Queuing Network Models for Multi-Channel P2P Live Streaming Systems

Di Wu†, Yong Liu‡, Keith W. Ross†
†Department of Computer & Information Science
‡Department of Electrical & Computer Engineering
Polytechnic Institute of NYU, Brooklyn, NY, USA 11201
Email: {dwu, yongliu, ross}@poly.edu

Abstract—**In recent years there have been several large-scale deployments of P2P live video systems. Existing and future P2P live video systems will offer a large number of channels, with users switching frequently among the channels. In this paper, we develop infinite-server queueing network models to analytically study the performance of multi-channel P2P streaming systems. Our models capture essential aspects of multi-channel video systems, including peer channel switching, peer churn, peer bandwidth heterogeneity, and Zipf-like channel popularity. We apply the queueing network models to two P2P streaming designs: the isolated channel design (ISO) and the View-Upload Decoupling (VUD) design. For both of these designs, we develop efficient algorithms to calculate critical performance measures, develop an asymptotic theory to provide closed-form results when the number of peers approaches infinity, and derive near-optimal provisioning rules for assigning peers to groups in VUD. We use the analytical results to compare VUD with ISO. We show that VUD design generally performs significantly better, particularly for systems with heterogeneous channel popularities and streaming rates.**

I. Introduction

In recent years there have been several large-scale industrial deployments of P2P live video systems, including Coolstreaming [1], PPLive[2], PPStream [3], UUSee [4] and Sopcast [5]. Recent measurement studies have verified that hundreds of thousands of users can simultaneously participate in these systems [6], [7].

Almost all live P2P video systems offer multiple channels. PPLive and its competitors each have over 100 channels; future systems with user-generated channels will likely have thousands if not millions of live channels. In P2P live video streaming systems, churn occurs on two different time scales: peers enter and leave the video application on long time scales; and peers change channels on short time scales. A recent study of a cable television system showed that users switch channels frequently [8].

A common practice in P2P video today is to organize peers viewing the same channel into a swarm, with peers in the same swarm redistributing video chunks exclusively to each other. We refer to such a design as *isolated-channel* or more simply as *ISO* P2P streaming systems. Unfortunately, this channel churn brings enormous instability to ISO P2P streaming systems. For example, when a peer switches from channel A to channel B, it stops uploading to its neighbors in swarm A. Those neighbors have to quickly find a new data feed to maintain steady video inflow. And in swarm B, the newly joining peer has to quickly find new neighbors with enough bandwidth and content to download from. Recent measurement studies [6], [7] of PPLive have identified several fundamental performance problems for isolated-channel systems, including large channel switching delay and playback lag; and poor small channel performance. Here, small channel refers to the channel with low popularity.

Recently we proposed a new approach to multi-channel P2P streaming, which we refer to as *View-Upload Decoupling (VUD)* [9]. VUD strictly decouples what a peer uploads from what it views, bringing stability to multi-channel systems and enabling cross-channel resource sharing. In VUD, each peer is assigned to one or more channels, with the assignments made *independently of what the peer is viewing*. For each assigned channel, the peer distributes (that is, uploads) the channel. This has the effect of creating a semi-permanent *distribution swarm* for each channel, which is formed by peers responsible for uploading that channel. VUD introduces additional overhead since each peer now needs to upload into its assigned swarm as well as to peers outside the swarm that want to view the channel. However, it is shown in [9] that by dividing each channel into multiple substreams and assigning peers to substream swarms, VUD's overhead can be made insignificant. In [9] simulation results showed that VUD enhanced with substreaming can reduce switching delay, chunk miss ratio and playback lags. Moreover, VUD greatly improves the streaming quality of small channels.

The goal of this paper is develop a tractable analytic theory for multi-channel P2P live streaming systems. Our analytic theory seeks to model both ISO and VUD designs for multi-channel P2P live streaming systems. Our models seek to capture the heterogeneous peers, multiple channels, and channel churn occurring at much faster time scale than peer churn. To this end, we shall show how both closed and open infinite-server queuing networks are appropriate and tractable tools for modeling multi-channel P2P streaming systems. The contributions of this paper are as follows:

- We develop infinite-server queuing network models that capture user channel-switching and peer churn behavior for both ISO and VUD.
- We use the queuing network models to derive nearly closed-form expressions for central performance metrics

such as the probability of system-wide universal streaming and the average number of happy channels. These expressions provide simple and efficient algorithms for calculating the performance metrics.
- We use the theory to explore qualitative properties of VUD and ISO. For example, we use the theory to study the asymptotic behavior of these systems. For VUD we derive an explicit critical parameter which determines when the probability of universal streaming goes to one as the number of peers approaches infinity. This asymptotic theory also provides a natural heuristic for dimensioning the VUD groups for large systems. Moreover, the theory enables us to prove that for VUD with substreaming, to optimize performance, each substream within a channel should have the same rate and the same resource index.
- We indicate how that theory can further be leveraged to optimally dimension the substream groups in VUD.
- We use the theory developed here to compare the performance of VUD and ISO. The results confirm simulation results in [9], showing that VUD can provide significantly better performance than ISO.

This paper is organized as follows. We conclude this section with a brief discussion of related work. In Section II, we establish the basic queuing network model for user behavior in multi-channel P2P streaming systems and define the related performance metrics. In Sections III and IV we further develop the queuing network models for VUD and ISO, respectively. In Section V, we briefly indicate how our results can be extended to allow for peer churn in addition to channel churn. In Section VI, we provide numerical results based on our derived expressions to compare the performance of ISO and VUD design. In Section VII, we propose a more refined heuristic for VUD streaming. Finally, in Section VIII, we summarize this paper and discuss future work.

A. Related Work

There are only a few theoretical performance studies of P2P streaming systems. Zhou et al. [10] developed a simple probability model to compare different chunk selection and peer selection strategies. In [11], Small et al. investigated the scaling laws and tradeoffs in P2P streaming systems. Kumar et al. [12] provided a stochastic fluid model to expose the fundamental characteristics and limitations of ISO P2P streaming systems. Massoulie et al. [13] studied the problem of efficient decentralized broadcasting in both edge-capacitated and node-capacitated networks, and proposed completely decentralized algorithms. Bonald et al. [14] proved that random peer, latest useful chunk algorithm can achieve dissemination at an optimal rate and within an optimal delay, up to an additive constant term. In [15], Liu et al. derived the performance bounds for minimum server load, maximum streaming rate, and minimum tree depth under different peer selection constraints.

However, all of the above papers focus on single-channel (ISO) streaming systems. In terms of multi-channel P2P streaming, to our knowledge there is only one related paper [16], which considers the problem of server capacity provisioning among multiple channels. To the best of our knowledge, this is the first paper on the modeling and stochastic analysis of multi-channel P2P streaming systems.

II. Closed Queuing Network Model

In P2P live video streaming systems, there are generally two types of churn: *peer churn* corresponding to peers entering and leaving the video application on long time scales; and *channel churn* corresponding to peers changing channels. When peers are always on-devices, such as Set-Top-Boxes (STBs) or home routers, then there is typically little peer churn, as the devices can participate in the distribution at all times. Peer churn is present in PC environments, where users turn on and off their PCs as well as start and quit the video streaming application. This peer churn occurs at a much slower time scale than channel churn. In this paper, we will mostly focus on the case of a fixed number of peers, n, modeling the always-on device environments [8] and slow-scale peer churn. However, in Section V we will show how the theory can easily be extended to model P2P live streaming systems with both peer and channel churn.

To build our queuing models, we will need to introduce some notations:

- Let $\mathcal{N}$ be the set of all peers; let n be the total number of peers.
- Let u_i be the upload rate of peer i. Note that our model allows for heterogeneous upload rates.
- Let J be the number of channels.
- Let r_j be the streaming rate (in kbps) of channel j. Thus, the channels are permitted to have different streaming rates.
- Let v_j denote the server rate for channel j.
- Let $1/\mu_j$ denote the expected amount of time a peer continuously views channel j. The distribution of the viewing time is arbitrary. For example, to model users who rapidly switch among channels before settling into a channel for an extended period of time, the distribution could have a lot weight around 10 seconds and again a lot of weight around one hour.
- Let p_{ij} denote the probability that a peer switches from channel i to channel j. Thus a peer visits channel i for a random period of time with mean $1/\mu_i$ then switches to channel j with probability p_{ij}. Let $\mathbf{P}$ denote the corresponding $J \times J$ matrix.

Now that we have established the basic model for user-behavior, we now derive the joint distribution of the number of peers viewing the J channels. To this end, let $\lambda = (\lambda_1, \cdots, \lambda_J)$ be the unique probability distribution that satisfies $\lambda = \lambda\mathbf{P}$. Let $\rho_j = \lambda_j/\mu_j$. Note that λ_j is the relative arrival rate into channel j, and ρ_j is the relative channel popularity. Small channels have relatively low values of ρ_j. For convenience, normalize the ρ_j's so that $\rho_1 + \cdots + \rho_J = 1$.

Let M_j be a random variable denoting the number of peers viewing channel j. For simplicity, let us assume that each of the n peers is viewing some channel. (This assumption can be relaxed without loss of tractability.) Because the total number of peers viewing channels is fixed at n, we have $M_1 + \cdots + M_J = n$. Thus, the random variables $M_1, \ldots, M_J$ are dependent. By viewing the system as an infinite-server Jackson queuing network [17], with each channel being a node in the network, and each peer as a customer which sojourns at node j for a random amount of time with mean $1/\mu_j$, we immediately arrive at the following result:

Lemma 2.1: For any $m_1, \ldots, m_J$ with $m_1 + \cdots + m_J = n$, we have

$$P(M_1 = m_1, \ldots, M_J = m_J) = n! \frac{\rho_1^{m_1}}{m_1!} \cdots \frac{\rho_J^{m_J}}{m_J!} \quad (1)$$

Note that $(M_1, \ldots, M_J)$ has a multinomial distribution. It follows directly from the lemma that $E[M_j] = n\rho_j$ and that M_j has the binomial distribution

$$P(M_j = m_j) = \frac{n!}{m_j!(n-m_j)!} \rho_j^{m_j} (1-\rho_j)^{(n-m_j)}$$

We remark that we can further generalize this model of steady-state user behavior by allowing for different classes of switching behavior. For example, we can have two classes of customers: one that cycles through all of the channels; and the other that randomly visits a specific subset of channels. Each class would have its own channel transition matrix $\mathbf{P}$. We could use classical results from multi-class queuing networks to provide the steady-state distribution of the multiclass system. However, in order to not complicate the presentation, we restrict our attention to one class of switching behavior in this paper.

A. Performance Metrics

We say that *universal streaming* occurs when every peer is receiving the channel it is viewing at the streaming rate of the channel. When the system is in the state of universal streaming, all peers in the system are satisfied. For a given design, the system may be enjoying universal streaming at certain times but not at all times. Thus, we can talk about the steady-state probability (equivalently, the steady-state fraction of time) the system is in universal streaming. Another natural performance metric is the expected number of channels which enjoy universal streaming (also called "*happy channels*"). Below we make these definitions more precise. The following definitions apply for both ISO and VUD designs.

Definition The *resource index* for channel j is defined as

$$\sigma_j(M_j) = \frac{b_j - o_j}{d_j(M_j)}$$

where b_j is the total upload bandwidth available for channel j, $d_j(M_j)$ is the total bandwidth required to achieve universal streaming in channel j given the number of viewers to be M_j, and o_j is VUD's bandwidth overhead for channel j.

Definition The *probability of universal streaming in channel j* is defined as

$$PU_j = P(\sigma_j(M_j) \geq 1)$$

The *probability of system-wide universal streaming across all the channels* is defined as

$$PS = P(\sigma_j(M_j) \geq 1, \;\; j = 1, \ldots, J)$$

The *average number of happy channels* is defined to be:

$$HC = \sum_{j=1}^{J} P(\sigma(M_j) \geq 1)$$

In this paper we focus on two natural performance measures: the probability of system-wide universal streaming PS and the average number of happy channels HC. We shall derive explicit expressions for these performance measures for both VUD and ISO.

III. Viewing-Uploading Decoupling (VUD)

We now describe more specifically VUD. Each channel is divided into substreams (for example, to create S substreams for a channel, the channel source could assign every Sth constant-size chunk to a substream). Let S_j denote the total number of substreams for channel j. For each substream, there is a subset of peers, called a *substream distribution group*, that is responsible for obtaining video chunks from the channel server, redistributing the substream chunks among the peers within the group, and distributing the substream chunks to peers that want to view the corresponding channel. Note if a peer is viewing channel j, then the peer needs to draw substreams from S_j different groups. Below we list some critical properties and observations about VUD:

1) The groups are semi-permanent, that is, the groups remain constant over medium time scales, and do not change as peers channel surf. If a peer is assigned to a distribution group for a substream, then it seeks to receive the complete substream.
2) Intuitively, the aggregate upload capacity of a substream distribution group should reflect the demand for the substream. The greater the average channel demand, the larger the corresponding substream groups.
3) Intuitively, the peers in a distribution group should be chosen in regions that match the geographical demand. However, in order to expose the intrinsic advantages of VUD design, we postpone locality considerations to subsequent work.
4) Within a distribution group, we do not require the deployment of a specific distribution mechanism. We allow for trees [18], [19], mesh-pull [1], [20], [21] and meshes with push-pull [22]. However, because the substream distribution groups are relatively stable, we should be able to employ tree-based mechanisms, which generally provide better delay performance than mesh-based mechanisms.

Let r_j^s be the rate of the sth substream in channel j. Thus, $r_j^1 + \cdots + r_j^{S_j} = r_j$. Let $\mathcal{N}_j^s$ be the set of peers handling substream s for channel j and let n_j^s be the number of such peers. Similar to the definition of channel resource index, we define the resource index for substream s in channel j.

Definition The *resource index for substream s in channel j* is defined as

$$\sigma_j^s(M_j) = \frac{b_j^s - o_j^s}{d_j^s(M_j)} = \frac{v_j^s + \sum_{i\in\mathcal{N}_j^s} u_i - n_j^s r_j^s}{M_j r_j^s}$$

where $b_j^s = v_j^s + \sum_{i\in\mathcal{N}_j^s} u_i$, $o_j^s = n_j^s r_j^s$ and $d_j^s(M_j) = M_j r_j^s$.

Definition In VUD with substreaming, the *probability of system-wide universal streaming* is defined as

$$PS = P(\sigma_j^s(M_j) \geq 1,\ s = 1,\dots,S_j,\ j = 1,\dots,J)$$

Our goal now is to develop an efficient algorithm for calculating the key performance measures PS and HC for VUD. To this end, let

$$\phi_j^s = \lfloor \frac{v_j^s + \sum_{i\in\mathcal{N}_j^s} u_i - n_j^s r_j^s}{r_j^s} \rfloor$$

and

$$\delta_j = \min_{1\leq s\leq S_j} \phi_j^s = \min_{1\leq s\leq S_j} \lfloor \frac{v_j^s + \sum_{i\in\mathcal{N}_j^s} u_i - n_j^s r_j^s}{r_j^s} \rfloor$$

Theorem 3.1: In VUD, the *probability of system-wide universal streaming* is

$$PS = P(M_j \leq \delta_j,\ j = 1,\dots,J\} = \sum_{\mathbf{m}\in\mathcal{M}} n! \frac{\rho_1^{m_1}}{m_1!} \cdots \frac{\rho_J^{m_J}}{m_J!}$$

where $\mathcal{M} = \{(m_1,\dots,m_J):\ m_1 + \cdots m_J = n,\ m_j \in \{0,1,\dots,\delta_j\},\ j = 1,\dots,J\}$. The *expected number of happy channels* is

$$HC = \sum_{j=1}^{J} P(M_j \leq \delta_j) = \sum_{j=1}^{J}\sum_{k=0}^{\delta_j} \binom{n}{k} \rho_j^k (1-\rho_j)^{n-k} \quad (2)$$

Proof:

$$\begin{aligned} PS &= P(\sigma_j^s(M_j) \geq 1,\ s = 1,\dots,S_j,\ j = 1,\dots,J) \\ &= P(M_j \leq \phi_j^s\ s = 1,\dots,S_j,\ j = 1,\dots,J) \\ &= P(M_j \leq \delta_j,\ j = 1,\dots,J\} \end{aligned} \quad (3)$$

The result for PS follows by combining (3) with Lemma 2.1. The result for HC has a similar proof. ■

Theorem 3.1 leads to an efficient convolution algorithm for calculating the the system-wide universal streaming probability PS. Specifically, let

$$\mathcal{M}_j(m) = \{(m_1,\dots,m_j):\ m_1 + \cdots m_j = m,\ m_i \in \{0,1,\dots,\delta_i\},\ i = 1,\dots,j\}$$

and

$$q_j(m) = \sum_{\mathbf{m}\in\mathcal{M}_j(m)} m! \prod_{i=1}^{j} \frac{\rho_i^{m_i}}{m_i!}$$

Then it follows from Lemma 2.1 that

$$PS = q_J(n) = \sum_{m_J=0}^{\delta_J} q_{J-1}(n - m_J) \cdot \frac{\rho_J^{m_J}}{m_J!}$$

With convolution, the computational effort to calculate the universal streaming probability is $O(Jn^2)$ [23].

A. Optimal Substreaming

In a channel with rate r_j and S_j substreams, for a given $\mathcal{N}_j$, one natural question is how to find an allocation of $\{\mathcal{N}_j^1,\dots,\mathcal{N}_j^{S_j}\}$ and $\{r_j^1,\dots,r_j^{S_j}\}$ to maximize the probability of universal streaming in channel j.

Theorem 3.2: For a given allocation $\mathcal{N}_j$ for channel j, to maximize the probability of universal streaming, the channel is divided into equal-rate substreams and the resource index of each substream is equal.

Proof: To maximize the probability of universal streaming, we need to maximize the number of supported viewers $\delta_j = \min_s(\frac{b_j^s}{r_j^s} - n_j^s)$. The problem is transformed into the following optimization problem:

$$\text{Maximize} \quad \min_s\{\frac{b_j^s}{r_j^s} - n_j^s\} \quad (4)$$

$$\text{Subject to:} \quad \sum_{s=1}^{S_j} b_j^s = b_j; \sum_{s=1}^{S_j} r_j^s = r_j; \sum_{s=1}^{S_j} n_j^s = n_j$$

To simplify the notation, let $x_s = \frac{b_j^s}{b_j}, y_s = \frac{r_j^s}{r_j}, z_s = \frac{n_j^s}{n_j}, S = S_j, a = \frac{b_j}{r_j}, c = n_j$, then the above problem is simplified as:

$$\text{Maximize} \quad \min_s\{a\frac{x_s}{y_s} - cz_s\} \quad (5)$$

$$\text{Subject to:} \quad \sum_{s=1}^{S} x_s = 1; \sum_{s=1}^{S} y_s = 1; \sum_{s=1}^{S} z_s = 1;$$

Let $t = \min_s\{a\frac{x_s}{y_s} - cz_s\}$. This problem can be further transformed into the following standard non-linear minimization problem:

$$\text{Minimize} \quad (-t) \quad (6)$$

$$\begin{aligned} \text{Subject to:} \quad & \sum_{s=1}^{S} x_s = 1; \sum_{s=1}^{S} y_s = 1; \sum_{s=1}^{S} z_s = 1; \\ & -ax_s + cy_s z_s + ty_s \leq 0, s = 1,\dots,S; \end{aligned}$$

We can use Lagrangian multiplier [24] to solve it. The Lagrangian of the above problem is given by

$$L = (-t) + \sum_{s=1}^{S} \beta_s(-ax_s + cy_s z_s + ty_s) + \nu_1(\sum_{s=1}^{S} x_s - 1) + \nu_2(\sum_{s=1}^{S} y_s - 1) + \nu_3(\sum_{s=1}^{S} z_s - 1)$$

Based on Karush-Kuhn-Tucker conditions [24], we have $x_s = \frac{\nu_2\nu_3}{ac}, y_s = -\frac{\nu_3}{c}, z_s = \frac{\nu_2 + t}{c}, s = 1,\dots,S$. In that case, the resource index of substream s in channel j is given by

$$\sigma_j^s(M_j) = \frac{b_j^s - n_j^s r_j^s}{r_j^s M_j} = \frac{1}{M_j}(a\frac{x_s}{y_s} - cz_s) = \frac{t}{M_j}$$

$\sigma_j^s(M_j)$ is equal for all the substreams. In addition, as y_s is independent of the index s and $y_s = \frac{r_j^s}{r_j}$, we can also derive that r_j^s is equal for all $s = 1,\dots,S_j$. ■

B. Asymptotic Analysis

In this section we use the model developed in this paper to explore VUD in a natural asymptotic regime, namely, when the number of peers approaches infinity. Our goal is to determine under what conditions PS, the probability of system-wide universal streaming, is high for a system with a large number of peers. In order to see the forest through the trees, we derive the results for the special case when each channel has a single substream.

1) Homogeneous Systems: Let's first consider the homogeneous case with all $u_i = u$ with $u > r_j$ for all $j = 1, \ldots, J$. A natural asymptotic regime is to let $n \to \infty$ and $n_j = K_j n$ for constants K_j, $j = 1, \ldots, J$. Thus, in this regime, the size of each distribution group goes to infinity, the relative sizes of the distribution groups are fixed. Note that $K_1 + \cdots + K_J = 1$ by definition. We will explore in this subsection how we can asymptotically dimension the groups (by choosing the K_j's) to maximize PS, the probability of system-wide universal streaming.

Because M_j/n almost surely approaches ρ_j as $n \to \infty$, it is straightforward to show that $\sigma_j(M_j)$ almost surely approaches $K_j(u - r_j)/r_j\rho_j$ as $n \to \infty$. Thus, channel j has universal streaming in the limit if and only if

$$K_j \geq \frac{r_j \rho_j}{u - r_j}$$

Let

$$\alpha = \sum_{j=1}^{J} \frac{r_j \rho_j}{u - r_j},$$

We refer to α as the *critical parameter*. We therefore have the following result:

Theorem 3.3: If $\alpha > 1$ then for all choices of $K_1, \ldots, K_J$, PS, the probability of system-wide universal streaming goes to zero. If $\alpha \leq 1$, then for any $K_1, \ldots, K_J$ with $K_j \geq r_j\rho_j/(u - r_j)$ and $K_1 + \cdots + K_J = 1$, the probability of system-wide universal streaming goes to 1. (One such choice is $K_j = r_j\rho_j/\alpha(u - r_j)$.)

The above theorem indicates that, for large systems, the performance will be poor when the critical parameter, α, exceeds 1. When $\alpha < 1$, the above theorem provides a simple heuristic for dimensioning the group sizes n_j, $j = 1, \ldots, J$: we can simply set $n_j = nr_j\rho_j/\alpha(u - r_j)$. Finally, it follows that to maximize HC for large systems, we order $r_j\rho_j/(u-r_j)$ from lowest to highest, assign $K_j = r_j\rho_j/(u - r_j)$ beginning at $j = 1$ until $K_1 + \ldots + K_{t+1} \geq 1$. We set $K_{t+1} = \cdots = K_J = 0$. Also observe the interesting parallel to loss networks, for which the case $\alpha = 1$ is analogous to the critically loaded regime for loss networks [25], [23].

2) Heterogeneous Systems: We now examine the asymptotics for heterogeneous systems. To keep the exposition simple, we consider here only two peer types: peers with low upload rate u^l and peers with high upload rates u^h. Suppose that the fraction of low upload rate peers is fixed and denoted by f. Thus the average upload rate of the peers is $u^l f + u^h(1 - f)$. We now consider an asymptotic regime $n_j^l = K_j^l n$ and $n_j^h = K_j^h n$ with $\sum_j K_j^l = f$ and $\sum_j K_j^h = 1 - f$. We can show that $\sigma_j(M_j) \geq 1$ asymptotically if and only if

$$(u^l - r_j)K_j^l + (u^h - r_j)K_j^h \geq r_j\rho_j \tag{7}$$

With $\xi_j := u^l - r_j$ and $\zeta_j := u^h - r_j$ and $\eta_j := r_j\rho_j$, we have the following result.

Theorem 3.4: The probability of system-wide universal streaming approaches 1 if there is a solution $((K_j^l, K_j^h),\ \ j = 1, \ldots, J)$ to the following set of linear equations:

$$\xi_j K_j^l + \zeta_j K_j^h \geq \eta_j, \quad j = 1, \ldots, J$$

$$\sum_{j=1}^{J} K_j^l = f; \ \ \sum_{j=1}^{J} K_j^h = 1 - f$$

If there is no solution, the probability of system-wide universal streaming approaches 0.

The above theorem leads to a natural heuristic for assigning peers to distribution groups in large system with n peers. We first solve the following optimization problem:

$$\begin{aligned} \text{Maximize} \quad & \min_j \{\xi_j K_j^l + \zeta_j K_j^h - \eta_j\} \\ \text{Subject to:} \quad & \sum_{j=1}^{J} K_j^l = f; \ \ \sum_{j=1}^{J} K_j^h = 1 - f \end{aligned} \tag{8}$$

We then set $n_j^l = K_j^l n$ and $n_j^h = K_j^h n$. Note that the above max-min optimization problem is easily transformed into a simple linear programming problem involving $2J$ variables and $J + 2$ constraints.

If all channels have the same rate r, then $\xi_j = \xi$ and $\zeta_j = \zeta$ become homogeneous, and the feasibility problem is straightforward to solve. In particular, there is a feasible solution if and only if $\alpha \leq 1$ where

$$\alpha = \frac{r}{(u^l - r)f + (u^h - r)(1 - f)}$$

One such feasible solution is $K_j^l = \rho_j f$ and $K_j^h = \rho_j(1 - f)$.

IV. Performance of ISO Design

We now leverage the infinite-server queuing network model to study the performance of ISO. Let $\mathcal{M}_j$ be the (random) set of nodes viewing channel j. For ISO, the probability of system-wide universal streaming is

$$PS = P(v_j + \sum_{i \in \mathcal{M}_j} u_i \geq M_j r_j,\ j = 1, \ldots, J)$$

where $\mathcal{M}_j$ is a random set and $M_j = |\mathcal{M}_j|$. We can easily calculate this probability using Monte Carlo methods and importance sampling [26], that is, we repeatedly generate samples $\{\mathcal{M}_1, \ldots, \mathcal{M}_J\}$, evaluate the event in the above probability (as 0 or 1) and take averages. Note that Monte Carlo methods, which exploit the underlying queuing model, are significantly more efficient than discrete-event simulation.

The expected number of happy channels is

$$HC = \sum_{j=1}^{J} P(v_j + \sum_{i \in \mathcal{M}_j} u_i \geq M_j r_j)$$

To calculate HC, we need to calculate the probability of universal streaming in each channel. Suppose there are two classes of peers: n^l peers with low upload rate u^l; and n^h peers with high upload rates u^h. For a given channel j, the probability of universal streaming is given by

$$\begin{aligned}
PU_j &= P(v_j + \sum_{i \in \mathcal{M}_j} u_i \geq M_j r_j) \\
&= P(v_j + u^h M_j^h + u^l M_j^l \geq (M_j^h + M_j^l) r_j) \\
&= \sum_{M_j^h=0}^{n^h} \sum_{M_j^l=0}^{n^l} P(v_j + u^h m_j^h + u^l m_j^l \geq (m_j^h + m_j^l) r_j \\
&\quad |M_j^h = m_j^h, M_j^l = m_j^l) P(M_j^h = m_j^h) P(M_j^l = m_j^l) \\
&= \sum_{M_j^h=0}^{n^h} \sum_{M_j^l=0}^{n^l} 1(v_j + u^h m_j^h + u^l m_j^l \geq (m_j^h + m_j^l) r_j) \\
&\quad P(M_j^h = m_j^h) P(M_j^l = m_j^l)
\end{aligned}$$

Where $P(M_j^h = m_j^h) = \binom{n^h}{m_j^h} \rho_j^{m_j^h} (1-\rho_j)^{n^h - m_j^h}$ and $P(M_j^l = m_j^l) = \binom{n^l}{m_j^l} \rho_j^{m_j^l} (1-\rho_j)^{n^l - m_j^l}$. From these equations, it is straightforward to derive efficient convolution algorithms to calculate HC, the expected number of happy channels [23].

A. *Asymptotics*

For ISO, we consider a heterogeneous system with two types of peers: peers with low upload rate u^l and peers with high upload rates u^h. Suppose that the fraction of low upload rate peers is fixed and denoted by f. Thus the average upload rate of the peers is $u^l f + u^h(1-f)$. In the case the critical parameter becomes:

$$\alpha = \frac{\max_j r_j}{u^l f + u^h (1-f)}$$

It can be shown that the probability of system-wide universal streaming goes to 1 if $\alpha \leq 1$ and goes to 0 otherwise.

For example, consider a heterogeneous system with $u^h = 4r, u^l = 2r, f = 0.5$. Suppose there are two channels in the system: channel 1 with streaming rate $r_1 = 5r$ and popularity $\rho_1 = 0.2$, and channel 2 with streaming rate $r_1 = r$ and popularity $\rho_2 = 0.8$. For ISO design, PS goes to zero as its critical value $\alpha > 1$. However, under VUD design with substreaming, by allocating all high-bandwidth peers to distribute channel 1 and all low-bandwidth peers to distribute channel 2, the resource index in each channel is larger than 1 and PS goes to 1 asymptotically.

V. Peer Churn and Channel Churn

Up to now we have assumed that there is only channel churn and no peer churn (that is, the number of peers in the system is fixed). Such a model is appropriate when channel churn occurs on a much faster time scale than peer churn; or when the peer devices are always on, as with set-top boxes. In this section we briefly indicate how the model and results can be extended to allow for peer churn as well as channel churn.

To include peer churn in the model, we will use an open network of infinite-server queues. In such an open system, peers join and leave the system freely. Let γ_i be the exogenous arrival rate for channel i, that is, the rate at which peers join the system (peer churn) at channel i. After viewing channel i, a peer leaves the system (peer churn) with probability p_{i0}, or switches to channel j with probability p_{ij}. One can treat J channels as an open Jackson network of J infinite-server queues (again with arbitrary sojourn time distributions). Let $\lambda = (\lambda_1, \cdots, \lambda_J)$ be the effective arrival rate vector for all channels, then

$$\lambda_i = \gamma_i + \sum_{j=1}^{J} \lambda_j p_{ji},$$

or in vector-matrix form

$$\lambda = \gamma + \lambda \mathbf{P},$$

where P is the $J \times J$ channel switching matrix with and $\sum_{j=1}^{J} p_{ij} = 1 - p_{i0}$. Let $\rho_j = \lambda_j / \mu_j$, and ρ_j is the expected number of viewers for channel j. From the theory of Jackson networks [17], we have

Lemma 5.1: For the multi-channel system with peer and channel churn, we have

$$P(M_1 = m_1, \ldots, M_J = m_J) = \prod_{j=1}^{J} \frac{\rho_j^{m_j} e^{-\rho_j}}{m_j!}$$

The marginal distribution for individual channels is Poisson with mean ρ_j,

$$P(M_j = m_j) = \frac{\rho_j^{m_j} e^{-\rho_j}}{m_j!}$$

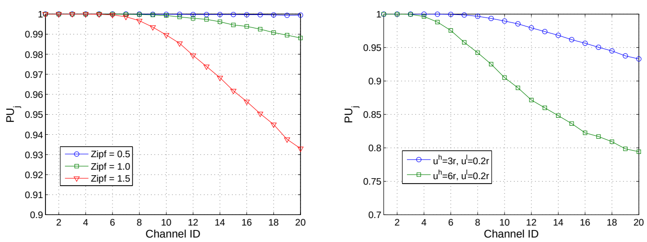
This result, analogous to Lemma 2.1, can be used to develop a parallel theory for P2P streaming systems with peer churn; in particular, the methodology developed in this paper we can be used to: derive efficient algorithms for calculating the probability of universal streaming and the average number of happy channels for both VUD and ISO designs; prove that all substreams should have the same rate and should have the same resource index; and derive asymptotic results for large systems, leading to simple heuristics for dimensioning the groups in VUD.

VI. Numerical Results

We develop a set of Matlab codes to calculate the performance metrics PU_j, PS of multi-channel streaming systems numerically based on the equations provided in this paper. In the default settings, there are 1,800 peers in the system, 20 channels, and each channel has the same rate r. The system includes two classes of peers: $f = 50\%$ low-bandwidth peers with $u^l = 0.2r$ and $1 - f = 50\%$ high-bandwidth peers with $u^h = 3r$. The upload bandwidth setting is reasonable according to the measurement results in [6]. For example, when the streaming rate r is 500kpbs, the residential peer may only have a low upload rate of 100kbps (i.e., $0.2r$), while the campus peer may have a high upload rate of 1.5Mbps (i.e. $3r$).

The channel popularity (determined by the ρ_j's) follows a Zipf distribution, with the default Zipf parameter as 1. For VUD design, we use the heuristic derived from asymptotic analysis (i.e., $K_j^l = \rho_j f$, $K_j^h = \rho_j(1-f)$ when all the channels have the same rate).

A. ISO Design



(a) ISO: varying channel popularity (b) ISO: varying upload bandwidth heterogeneity

Fig. 1. ISO design. (a) varying channel popularity (i.e., Zipf parameter). The probability of universal streaming in small channels is lower than large channels; (b) varying upload bandwidth heterogeneity, Zipf = 1.5. When increasing node heterogeneity, small channel problem becomes more obvious.

Figure 1 shows the value of PU_j, the probability of universal streaming for channel j, for the 20 channels under different popularity distributions. The channels are ranked based on their popularity in descending order. We see from Figure 1(a) that less popular channels can suffer with ISO, particularly for larger values of the Zipf parameter (when the average number of peers in the less popular channels becomes smaller). We see from Figure 1(b) that upload bandwidth heterogeneity further deteriorates the probability of universal streaming in small channels. Note that, in the two bandwidth settings, the average upload bandwidth $\bar{u}$ is made to be equal by adjusting the fraction of low-bandwidth peers f. Under the same $\bar{u}$, when the nodes become more heterogeneous, the number of high-bandwidth peers will decrease and the departure of high-bandwidth peers will impact more significantly on less popular channels.

Figure 2 shows the probability of system-wide universal streaming PS under different critical values. We vary the critical value by adjusting the streaming rate in the system. For a system with both 900 peers and 1800 peers, as we expect from the asymptotic theory, the performance is good when $\alpha < 1$ and bad when $\alpha > 1$. This figure also points out that the system with smaller population is more sensitive to the critical value.

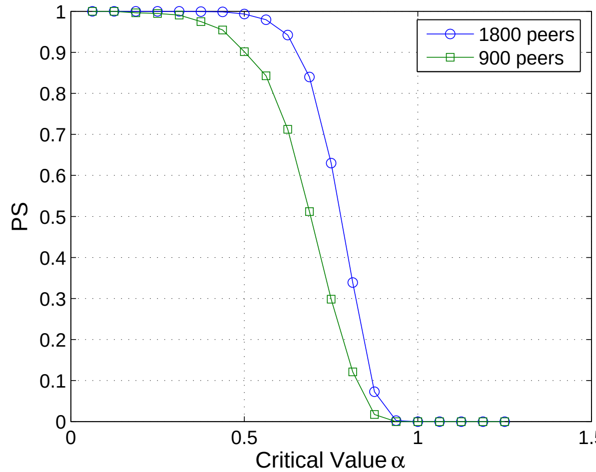
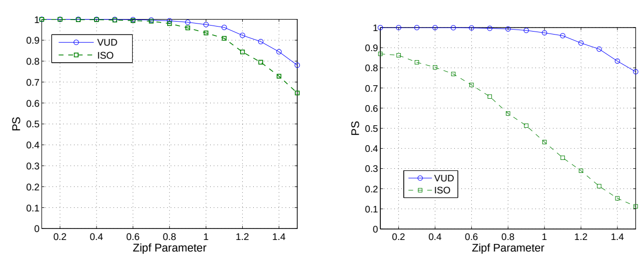


Fig. 2. ISO design: impact of critical value. When $\alpha > 1$, the probability of system-wide universal streaming drops to zero.

B. Comparison between ISO and VUD

We now compare the performance of ISO and VUD design, using the model of this paper.

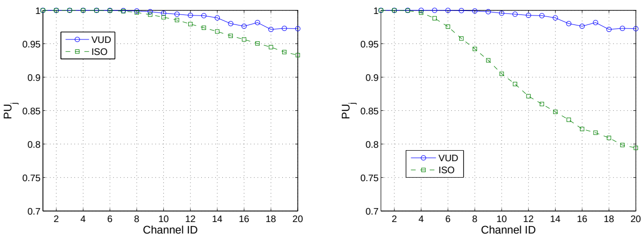
Figure 3(a) shows that VUD design can achieve higher probability of system-wide universal streaming, PS, under different channel popularity. For ISO design, as we have already seen that it suffers from small-channel problem, its probability of system-wide universal streaming decreases when the number of small channels increase (i.e., Zipf parameter increases). On the contrary, VUD design can explicitly provision bandwidth among channels, and increase the probability of universal streaming in small channels. Figure 3(b) further shows that, when increasing the degree of node heterogeneity (i.e., setting $u^h = 6r$, $u^l = 0.2r$) , the advantage of VUD becomes even more pronounced.



(a) $u^h = 3r, u^l = 0.2r$ (b) $u^h = 6r, u^l = 0.2r$

Fig. 3. ISO vs. VUD. Comparison of the probability of system-wide universal streaming PS: (a) $u^h = 3r$, $u^l = 0.2r$. VUD can achieve higher PS. (b) $u^h = 6r$, $u^l = 0.2r$. When increasing the degree of node heterogeneity, VUD has more advantages over ISO.

In Figure 4, we compare the probability of universal streaming in each channel, PU_j, between ISO and VUD. As VUD uses a heuristic, the VUD curve is not smooth and exhibits some fluctuation. In both bandwidth settings, VUD clearly has higher PU_j in small channels. Anyway, the heuristic used here is quite simple. In the later section, we will introduce a more refined heuristic to further increase the probability of universal streaming in small channels, without sacrificing the performance in large channels.



(a) $u^h = 3r, u^l = 0.2r$ (b) $u^h = 6r, u^l = 0.2r$

Fig. 4. ISO vs. VUD. Comparison of the probability of universal streaming in each channel PU_j. Zipf = 1.5. (a) $u^h = 3r$, $u^l = 0.2r$. VUD has higher PU_j in small channels; (b) $u^h = 6r$, $u^l = 0.2r$. When nodes are more heterogenous, VUD has more advantages over ISO.

In some existing P2P streaming systems, the channels may have different streaming rates (e.g., HDTV channels). In Figure 5, we also compare the performance of ISO and VUD when there are heterogeneous streaming rates in the system. In our setting, there are 4 streaming rates among 20 channels: channels 1-5 with $2r$; channels 5-10 with $1.5r$; channel 11-15 with r; channels 16-20 with $0.5r$.

Figure 5 shows that VUD design also achieves higher PU_j for high-rate channels than ISO design. The reason is that VUD design can allocate abundant bandwidth in other low-rate channels to high-rate channels and improve the universal-streaming probability, PU_j, in high-rate channels accordingly. However, in isolated-channel design, it is infeasible to balance the upload bandwidth resource among channels.

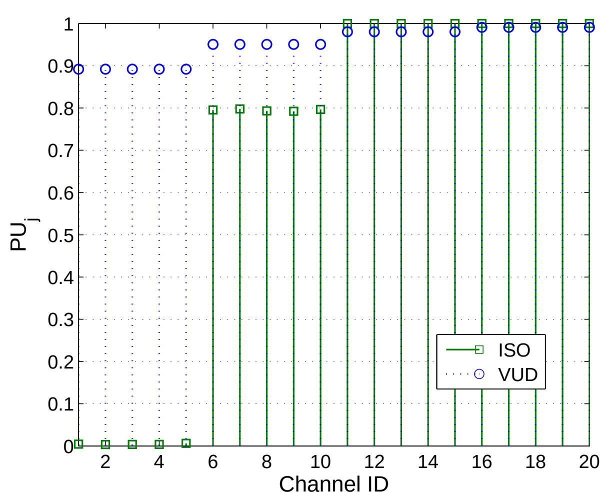


Fig. 5. ISO vs. VUD: heterogeneous streaming rates among channels. There are 4 streaming rates: channels 1-5 with $2r$; channels 5-10 with $1.5r$; channel 11-15 with r; channels 16-20 with $0.5r$. VUD also achieves higher PU_j for high-rate channels.

VII. Refined Heuristic for VUD Substreaming

Although the simple heuristic derived from our asymptotic analysis can perform better than isolated-channel design, the small channels still cannot achieve exactly the same probability of universal streaming as that of large channels.

In this section, we propose another more refined heuristic. The basic idea is to level the probability of universal streaming for all substreams, i.e., we need to guarantee

$$P(\sigma_j^s(M_j) \geq 1) = C, \forall j = 1, \ldots, J; \forall s = 1, \ldots, S_j$$

We still consider a heterogeneous system with two classes of distribution peers in the system. Their upload capacities are u^h and u^l respectively. Among all the distribution peers, let the total number of high-bandwidth peers be n^h, and the total number of low-bandwidth peers be n^l.

Let $n_j^{s(h)}$ and $n_j^{s(l)}$ be the number of high-bandwidth peers and low-bandwidth peers allocated to the substream s of channel j respectively. We have $\sum_{j=1}^J \sum_{s=1}^{S_j} n_j^{s(h)} = n^h$ and $\sum_{j=1}^J \sum_{s=1}^{S_j} n_j^{s(l)} = n^l$.

Since $E[M_j] = n\rho_j$ and $Var[M_j] = n\rho_j(1-\rho_j)$, let

$$M'_j \triangleq \frac{M_j - n\rho_j}{\sqrt{n\rho_j(1-\rho_j)}},$$

following the central limit theorem, M'_j converges in probability to a standard normal distribution $N(0,1)$, $\forall j$. C is a constant value.

$$\begin{aligned} & P(\sigma_j^s(M_j) \geq 1) = C \\ \Rightarrow\ & P\left(M_j \leq \frac{v_j^s}{r_j^s} + n_j^{s(h)}\frac{u^h - r_j^s}{r_j^s} + n_j^{s(l)}\frac{u^l - r_j^s}{r_j^s}\right) = C \\ \Rightarrow\ & P\left(M'_j \leq \frac{\frac{v_j^s}{r_j^s} + n_j^{s(h)}\frac{u^h - r_j^s}{r_j^s} + n_j^{s(l)}\frac{u^l - r_j^s}{r_j^s} - n\rho_j}{\sqrt{n\rho_j(1-\rho_j)}}\right) = C \end{aligned}$$

Since all M'_j follows $N(0,1)$, we have

$$\frac{\frac{v_j^s}{r_j^s} + n_j^{s(h)}\frac{u^h - r_j^s}{r_j^s} + n_j^{s(l)}\frac{u^l - r_j^s}{r_j^s} - n\rho_j}{\sqrt{n\rho_j(1-\rho_j)}} = C_1$$

with $\sum_{j=1}^J \sum_{s=1}^{S_j} n_j^{s(h)} = n^h$ and $\sum_{j=1}^J \sum_{s=1}^{S_j} n_j^{s(l)} = n^l$.

In this case, we have $2 + \sum_{j=1}^J S_j$ equations with $1 + 2\sum_{j=1}^J S_j$ variables. There are multiple solutions that satisfy the above equations.

Considering a special where all the channels have the same streaming rate (i.e., $r_j = r$) and the same number of substreams (i.e., $S_j = S$). We can get the closed-form value of C_1 as:

$$C_1 = \frac{\sum_{j=1}^J v_j + (u^h - \frac{r}{S})n^h + (u^l - \frac{r}{S})n^l - rn}{r\sum_{j=1}^J \sqrt{n\rho_j(1-\rho_j)}}$$

and one feasible solution is

$$n_j^{s(h)} = \frac{n^h \frac{r}{S}(n\rho_j + C_1\sqrt{n\rho_j(1-\rho_j)} - \frac{v_j^s}{r/S})}{n^h(u^h - \frac{r}{S}) + n^l(u^l - \frac{r}{S})}$$

$$n_j^{s(l)} = \frac{n^l \frac{r}{S}(n\rho_j + C_1\sqrt{n\rho_j(1-\rho_j)} - \frac{v_j^s}{r/S})}{n^h(u^h - \frac{r}{S}) + n^l(u^l - \frac{r}{S})}$$

For very small channels, say $E[M_j] < 5$, M_j' may not be well approximated by $N(0,1)$. Fortunately, it is not expensive to do over provisioning for them.

Figure 6 compares the refined heuristic (denoted by "*refined VUD*") with the simple heuristic (denoted by "*simple VUD*") used in Section VI and ISO design. It is found that, compared with the simple VUD, the refined VUD provisions more bandwidth for small channels and does equalize PU_j among different channels. The small channels have the same probability of universal streaming as large channels.

For the implementation, VUD design can be implemented in either a centralized or decentralized manner (see our technical report [9] for details).

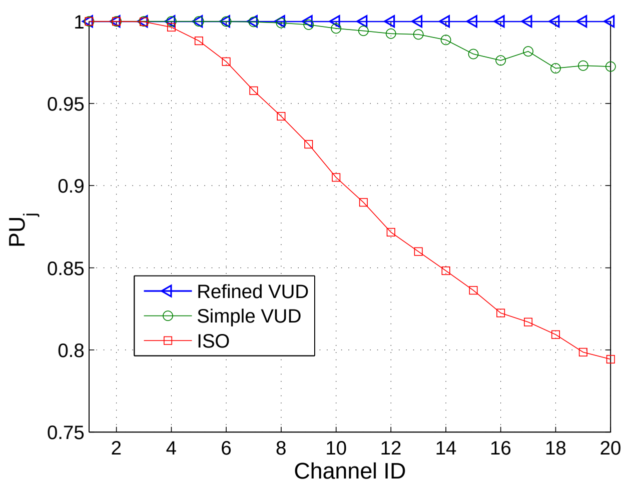


Fig. 6. Comparison of Refined VUD, Simple VUD and ISO. Probability of universal streaming in each channel PU_j. Zipf = 1.5, $u^h = 6r$, $u^l = 0.2r$. Refined VUD can achieve higher PU_j in small channels.

VIII. Conclusion

In this paper, we studied the performance of multi-channel P2P live streaming systems using close and open queueing network models. Our models capture the essential aspects of multi-channel video systems, including peer channel switching, peer churn, peer bandwidth heterogeneity, and Zipf-like channel popularity. The analytic models allow us to identify the key factors determining the performance of ISO and VUD designs. We analytically showed that VUD can provide significantly better performance than ISO in heterogeneous P2P streaming systems. We also derived efficient VUD swarm provisioning rules.

Our models can be further extended to capture heterogeneous peer channel switching patterns. With scalable video coding, peers who download different subsets of sub-streams of channel perceive different video quality. We will study new substream dimensioning algorithms to maximize the overall video quality on all peers. On the practice side, we will apply the obtained theoretical results to our ongoing VUD system development effort [9]. We will use the VUD swarm provisioning rules to assign peers to distribution swarms in real VUD systems. The algorithms to calculate the critical performance measures will be used to guide peer admission control when VUD system operates in resource-deficit regions.

References

[1] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "DONet/CoolStreaming: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming," in *Proc. of IEEE INFOCOM*, Mar. 2005.
[2] "PPLive Homepage," http://www.pplive.com.
[3] "PPStream Homepage," http://www.pps.tv.
[4] "UUSee Homepage," http://www.uusee.com/.
[5] "Sopcast Homepage," http://www.sopcast.com/ .
[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*, 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*, December 2007.
[8] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft, "On Next-Generation Telco-Managed P2P TV Architectures," in *Proc. of IPTPS*, 2008.
[9] D. Wu, C. Liang, Y. Liu, and K. Ross, "View-Upload Decoupling: A Redesign of Multi-Channel P2P Video Systems," in *Technical Report, Polytechnic Institutue of New York University*, 2008, http://eeweb.poly.edu/faculty/yongliu/docs/VUD1.pdf.
[10] Y. Zhou, D. Chiu, and J. C. Lui., "A Simple Model for Analysis and Design of P2P Streaming Algorithms," in *Proc. of IEEE ICNP*, 2007.
[11] T. Small, B. Liang, and B. Li, "Scaling Laws and Tradeoffs in Peer-to-Peer live multimedia streaming," in *Proc. of ACM Multimedia*, 2006, pp. 539–548.
[12] R. Kumar, Y. Liu, and K. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," in *Proc. of IEEE INFOCOM*, 2007.
[13] L. Massoulie, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized decentralized broadcasting algorithms," in *Proc. of IEEE INFOCOM*, 2007.
[14] T. Bonald, L. Massoulie, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: optimal performance trade-offs," in *Proc. of ACM SIGMETRICS*, 2008.
[15] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, "Performance Bounds for Peer-Assisted Live Streaming," in *Proc. of ACM SIGMETRICS*, 2008.
[16] C. Wu, B. Li, and S. Zhao, "Multi-channel Live P2P Streaming: Refocusing on Servers," in *Proc. of IEEE INFOCOM*, Apr. 2008.
[17] F. P. Kelly, *Reversibility and Stochastic Networks*. Wiley, Chichester, 1979.
[18] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth multicast in cooperative environments," in *Proc. of ACM SOSP*, 2003.
[19] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat, "Bullet: High Bandwidth Data Dissemination using an Overlay Mesh," in *Proc. of ACM SOSP*, 2003.
[20] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. E. Mohr, "Chainsaw: Eliminating Trees from Overlay Multicast," in *Proc. of IPTPS*, 2005.
[21] N. Magharei and R. Rejaie, "PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming," in *Proc. of IEEE INFOCOM*, 2007.
[22] M. Zhang, L. Zhao, Y. Tang, J.-G. Luo, and S. Yang, "A Peer-to-Peer Network for Streaming Multicast Through the Internet," in *Proc. of ACM Multimedia*, 2005.
[23] K. Ross, *Multiservice Loss Networks for Broadband Telecommunications Networks*. Springer-Verlag, 1995.
[24] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
[25] F. P. Kelly, "Loss networks," *Annals of Applied Probability*, vol. 1:319-378, 1991.
[26] K. Ross, D. Tsang, and J. Wang, "Monte Carlo Summation and Integration Applied to Multichain Queueing Networks," *Journal for the Association of Computing Machinery*, vol. 41, pp. 1110–1135, 1994.