of Theorem 4 with (20). Thus, $\Delta = (s_1 + s_2/uN) + 1 - (1 + P_{\theta_3})(r/u)$. Then

$$\lim_{N \rightarrow \infty} \Delta = 1 - (1 + P_{\theta_3}) \frac{r}{u}. \quad (21)$$

Based on Theorem 4 and (21), if $P_{\theta_3} > (u-r/r)$, the bandwidth allocation problem is infeasible, which implies that the critical point $P^* = (u-r/r)$. ■

The intuition behind Corollary 1 is that in *HPCA-2*, multi-channel peers consume more bandwidth than single-channel peers, but their upload bandwidths are the same as single-channel peers⁷. Therefore, they are the cause for bandwidth deficit, and the fraction of multichannel peers that can be supported by the system is bounded. Particularly, Corollary 1 provides insights that the ratio of peer's upload bandwidth over streaming rate is a key parameter for *HPCA-2* to determine whether the system is *PCA* feasible. Therefore, in the following general cases, we should investigate the impact of peers' upload bandwidth and streaming rates for different channels on the three designs.

B. Closed-Form Discriminant for Homogenous Two-Channel System With ACA Design

We can use a similar method to obtain the closed-form discriminant for Homogeneous 2 channels with ACA design (*HACA-2*). However, we use Theorem 1 to obtain the discriminant directly.

Theorem 5: The *HACA-2* has the same discriminant as the *HPCA-2*.

Proof: Based on the system-wide feasible condition, we obtain the following inequality *HOMO-2*:

$$Nr(P_{\theta_1} + P_{\theta_2} + 2P_{\theta_3}) \leq Nu(P_{\theta_1} + P_{\theta_2} + P_{\theta_3}) + s_1 + s_2. \quad (22)$$

Let $N \rightarrow \infty$, we can derive that if $P_{\theta_3} \leq (u-r/r)$, *HOMO-2* is system-wide feasible. Therefore, the condition for system-wide feasibility of *HOMO-2* is the same as the condition for *PCA* feasibility of *HPCA-2*. Based on Theorem 1, we have proven Theorem 5. ■

C. Closed-Form Discriminant for Homogenous Two-Channel System With NBA Design

Then, we compare the *NBA* and *PCA* design in *HOMO-2* (referred to as *HNBA-2*) to determine which design should be used.

For *NBA* design, we substitute $x_1^{\theta_3}$ and $x_2^{\theta_3}$ with $1/2$ to the constraints of *HNBA-2* and get the following conclusion. When $0 < (r/u) < (1/2)$, *HNBA-2* is always *NBA* feasible. When $(1/2) < (r/u) < 1$, if $P_{\theta_3} \leq \min(P_{\theta_1}, P_{\theta_2})(2(u-r)/2r-u)$, *HNBA-2* is *NBA* feasible.

Let P_{\min} denote $\min(P_{\theta_1}, P_{\theta_2})$. We can determine whether to use *NBA* or *PCA* design by comparing $P_{\min}(2(u-r)/2r-u)$ and $u-r/r$. If $P_{\min}(2(u-r)/2r-u) < (u-r/r)$, when P_{θ_3} in the interval $(P_{\min}(2(u-r)/2r-u), (u-r/r))$, *PCA* design should be used. If $P_{\min}(2(u-r)/2r-u) > (u-r/r)$, when *HOMO-2* is system-wide feasible, we should use *NBA* design. The precise conclusion is summarized as follows.

Conclusion for NBA:

- 1) If $(1/2) < (r/u) < 1$, and $P_{\min} < (2r-u/2r)$, when $P_{\min}(2(u-r)/2r-u) < P_{\theta_3} \leq (u-r/r)$, we should

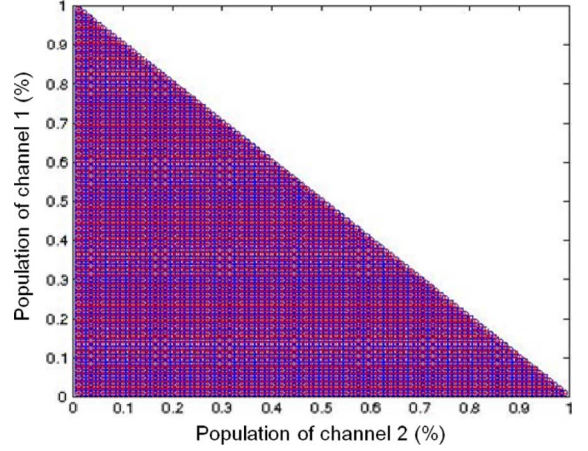


Fig. 5. Feasible regions of the three designs when $(r/u) \leq 0.5$. All the three designs have the same feasible region.

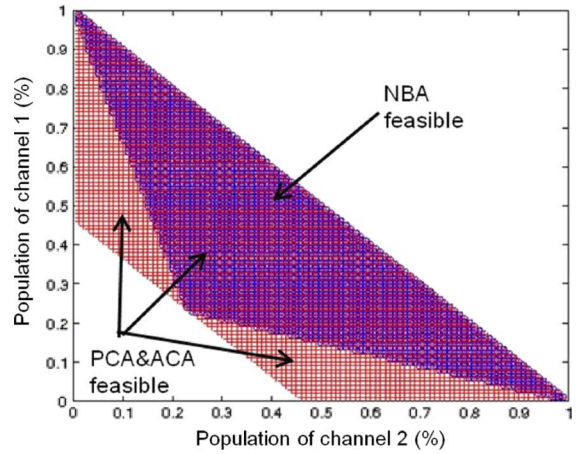


Fig. 6. Feasible regions of the three designs when $r/u = 0.65$. *PCA* and *ACA* have larger feasible region when $(r/u) > 0.65$.

use *PCA* design, and when $P_{\theta_3} \leq P_{\min}(2(u-r)/2r-u)$, we should use *NBA* design.

- 2) If $(1/2) < (r/u) < 1$, and $P_{\min} \geq (2r-u/2r)$, when $P_{\theta_3} \leq (u-r/r)$, we should use *NBA* design.
 - 3) If $0 < (r/u) < (1/2)$, we should always use *NBA* design.
- We will visualize the results for *NBA*, *PCA*, and *ACA* next.

D. Discussions

We visualize the results of *HOMO-2* with Figs. 5–7. In these figures, for a given r/u , the feasible region of the three designs is represented by the area defined by populations of peers watching only channel A (P_{θ_1}) and channel B (P_{θ_2}). From Fig. 5, we can conclude that if $(r/u) \leq 0.5$, *NBA* design will be good enough. Based on Figs. 6 and 7, the feasible region shrinks with the increase of r/u , in that the bandwidth demand in the system is close to the bandwidth supply.

Moreover, the results of *HOMO-2* have two important implications for designing real P2P streaming systems and comparing different designs. First, in a multichannel P2P streaming system with well-balanced resources allocated among different channels (i.e., the bandwidth of different peer sets can be roughly considered the same), the maximum achievable streaming rate is restricted by peers' upload bandwidth, and *NBA* design can only

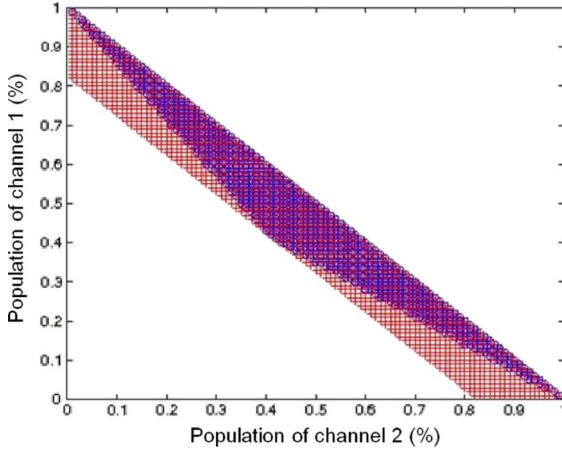


Fig. 7. Feasible regions of the three designs when $r/u = 0.85$. When r/u increases, the feasible regions of all three designs decrease.

support low-quality videos, or it requires more bandwidth than *PCA* to sustain the same streaming rate. As shown in Figs. 5–7, the feasible region shrinks when bandwidth demands approach to bandwidth supplies in the whole system. However, based on measurement studies [5], [6], the resources allocated among different channels are highly unbalanced, which intuitively indicates that *NBA* design is not good enough for sustaining satisfactory quality of service. Therefore, choosing the proper design is urgent for real systems.

Second, the results of *HOMO-2* analysis show that the population and upload bandwidth of different user sets and the streaming rates of different channels greatly influence the feasible regions of different designs. In fact, these factors determine the bandwidth supply and demand relationship in the system, which should be carefully considered in evaluating general cases in Section V. A less obvious factor shown in *HOMO-2* that influences the design space is the channel structure, which can be defined as the channel sets in the system (i.e., how users subscribe to or watch channels in the whole P2P streaming system). For example, in *HOMO-2*, there are three channel sets: channel set $\{1\}$, set $\{2\}$, and set $\{1, 2\}$, which determine the way of cross-channel sharing. If a *HOMO-2* does not have channel set $\{1, 2\}$, channels 1 and 2 cannot share resources. In real systems, there might be a large number of channel sets corresponding to a variety of channel structures. Therefore, when comparing different designs for the general case, the channel-structure factor should be given careful consideration. Furthermore, the intuition behind observations of the two-channel system is the influence of bandwidth demand and supply of the two channels. Since our observations are based on homogenous bandwidth assumptions, we can expect that heterogeneous scenarios can introduce higher degrees of bandwidth demand and supply heterogeneity, which have higher impact on the feasible regions of different designs. However, it is difficult to obtain closed-form solutions for heterogeneous scenarios, which can be achieved with our numerical simulations and is a reason of the requirement of efficient computation models (i.e., our linear programming models).

V. NUMERICAL RESULTS

In this section, we investigate the characteristics of the three designs with extensive numerical simulations.

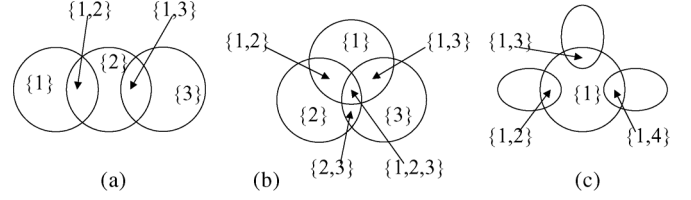


Fig. 8. Three types of channel structures: (a) chain, (b) mesh, and (c) star.

A. Experiments Setup

We develop a configurable simulator using C++ and integrate it with the *CPLEX*⁶ [28] optimization library to solve the bandwidth allocation problems of the three models. Therefore, we can compare the three designs for various peer population distributions, upload bandwidth distributions, and channel structures by changing the following parameters.

As shown in the two-channel case, the channel streaming rate, the peer upload bandwidth, and the peer population greatly influence the feasible region of the three designs. Therefore, the design space can be determined by the channel structure, the peer population distribution, and the peer bandwidth distribution. To explore the design space defined above, we study each design for multichannel systems with the following two groups of parameters: 1) channel information parameters; 2) system information parameters.

The channel information parameters include the streaming rate r_c for channel c and the channel structure. We consider two types of channel streaming rates:

- homogeneous streaming rate, where all channels have the same streaming rate;
- heterogeneous streaming rates, where different channels have different streaming rates, corresponding to different video qualities.

The channel structure determines whether two or more channels overlap with each other. We use the following three types of channel structures to investigate the three designs, as illustrated in Fig. 8:

- a chain structure where a user can view only the feeds from either a single camera or two consecutive cameras in a row of cameras;
- a mesh structure where every user watches a random number of channels;
- a star structure where there is one popular channel that every user watches.

The above three channel structures cover a variety of general and special cases in a real system. For example, the chain structure might be an application of camera monitoring systems used in traffic, zoos, etc. [12], while the star structure might be a P2P streaming system with PIP function [2]. For a given channel structure, the number of peers in each channel set is determined by the system information parameters below.

The system information parameters include the number of channels, the number of peers, the maximum number of simultaneously subscribed/watched channels, the bandwidth distribution of channel sets, and the population distribution of the

⁶IBM ILOG CPLEX optimizer is a high-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming.

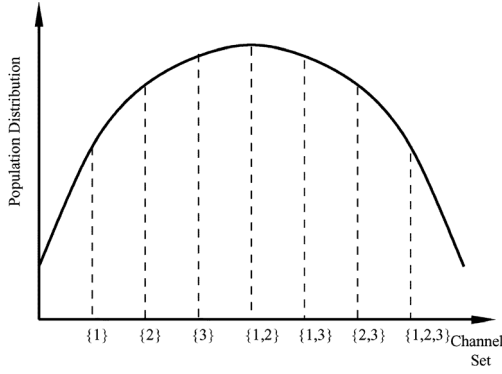


Fig. 9. Population distribution of mesh structure with three channels, beta distribution with parameters (2, 2).

channel sets. We use the beta distribution with parameters y, z to control the bandwidth and population distribution. Beta distribution is a general type of statistical distribution, with the probability function $P(x) = (1-x)^{z-1}x^{y-1}/B(y, z)$, where $B(y, z)$ is the beta function defined as $B(y, z) = (y-1)!(z-1)!/(y+z-1)!$ [29]. The reason why we use beta distribution to control the bandwidth and population distributions is that it can generate distributions with various shapes representing different scenarios in real systems. For details, please refer to Section V-B.

The population of each channel set is determined as follows: We first arrange the channel sets in lexicographical order and then assign a channel set a fraction f of the total number of peers N , which means the number of peers watching that channel set is $f * N$. We use the beta distribution to determine the fraction f . Fig. 9 illustrates an example of assigning population distribution to a mesh channel structure with three channels and beta distribution parameters (2, 2).

Since the goal of these simulations is to compare the *relative* order of the three designs, we set total upload bandwidth of each channel set as follows: For every group of simulations, we first calculate the total bandwidth demand based on the population of each channel set and the streaming rate of each channel. Then, we set the total upload bandwidth supply U to be equal to the total bandwidth demand. That is, we only consider the cases when the system is feasible. Finally, we use a similar method of obtaining population fraction to assign each channel set a fraction f' of the total upload bandwidth U , generated by the bandwidth distribution function, which means that the upload bandwidth of that channel set is $f' * U$. Note that the beta distribution for controlling upload bandwidth differs from the one for controlling population. For example, Fig. 10 shows how to assign bandwidth distribution for a three-channel system with chain channel structure, where the user sets are $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$, $\{2, 3\}$ and the fractions of bandwidth of the user sets are determined by the beta distribution with parameters (1, 1). *With this bandwidth setting, system-wide feasibility is guaranteed, which implies that ACA feasible condition always holds. Therefore, all the simulation results show the relative performance of the three designs.* For the reason why ACA feasible condition always holds, please refer to Sections III-B and III-C.

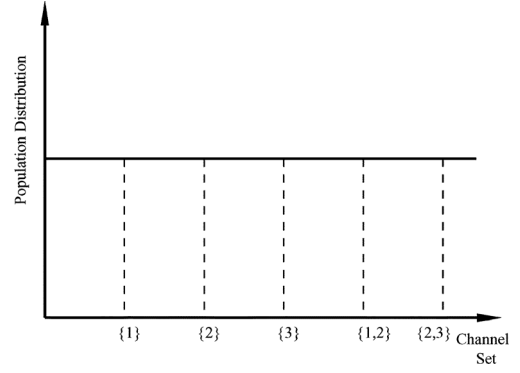


Fig. 10. Bandwidth distribution of chain structure with three channels, beta distribution with parameters (1, 1).

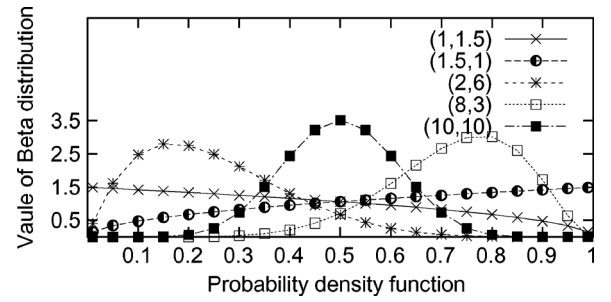


Fig. 11. Five population distributions with their corresponding beta parameters.

B. Simulation Parameters

The number of peers for all simulations is 100 000. Since both bandwidth and population distributions influence the design space of the three designs and the space determined by them is continuous, we try to choose as many representative cases as possible to approximate the continuous space. For the beta distribution for controlling population distribution, the parameters y and z vary from (1, 1) to (10, 10) with the step size 0.5. Therefore, there are total of $19 * 19 = 361$ population distributions. Among these population distributions, we first select five representative populations to report the evaluation results for the three designs and then present the results of all distributions.

The five population distributions are shown in Fig. 11. The population distribution with parameters (1.0, 1.5)⁷ denotes that the majority of users are watching some specific channel sets with a single channel (the channel sets are arranged in lexicographical order) and there is a large number of channel sets of multiple channels with small number of users. By contrast, the population distribution with parameters (1.5, 1.0) represents the opposite situation, where there is a large number of users watching multiple channels. Both cases reflect the situation of the long-tail channel popularity of current video streaming applications [30], [31], such as P2P-VoD and IPTV systems. Population distributions with parameters (2, 6) and (8, 3) represent the cases where there are some major events attracting most of

⁷We use the parameters (y, z) to represent a specific beta distribution hereinafter. For example, (1.0, 1.5) denotes the beta distribution with parameters (1.0, 1.5).

TABLE I
RELATIVE FEASIBLE SOLUTION SPACE SIZE OF THREE DESIGNS FOR 300 kb/s STREAMING RATE

Population Distribution	PCA mesh	PCA chain	PCA star	NBA mesh	NBA chain	NBA star	ACA mesh	ACA chain	ACA star
(1.0, 1.5)	98.6%	1.94%	22.2%	0%	0%	0%	100%	100%	100%
(1.5, 1.0)	99.4%	9.7%	23.3%	0%	0%	0%	100%	100%	100%
(2.0, 6.0)	98.9%	0.83%	20.2%	0%	0%	0%	100%	100%	100%
(8.0, 3.0)	99.7%	0.28%	31%	0%	0%	0%	100%	100%	100%
(10, 10)	98.9%	0.55%	46.5%	0%	0%	0%	100%	100%	100%
overall	99.1%	3.1%	35.2%	0%	0%	0%	100%	100%	100%

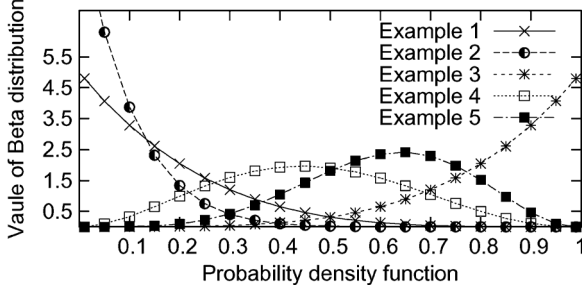


Fig. 12. Examples of other beta distributions used for control bandwidth and population distributions.

the users (e.g., Olympic Games live broadcasting), (2, 6) denotes that most users watch channel sets with a single channel, and (8, 3) denotes that most users watch channel sets with multiple channels. Finally, population distribution with parameters (10, 10) denotes a normal population distribution. The other 356 distributions are variations of the five distributions, and some examples are shown in Fig. 12.

Similar to controlling the populations of different channel sets, the bandwidth distribution of these channel sets is controlled by another beta distribution with parameters also varying from (1, 1) to (10, 10) with step size 0.5. The goal is to create different unbalanced bandwidth distributions among different channels covering the measurement studies [5], [6]. Therefore, for each specific population distribution, we evaluate a total number of $19 \times 19 = 361$ bandwidth distributions. For all population distributions, we evaluate a total number of $361 \times 361 = 130\,321$ cases. The *size of feasible region for a specific design with a specific channel structure* is roughly defined as the number of feasible cases over the number of totally studied cases (i.e., 361 cases with different bandwidth distributions). As an example, if the PCA design with a chain channel structure and a population distribution with parameters (10, 10) has 30 feasible cases, the size of feasible solution space for this case is $30/361 = 8.3\%$. The *size of feasible region for a specific design with all channel structures* is defined as the number of feasible cases over the total number of studied cases (i.e., $361 \times 361 = 130\,321$ cases with different bandwidth and population distributions). If the PCA design with a chain channel structure has 20 000 feasible cases, then the overall size of feasible region for this PCA with chain is $20\,000/130\,321 = 15.3\%$.

For the chain channel structure, there are 10 channels, and a peer can watch up to two consecutive channels. For the mesh channel structure, there are 10 channels, and a peer can arbitrarily join/subscribe to up to four channels with maximum 200 channel sets. For the star channel structure, there are 20 channels, and all peers watch a common channel and

another channel. We simulate four streaming rates of 300 kb/s, 900 kb/s, 1 Mb/s, and 1.5 Mb/s in different simulation groups.

In Sections V-C and V-D, we present the numerical results in two groups: 1) multichannel systems with homogeneous streaming rate; 2) multichannel systems with heterogeneous streaming rates for different channels.

C. Multichannel Systems With Homogeneous Streaming Rate

We study the size of feasible solution spaces for three designs with homogenous streaming rate across all channels. Initially, we simulate the NBA and PCA designs with a low streaming rate (300 kb/s) for all channel structures and for all bandwidth distributions and user population distributions. We summarize the results in Table I. The simulation results show that the feasible solution space (solution space for short) of NBA design with all channel structures is always empty because for systems with unbalanced bandwidth, without the channel-aware bandwidth allocation strategy, NBA design can rarely lead to a bandwidth allocation that satisfies the bandwidth demands for all channels. From Columns 1 to 3 of Table I, we can see that the solution spaces of PCA design increase in the order of chain, star, and mesh channel structures. The solution space of the ACA design is 100% for all channel structures since we simulate the scenarios where system feasibility (refer to Definition III-A.3) is always guaranteed.

We then simulate the NBA and PCA designs with a much higher streaming rate (1 Mb/s) for all channel structures and for all bandwidth distributions and population distributions, which corresponds to the high-definition videos (HD). The simulation results show that the solution space of PCA design does not change in HD scenarios. Therefore, we do not list the results. Similarly, the solution space of NBA design for all channel structures does not change with the increase of the streaming rate and is always empty. The solution space of the ACA design is 100% for all channel structures. Based on this group of simulations, we can conclude that for systems with homogenous streaming rate, the solution space is not affected by the streaming rate, and the channel structure has a greater impact on the feasibilities of the three designs. The solution space size depends on the bandwidth imbalance among different channels and channel structures.

D. Multichannel Systems With Heterogeneous Streaming Rates

Many of the commercial multichannel P2P streaming systems support heterogeneous streaming rates for different channels in order to provide different video qualities, such as high-definition videos and standard-definition videos. Therefore, in this section, we simulate multichannel systems with heterogeneous streaming rates to investigate their impact on the solution space of the three designs. For all of the following simulations,

TABLE II
RELATIVE FEASIBLE SOLUTION SPACE SIZE OF THREE DESIGNS FOR HETEROGENOUS STREAMING RATES

Population Distribution	PCA mesh	PCA chain	PCA star	NBA mesh	NBA chain	NBA star	ACA mesh	ACA chain	ACA star
(1.0, 1.5)	96.1%	0%	0.55%	0%	0%	0%	100%	100%	100%
(1.5, 1.0)	97%	0%	0.28%	0%	0%	0%	100%	100%	100%
(2.0, 6.0)	97.5%	0%	0%	0%	0%	0%	100%	100%	100%
(8.0, 3.0)	93.9%	0%	0%	0%	0%	0%	100%	100%	100%
(10, 10)	97.5%	0%	0%	0%	0%	0%	100%	100%	100%
overall	96.7%	0.04%	0.02%	0%	0%	0%	100%	100%	100%

the fractions of channels with streaming rate 300 kb/s, 900 kb/s, and 1.5 Mb/s are 40%, 30%, and 30%, respectively.

The results are summarized in Table II. The solution space of the *NBA* design with all channel structures is still empty due to the even higher bandwidth imbalance with heterogenous streaming rates. The streaming rate diversity greatly influences the solution space of the *PCA* design with chain and star channel structures, where the solution spaces of these two channel structures shrink to almost empty, as shown in Columns 2 and 3 in Table II. However, from Column 1 of Table II, for the mesh channel structure, the solution space of the *PCA* is almost the same as that of the *ACA* design. Therefore, when building a multichannel system with mesh channel structure, we can use the simple *PCA* design since it can achieve a similar performance to that of *ACA*. The solution space change of *PCA* design with the increased bandwidth imbalance due to heterogenous streaming rates implies that *PCA* with mesh channel structure has the ability to balance bandwidth among different channel, whereas chain and star channel structures eliminate such ability. We investigate the possible reason below.

We define the average number of views for a specific channel structure as the sum of the number of channels watched by a peer across all peers divided by the total number of peers in the system. For example, assuming that the system has two peers, peer 1 watches one channel, and peer 2 watches two channels. Therefore, the average number of views in the system is $1 + 2/2 = 1.5$. From the calculation, we can see that the average number of views depends on the channel structure and the population distribution. Fig. 13 illustrates the average number of views for different channel structures against all simulated population distributions. From this figure, we can see that for the chain and star structures, the average number of views is below 2. By contrast, the average number of views for mesh structure is almost always greater than 2.5, which results in a larger solution space for *PCA*. Intuitively, the average number of views reflects the overlap among different overlays corresponding to different channels. Higher average number of views implies higher ability of balancing the bandwidth among channels with *PCA* design. The *ACA* design can also benefit from this result since it can simply maintain the average number of views to be above some threshold instead of designing very complex schemes to maintain the helper group. For example, VUD [16] proposes a complex scheme to maintain the helper group.

VI. DISCUSSIONS AND CONCLUSION

In this paper, we focus on two fundamental problems in designing multichannel P2P streaming systems: 1) what are the

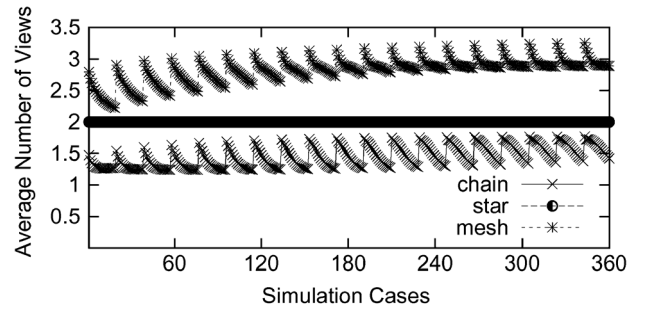


Fig. 13. Average number of views for different channel structures of different simulations.

general characteristics of existing and potential designs; 2) and which design can be used to achieve the desired streaming quality with the lowest implementation complexity.

To answer the first question, we develop simple models based on linear programming models and network-flow graphs for *NBA*, *PCA*, and *ACA* designs, which capture the main characteristics of cross-channel bandwidth allocation when designing multichannel systems. We also prove that the *ACA* model with overhead is NP-complete.

To answer the second question, we first study a special homogenous two-channel system and derive the closed-form results. Our results show that for this special case, there is no need to use the complex *ACA* design. The feasible solution space of *NBA* is much smaller than that of *PCA*. *NBA* can either support low-quality videos or sustain the high streaming quality while consuming more bandwidth than *PCA*. Furthermore, the simple two-channel case implies that not only do the bandwidth and population distributions influence the feasible solution space, but the channel structure of the system does as well.

Furthermore, we develop a C++-based simulator to numerically solve the cross-channel bandwidth allocation problems with various streaming rates, channel structures, bandwidth, and population distributions. The extensive numerical results show the following.

- 1) *NBA* design can rarely achieve desired streaming quality in general cases.
- 2) For the mesh channel structure, which is the case for general multichannel systems, the *PCA* design can achieve a similar performance as that of the *ACA* design even with heterogenous streaming rates, which indicates that we can build a general multichannel system with a simpler design.
- 3) For special chain and star channel structures, which correspond to special P2P applications, *PCA* cannot achieve the

desired streaming quality in most cases, and therefore the more complex *ACA* design should be used.

- 4) For multichannel systems, where the channels are isolated from one another (i.e., different channels do not overlap), *ACA* should always be used.

In addition, based on our simulations, channel structures play a critical role in determining feasible region sizes of the three designs, in that channel structures determine the fraction of peers subscribing to multiple channels, and these peers are necessary for enabling cross-channel bandwidth sharing. Compared to *NBA* and *PCA* designs, *ACA* design is more powerful since it allows a peer to subscribe to unwatched channels. However, there is a tradeoff between peer membership management complexity and the gain of cross-channel bandwidth sharing. It is very challenging to study this tradeoff, which is left as a future work in designing multichannel P2P streaming systems.

Due to simplicity reasons, our approaches for comparing the three designs have two limitations: 1) we do not study the system performance at transition states, which implies that our model lacks the ability of analyzing the performance under peer dynamics (e.g., channel switching, peer leaving); and 2) our models do not precisely model the implementation complexity of the three designs and our discussions are based on the intuition that *ACA* has the highest implementation complexity and *NBA* has the lowest implementation complexity. The latter will be worth studying in future work since it will have broader impact on analyzing the design of distributed systems.

APPENDIX A PROOF OF THEOREM 1

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 $\sum_{c:c \in \Theta} \sum_{\theta:c \in \theta} |S_\theta| r_c$, and the right-hand side is $\sum_{c:c \in \Theta} (\sum_{\theta:c \in \theta} x_c^\theta (\sum_{m \in S_\theta} u_m) + \sum_{\theta:c \notin \theta} y_c^\theta (\sum_{m \in S_\theta} u_m)) + \sum_{c:c \in \Theta} s_c$. By (10), $\sum_{c:c \in \Theta} (\sum_{\theta:c \in \theta} x_c^\theta (\sum_{m \in S_\theta} u_m) + \sum_{\theta:c \notin \theta} y_c^\theta (\sum_{m \in S_\theta} u_m)) = \sum_{m \in M} u_m$, which implies that inequality (13) holds.

(\Leftarrow): Since the demand of each channel c is $\sum_{\theta:c \in \theta} |S_\theta| r_c$, we order the channels in a nondecreasing order $\{c_1 \dots c_n\}$, where n is the total number of channels and $\sum_{\theta:c_i \in \theta} |S_\theta| r_{c_i} \leq \sum_{\theta:c_j \in \theta} |S_\theta| r_{c_j}$, if $i \leq j$. The total bandwidth supply is defined by the right-hand side of inequality (13). Since *ACA* design allows a user to subscribe to any channel, we can use the following water-filling approach to satisfy the bandwidth demand of a channel c_i : We allocate peers' upload bandwidth based on the ordered channels $\{c_1 \dots c_n\}$ to ensure that the allocated bandwidth for a specific channel c_i is exactly equal to the bandwidth demand $\sum_{\theta:c_i \in \theta} |S_\theta| r_{c_i}$. Then, we prove that constraint (9) is satisfied for every channel c by contradiction. Assume that $\exists k, \sum_{\theta:c_k \in \theta} |S_\theta| r_{c_k}$ cannot be satisfied, which indicates that the remaining bandwidth is less than $\sum_{\theta:c_k \in \theta} |S_\theta| r_{c_k}$. Since channels are ordered in a nondecreasing order based on their

respective bandwidth demands, $\forall p \in [k+1, n]$, bandwidth demand $\sum_{\theta:c_p \in \theta} |S_\theta| r_{c_p}$ is not satisfied. Let *Residual* denote the remaining bandwidth. Therefore, the total bandwidth supply is

$$\begin{aligned} & \sum_{j \in [1, k-1]} \sum_{\theta:c_j \in \theta} |S_\theta| r_{c_j} + \text{Residual} \\ & < \sum_{j \in [1, k-1]} \sum_{\theta:c_j \in \theta} |S_\theta| r_{c_j} + \sum_{p \in [k, n]} \sum_{\theta:c_p \in \theta} |S_\theta| r_{c_p} \quad (23) \end{aligned}$$

which contradicts with inequality (13). Therefore, constraint (9) is satisfied $\forall c \in \Theta$. ■

APPENDIX B PROOF OF THEOREM 2

In order to prove that the above problem is NP-complete, we introduce a known NP-complete problem *Exact Weight Perfect Matching* (EWPM) of a bipartite graph [32], [33].

EWPM: An edge weight bipartite graph and a positive integer α .

Question: Does there exist a perfect matching M with $\text{Weight}(M) = \alpha$?

Proof: Reducing from EWPM of a bipartite graph.

Step 1: Given a solution to Simplified *ACA-O* Decision, it can be verified in polynomial time since the verification time is bounded by the number of edges in the solution. Therefore, Simplified *ACA-O* Decision is in class NP.

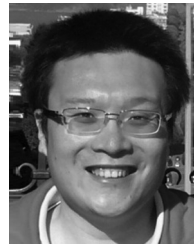
Step 2: Given an instance of EWPM, we can construct a network-flow graph as follows. We assume that the bipartite graph can be represented by $G' = (X, Y, E')$. We plan to construct a network-flow graph $G = (\{S, D, D', s, t\}, E)$. First, we merge all vertices in Y into one vertex y and divide y into y_1, y_2 connected by an edge with capacity α . We set $S = X$ and $D = y_1$ and $D' = y_2$ with edges E' connecting S and D whose edge capacities are the weights of E' . Then, we add s and t , where s connects all vertices in S with capacities $+\infty$ and t is connected to y_2 with the capacity $+\infty$. Finally, we place newly added edges in E . The construction process takes polynomial time. Therefore, if there is a polynomial-time algorithm that solves Simplified *ACA-O*, EWPM can be solved polynomially.

Step 3: Given an instance of Simplified *ACA-O*, $G = (\{S, D, D', s, t\}, E)$. We construct an instance of EWPM, $G' = (X, Y, E')$ as follows. We first set α to be the sum of capacities of edges connecting vertices in D and D' . Then, we remove s and t and edges connecting to them. Next, we set $X = S$ and $Y = D$. The capacities of edges connecting vertices in S and D are set to be the weight of edges connecting X and Y . We remove D' and edges connecting to them. Since the number of suppliers is larger than or equal to the number of consumers, we might need to merge some vertices in Y as well as the connected edges to maintain an even number of vertices to guarantee a perfect matching. Finally, we add a pair of vertices a and b into X and Y , respectively. The weight of edge that connects a and b is the difference between α and sum of weights of all edges in E' except for (a, b) . Since system-wide feasible condition holds, the weight of edge (a, b) is nonnegative.

Step 4: Combining all above steps, we prove that Simplified *ACA-O* is NP-complete and that it is equivalent to EWPM in its complexity. ■

REFERENCES

- [1] UUSee, "UUSee," [Online]. Available: <http://www.uusee.com>
- [2] PPStream, "PPStream," 2006 [Online]. Available: <http://www.ppstream.com>
- [3] X. Zhang, J. Liu, B. Li, and Y. S.-P. Yum, "CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, vol. 3, pp. 2102–2111.
- [4] PPLive, "PPLive," 2007 [Online]. Available: <http://www.pplive.com>
- [5] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1672–1687, Dec. 2007.
- [6] X. Hei, Y. Liu, and K. Ross, "Inferring network-wide quality in P2P Live streaming systems," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1640–1654, Dec. 2007.
- [7] M. Zhang, Q. Zhang, and S. Yang, "Understanding the power of pull-based streaming protocol: Can we do better?," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1678–1694, Dec. 2007.
- [8] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 919–927.
- [9] D. Wu, C. Liang, Y. Liu, and K. W. Ross, "View-upload decoupling: A redesign of multi-channel P2P video systems," in *Proc. IEEE INFOCOM Mini-Conf.*, Rio de Janeiro, Brazil, 2009, pp. 1–6.
- [10] Microsoft, Redmond, WA, "Silverlight RichMedia Playe," 2011 [Online]. Available: <http://www.microsoft.com/media/en/us/resources/microsoft-media-entertainment-white-papers.aspx>
- [11] C. Wu and B. Li, "Strategies of conflict in coexisting streaming overlays," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 481–489.
- [12] M. Wang, L. Xu, and B. Ramamurthy, "A flexible divide-and-conquer protocol for multi-view peer-to-peer live streaming," in *Proc. IEEE P2P*, 2009, pp. 291–300.
- [13] D. A. Tran, K. A. Hua, and T. Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 2003, vol. 2, pp. 1283–1292.
- [14] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth multicast in a cooperative environment," in *Proc. ACM SOSP*, Lake Bolton, NY, Oct. 2003, pp. 298–313.
- [15] C. Wu, B. Li, and S. Zhao, "Multi-channel live P2P streaming: Refocusing on servers," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 1355–1363.
- [16] D. Wu, Y. Liu, and K. W. Ross, "Queueing network models for multi-channel P2P live streaming systems," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, 2009, pp. 73–81.
- [17] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, "Performance bounds for peer-assisted live streaming," *Perform. Eval. Rev.*, vol. 36, no. 1, pp. 313–324, 2008.
- [18] L. Massoulié, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized decentralized broadcasting algorithms," in *Proc. IEEE INFOCOM*, 2007, pp. 1073–1081.
- [19] F. Liu, B. Li, L. Zhong, B. Li, H. Jin, and X. Liao, "Flash crowd in P2P live streaming systems: Fundamental characteristics and design implications," *IEEE Trans. Parallel Distrib. Syst.*, 2011, to be published.
- [20] Y. Zhou, D. Chiu, and J. Lui, "A simple model for analyzing P2P streaming protocols," in *Proc. IEEE ICNP*, Beijing, China, Oct. 2007, pp. 226–235.
- [21] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: Optimal performance trade-offs," *Perform. Eval. Rev.*, vol. 36, no. 1, pp. 325–336, 2008.
- [22] M. Wang, L. Xu, and B. Ramamurthy, "Linear programming models for multi-channel P2P streaming systems," in *Proc. IEEE INFOCOM Mini-Conf.*, 2010, pp. 1–5.
- [23] M. Wang, L. Xu, and B. Ramamurthy, "Comparing multi-channel peer-to-peer video streaming system designs," in *Proc. IEEE LANMAN*, Long Branch, NJ, 2010, pp. 1–6.
- [24] M. Wang, "Multi-channel peer-to-peer streaming systems as resource allocation problems," Ph.D. dissertation, University of Nebraska–Lincoln, Lincoln, NB, May 2011.
- [25] 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 *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 1031–1039.
- [26] F. Wang, J. Liu, and Y. Xiong, "Stable peers: Existence, importance, and application in peer-to-peer live video streaming," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 1364–1372.
- [27] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, MA: Athena Scientific, Feb. 1997.
- [28] IBM, Armonk, NY, "CPLEX," [Online]. Available: <http://www.ilog.com/products/cplex/>
- [29] Wolfram Research, Champaign, IL, "Beta-distribution," 2012 [Online]. Available: <http://mathworld.wolfram.com/BetaDistribution.html>
- [30] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," *Oper. Syst. Rev.*, vol. 40, no. 4, pp. 333–344, 2006.
- [31] T. Qiu, Z. Ge, S. Lee, J. Wang, Q. Zhao, and J. Xu, "Modeling channel popularity dynamics in a large IPTV system," in *Proc. ACM SIGMETRICS*, 2009, pp. 275–286.
- [32] M. Gori, M. Maggini, and L. Sarti, "Exact and approximate graph matching using random walks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 7, pp. 1100–1111, Jul. 2005.
- [33] G. Zhu, X. Luo, and Y. Miao, "Exact weight perfect matching of bipartite graph is NP-complete," in *Proc. WCE*, London, U.K., Jul. 2008, vol. 2, pp. 1–6.



Miao Wang (M'11) received the B.E. and M.E. degrees in computer science from Xi'an Jiaotong University, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science from the University of Nebraska–Lincoln (UNL) under the supervision of Dr. Lisong Xu and Dr. Byrav Ramamurthy in May 2011.

He is a Software Research Engineer with A10 Networks, Inc., San Jose, CA. His research interests span peer-to-peer computing, multimedia streaming, high-availability data center networks, and network congestion control.

Dr. Wang received a UNL Folsom Doctoral Dissertation Honorable Mention for his thesis "Multi-Channel Peer-to-Peer Streaming Systems as Resource Allocation Problems."



Lisong Xu (M'04) received the B.E. and M.E. degrees in computer science from the University of Science and Technology Beijing, Beijing, China, in 1994 and 1997, respectively. He completed his thesis "Performance Analysis Of Optical Burst Switched Networks" under the direction of Dr. Harry Perros and Dr. George Rouskas, and received the Ph.D. degree in computer science from North Carolina State University, Raleigh, in 2002.

From 2002 to 2004, he was a Post-Doctoral Research Fellow with North Carolina State University, working on congestion control for high-speed long-distance networks (TCP-BIC/CUBIC) with Dr. Injong Rhee, and on the PhysViz Project with Dr. James Lester. He is currently an Associate Professor in computer science and engineering with the University of Nebraska–Lincoln.

Dr. Xu is a recipient of the NSF CAREER Award in 2007, the UNL CSE Student Choice Outstanding Teaching Award in 2006, 2007, 2008, 2010, and 2011, and the UNL College Distinguished Teaching Award in 2011.



Byrav Ramamurthy (S'96–M'98) received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Madras, India, in 1993, and the M.S. and Ph.D. degrees in computer science from the University of California, Davis, in 1995 and 1998, respectively.

He is currently a Professor with the Department of Computer Science and Engineering, University of Nebraska–Lincoln (UNL). He is the author of the book *Design of Optical WDM Networks—LAN, MAN and WAN Architectures* (Kluwer/Springer, 2000) and a coauthor of the book *Secure Group Communications Over Data Networks* (Kluwer/Springer, 2004). His research areas include optical and wireless networks, peer-to-peer networks for multimedia streaming, network security and telecommunications. His research work is supported by the U.S. National Science Foundation, U.S. Department of Energy, U.S. Department of Agriculture, AT&T Corporation, Agilent Tech., Ciena, HP and OPNET Inc.

Prof. Ramamurthy served as the Chair of the IEEE Communication Society's Optical Networking Technical Committee (ONTC). He served as the IEEE INFOCOM 2011 TPC Co-Chair.