# Impacts of Peer Churn on P2P Streaming Networks*

Xiaohan Kang[1], Juan José Jaramillo[2] and Lei Ying[1]

***Abstract*— Peer-to-peer (P2P) technology has been broadly adopted in live media streaming in recent years. In this paper, we consider a P2P streaming network where a server generates content chunks, and transmits each chunk to a randomly selected peer. Peers then exchange chunks among themselves according to some chunk selection policy. While the performance of different chunk selection policies has been intensively analyzed assuming no peer arrival or departure, the impact of peer churn on P2P live streaming has not been well understood yet. We show that unlike in the static network scenario, larger buffer size does not necessarily result in higher playout probability in the presence of peer churn. We further analyze the relation between the buffer size, playout probability and peer churn under different chunk selection policies and characterize the impact of peer churn on the performance of these chunk selection policies via both theoretical analysis and simulations.**

## I. INTRODUCTION

Internet video became the largest traffic type in 2010, accounting for $40\%$ of all global consumer traffic, and it is expected to reach $62\%$ by the end of 2015 [1]. Thus, the importance of studying efficient and scalable methods to deliver real-time media is paramount. Because of its inherent scalability, P2P technology has been extremely successful in delivering streaming videos to millions of users.

One approach of P2P streaming is to use multicast networking at the IP or higher layer, where data is pushed along pre-constructed multicast trees [2], [3], [4]. This approach, however, requires significant infrastructural overheads, particularly in networks with peer churn because the multicast trees need to be reconfigured every time a peer arrives or departs [5]. An alternative approach is unstructured P2P streaming [6], [7], [8]. In these networks, the streaming content is divided into small chunks that are disseminated to a small number of peers, and the content is then duplicated within peers in a gossip-like fashion without the intervention of any centralized servers, so unstructured P2P streaming is robust under peer churn. Because of its robustness, unstructured P2P streaming has become the dominant architecture of large-scale P2P streaming networks, such as CoolStreaming [6], PPTV [9] and PPStream [10]. In an unstructured streaming network, each peer maintains a playout buffer. The chunk at the tail of the buffer is the one to be played out in the current time slot. At the end of each time slot, all the chunks are moved one position closer to the tail. In each time slot, every peer randomly contacts another peer to download a chunk missing in the buffer so that when the chunk needs to be played out, it is present in the buffer with high probability. The probability that any peer is able to obtain a specific chunk by the time of playout is called playout probability. Among other metrics, the playout probability and the buffer size are essential to evaluate the performance of a P2P live streaming network since the playout probability measures the continuity of the streaming experience, while the buffer size measures the delay of the playback of the live event. There are two main problems that need to be studied in P2P streaming protocol design: how to select a peer to contact [11] and how to select a chunk to download [12], [13], [14]. Among various chunk selection policies, the rarest-first policy, which tries to obtain the most recently generated chunk, is most popular, while the greedy policy, which attempts to download the chunk closest to playout, is an intuitive alternative. The combination of the two, called the hybrid policy, achieves order sense optimality in minimizing the buffer size when peers are static [15]. In this paper we study the problem of chunk selection, and we focus on the three selection policies described above.

The main advantage of unstructured P2P streaming over structured P2P streaming is that the unstructured approach has robust performance under peer churn. However, existing works of unstructured P2P streaming (see [15], [16] and references within) almost exclusively ignore peer churn and assume no peer arrival/departure. This is mainly due to the difficulty of establishing a tractable model of P2P streaming networks under peer churn. To the best of our knowledge there have only been empirical studies of peer churn in P2P file sharing networks such as Gnutella or BitTorrent [17], models for structured P2P streaming as in [18], as opposed to unstructured networks, or via simulations as in [5].

In this paper we study peer churn with a constant arrival and departure rate (churn rate). We try to answer the following questions in the presence of peer churn: 1) *Does larger buffer guarantee higher playout probability?* 2) *Is there a limit of the playout probability one can achieve?* 3) *Do some chunk selection policies perform better than others in terms of streaming continuity and delay, and why?* We tackle these questions by fluid and discrete methods and simulations.

Our contributions are summarized as follows:

1) Contrary to the previous result for the scenario without peer churn, where increasing the buffer size always leads to higher playout probability, we derive theoretical upper and lower bounds of the buffer size for the rarest-first policy for a given target playout probability,

*Research supported by NSF CNS-0963807.

[1]X. Kang and L. Ying are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA {xkang,leiying}@iastate.edu

[2]J. J. Jaramillo is with the Department of Applied Math and Engineering, Universidad EAFIT, Medellín, Colombia jjaram93@eafit.edu.co

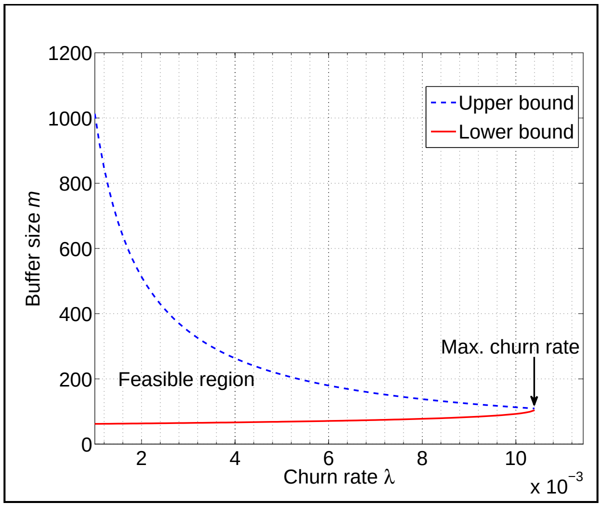



Fig. 1. Feasible region for the rarest-first policy to select a buffer of size $m$ with churn rate $\lambda$ to achieve a playout probability of $0.98$ in a P2P network with $1000$ peers

as depicted in Fig. 1. We see that increasing the buffer size does not necessarily lead to higher playout probability in the presence of peer churn.

2) We prove that the level of peer churn $\lambda$ in the system limits the maximum playout probability that can be achieved by any chunk selection policy to $1 - \lambda$.
3) We compare the performance of different chunk selection policies via simulations, and show that the system suffers from the early departures of rare chunks and contacting newcomers. We also show that the rarest-first policy is a robust alternative of the hybrid policy.

## II. Model Description

We consider a P2P content streaming network with one server and $N$ peers. We assume that all peers can communicate with each other and with the server, i.e., each peer may download chunks from any other peers or the server. The streaming content is divided into chunks of equal time duration. Time is slotted with each time slot equal to the chunk content duration and all peers are synchronized with the server. In the content streaming network, at every time slot, the server generates a new chunk of live data that needs to be distributed to all peers. We assume the server has limited capacity such that at every time slot the server can only contact one peer to distribute the newly generated chunk, so the cooperation of peers is necessary to distribute the chunk to all peers. We also assume the peers have limited download capacity, so each peer can only download a chunk of data per time slot, either from the server or from another peer. The upload capacity of the peers is assumed unlimited, so if a peer is contacted by multiple peers she can fulfill all the download requests. This simplifying capacity assumption has been used in [13], [15], [16].

Due to limited capacity constraints, the distribution of a specific chunk in the network takes more than one time slot. Thus, peers use buffers of size $m$ to delay playout to allow a given chunk to have sufficient time to be replicated, so that most of the peers will receive the chunk when it needs to be played out. We assume that the buffer positions are indexed by $i \in \{1, 2, \ldots, m\}$, and that the server distributes the newly generated chunk to position $1$ of the buffer, while the chunk in position $m$ is played out. Later after playout, the chunk at buffer position $i$ is shifted to buffer position $i+1$ for $1 \leq i \leq m-1$.

We assume that at every time slot the server randomly contacts one peer from a discrete uniform distribution to circulate the newly generated chunk, and that every peer, if not selected by the server, randomly contacts another peer from a discrete uniform distribution to download one chunk[1]. We also assume that the chunk distribution model is *two-sided pull-based* [19], that is, every peer gets to decide which chunk it is going to download from another peer, subject to chunk availability of both the target/contacted peer and the recipient/requesting peer.

We consider the case where peers arrive and depart from the network. Although the flash crowd is a very typical and interesting peer churn phenomenon in P2P live streaming systems [20], [21], [22], we focus on the scenario when the peer population tends to be stable. We assume that in each time slot a fixed number of peers (randomly selected from all the peers according to a discrete uniform distribution) depart and are replaced by the same number of newcomers, so the total number of peers in the system remains constant. Then we denote the (normalized) churn rate of peers, the number of peer arrivals (departures) divided by the total number of peers in the network, by $\lambda \in (0, 1)$. We label the initial peers by indices $1, 2, \ldots, N$, and in each time slot the newcomers take over the indices of the departing peers, while the peers staying in the network have their indices unchanged. To help the reader understand our model, in Fig. 2 we illustrate the sequence of events we consider in a given time slot.

We let $X(t) \in M_{N\times m}(\{0,1\})$ be an $N$-by-$m$ matrix with binary entries denoting the availability of the chunks of all peers by setting $X_{ij}(t) = 1$ if Peer $i$ has the chunk in the buffer position $j$ at the beginning of the time slot $t$, and $X_{ij}(t) = 0$ otherwise, where $1 \leq i \leq N$ and $1 \leq j \leq m$. Then after specifying the chunk selection policy $\mu$, $X(t)$ is a Markov chain with the finite state space $M_{N\times m}(\{0,1\})$. Although $(X(t), t \geq 0)$ is itself a Markov chain, peers with different ages generally have different distributions over the chunks. We thus consider another Markov chain $Y(t) = (X(t), A(t))$, where $A(t) = (A_i(t), 1 \leq i \leq N)$ and $A_i(t)$ is the age of Peer $i$ denoting the number of time slots Peer $i$ has been in the system. Then $(Y(t), t \geq 0)$ is a Markov chain with an infinite state space $\mathcal{S} = M_{N\times m}(\{0,1\}) \times \mathbb{Z}_+^N$, where $\mathbb{Z}_+$ denotes the positive integers.

We assume the system is in the steady state, that is, the Markov chain has converged to the unique stationary distribution, and we let $\pi\colon \mathcal{S} \to \mathbb{R}$ be the pmf of the stationary distribution. We further assume that peers with ages greater than or equal to $K$ have the same steady state

[1]Throughout this paper we assume random peer selection policy from a discrete uniform distribution unless otherwise specified.

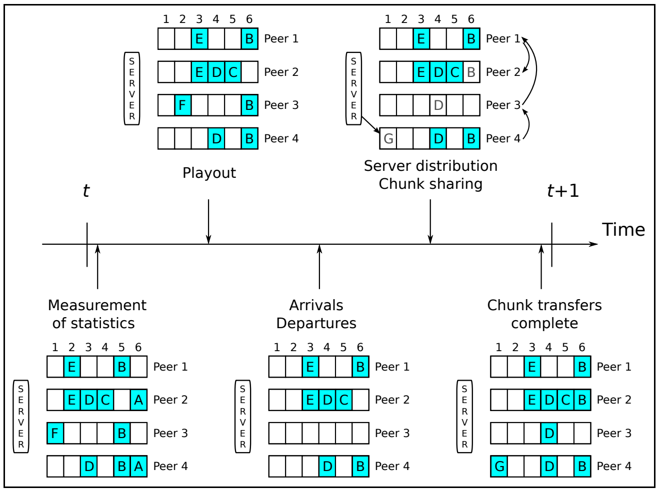



Fig. 2. Sequence of events in a given time slot. In this example, the peer population is $N = 4$ and the buffer size for each peer is $m = 6$. The buffer position 1 is the most recently generated chunk and the buffer position 6 is the chunk to playout. The chunk availability matrix $X(t)$ is given at the beginning of the time slot $t$. Then Chunk A at position 6 is played out and the buffers shift forward. After the playout Peer 3 departs and is replaced by a newcomer. Then the server chooses Peer 4 to distribute the new chunk G and every peer except Peer 4 selects a peer to request chunks. All chunk transfers complete before the end of this time slot.

distribution over their chunks, for some fixed positive integer $K$. Then in each time slot $t$ we can classify all the peers into $K$ classes by their ages, where Peer $i$ is in Class $j \in \{1, 2, \ldots, K-1\}$ if $A_i(t) = j$ and is in Class $K$ if $A_i(t) \geq K$ for $1 \leq i \leq N$. We first define

$$\begin{aligned} p_j^{(l)}(i,t) &= \Pr(X_{li}(t) = 1 \mid A_l(t) = j) \\ &= \frac{\sum_{s\in\mathcal{S}:\ a_l(s)=j, x_{li}(s)=1} \pi(s)}{\sum_{s\in\mathcal{S}:\ a_l(s)=j} \pi(s)} \end{aligned}$$

for $1 \leq j \leq K-1$, $1 \leq i \leq m$, $1 \leq l \leq N$ and $t \geq 0$, and

$$\begin{aligned} p_K^{(l)}(i,t) &= \Pr(X_{li}(t) = 1 \mid A_l(t) \geq K) \\ &= \frac{\sum_{s\in\mathcal{S}:\ a_l(s)\geq K, x_{li}(s)=1} \pi(s)}{\sum_{s\in\mathcal{S}:\ a_l(s)\geq K} \pi(s)} \end{aligned}$$

for $1 \leq i \leq m$, $1 \leq l \leq N$ and $t \geq 0$, where $s = (x(s), a(s)) \in \mathcal{S}$. We then define the *buffer occupancy probability* of the buffer position $i$ for a Class $j$ peer by

$$p_j(i) = p_j^{(l)}(i,t)$$

for $1 \leq j \leq K$, $1 \leq i \leq m$, $1 \leq l \leq N$ and $t \geq 0$. This definition is well-defined due to the steady state assumption and the symmetry of peers in the steady state.

We assume that in the steady state system the buffer occupancy probabilities are independent across peers and buffer positions, then for any chunk availability vector $\sigma \in \{0,1\}^m$ and any peer in Class $j \in \{1, 2, \ldots, K\}$, the pmf of the chunk availability distribution is

$$\nu_j(\sigma) = \prod_{i:\ \sigma_i=1} p_j(i) \prod_{i:\ \sigma_i=0} (1 - p_j(i)).$$

It must be noted that in our model we call $p_K(m)$ the playout probability, which is the buffer occupancy probability of the buffer $m$ for a Class $K$ peer.

Since the peer population in the P2P streaming network is large and the churn rate $\lambda$ is usually much smaller than 1, the time for a peer to reach steady-state would be small compared to the expected length of time it stays in the system, then we can approximate the system by assuming that departures only happen in Class $K$. Hence we have a fraction $\lambda$ of the peers in Class $j$ for $1 \leq j \leq K-1$, and a fraction $1 - (K-1)\lambda$ of the peers in Class $K$. To have enough peers for departures in Class $K$, we must have

$$K \leq 1/\lambda. \tag{1}$$

Furthermore, we use a mean field analysis approximation: Since we do not know which peer will be contacted during the chunk sharing stage, we assume that the contacted peer has a buffer occupancy probability given by

$$\bar{p}(i) = (1 - K\lambda) p_K(i) + \lambda \sum_{j=1}^{K-1} p_j(i)$$

for $1 \leq i \leq m$, and $\bar{p}(i)$ is independent across $i$. Note from Fig. 2 that chunk sharing occurs after peer departure, so the fraction of peers in class $K$ that can be contacted is $1 - K\lambda$ instead of $1 - (K-1)\lambda$. The assumptions in the last four paragraphs help to simplify the equations we present next.

Denote by $s_j(i)$ the probability that a peer in Class $j$ downloads the chunk in position $i$ using a given chunk selection policy, provided that the target has chunk $i$ but the recipient does not. By the independence of the occupancy probability across buffer positions, $s_j(i)$ is equal to the probability that no other chunk of the Class $j$ peer is preferred by the chunk selection policy. Given this model and assumptions, we can now write the network evolution equations at the beginning of any time slot:

$$p_{j+1}(i+1) = p_j(i) + s_j(i)\bar{p}(i)(1 - p_j(i)) \tag{2}$$

for $1 \leq i \leq m-1$ and $0 \leq j \leq K-1$, and

$$p_K(i+1) = p_K(i) + s_K(i)\bar{p}(i)(1 - p_K(i)) \tag{3}$$

for $1 \leq i \leq m-1$, with boundary conditions

$$p_j(1) = 1/N \quad \text{for } 1 \leq j \leq K, \tag{4}$$

and for convenience we define $p_0(i) = 0$ for $1 \leq i \leq m$. One can think of $p_0(i)$ as the occupancy probability at buffer position $i$ of newcomers.

Some comments about the network evolution equations are in order. In (2) and (3) we have that the buffer $i+1$ is occupied if either the peer had the chunk in buffer $i$ in the previous time slot, or it downloaded it via chunk sharing. To download a chunk we need to have (i) the peer does not have the chunk, (ii) the contacted peer has the chunk, and (iii) the peer selects that chunk for download. By aforementioned assumptions we approximate this probability by $(1 - p_j(i))\bar{p}(i)s_j(i)$. It also must be noted that the probability that a peer is contacted by the server is $1/N$, which explains the boundary condition (4). Note that we have assumed unlimited upload capacity of peers, global knowledge neighborhood, invariant peer population and that departures only happen to peers of age at least $K$. The above assumptions help to make the model tractable. Via simulations, we will validate that the results derived based on these assumptions also hold in realistic networks.

## III. THE RAREST-FIRST POLICY

In this section, we study the impact of peer churn on a widely-used chunk selection policy — the rarest-first policy. Note that we assume random peer selection policy throughout this section.

The rarest-first policy makes peers download the chunk in the lowest buffer position (farthest from playout) that the contacted peer has and the requesting peer does not have. The idea is to first duplicate the chunks that have been in the system for the shortest period of time and therefore are likely to have the least number of copies. It must be noticed that for the rarest-first policy to download a chunk in buffer position $i$ it must be the case that the peer did not get contacted by the server and that the peer could not download another chunk in a lower buffer position; thus, the probability that a peer in class $j$ downloads the chunk in position $i$ is given by

$$s_j(i) = (1 - 1/N)\prod_{l=1}^{i-1} \left(p_j(l) + (1 - p_j(l))\,(1 - \bar{p}(l))\right) \quad (5)$$

for $2 \le i \le m - 1$, $0 \le j \le K$, with boundary conditions

$$s_j(1) = 1 - 1/N \quad \text{for } 0 \le j \le K.$$

To understand (5), we have that a peer does not get contacted by the server with probability $1 - 1/N$, and the chunk in buffer position $1 \le l \le i - 1$ does not get downloaded if the contacting peer already has the chunk or if both peers do not have a copy, which happens with probability $p_j(l) + (1 - p_j(l))\,(1 - \bar{p}(l))$.

Although the expression in (5) is complex, it can be further simplified based on an interesting observation we have from the simulations. Fig. 3 shows the occupancy probabilities $p_j(i)$ for a system with random peer churn such that the average number of peers is 1000 and the buffer size is 50. [2] It can be observed that the probability $p_j(i)$ is identical to $p_K(i)$ up to a threshold buffer position $i$ such that $i \le T_j$, after which the probability sharply decreases to 0. We even observed that there exists a constant $c$ such that $T_j \approx j + c$. In other words, $p_j(i) = p_K(i)$ for $i \le j + c$. From this approximation we get the following result.

*Lemma 1:* Assuming that for the rarest-first policy the buffer occupancy probabilities can be approximated by $p_j(i) = p_K(i)I_{\{i \le j+c\}}$ for some non-negative constant $c$, where $I_{\{\cdot\}}$ is an indictor function, we have that

$$s_j(i) = \begin{cases} 1 - p_K(i) & \text{if } i \le j + c + 1 \\ 1 - p_K(j + c + 1) & \text{if } i > j + c + 1 \end{cases}. \qquad \diamond$$

The proof of Lemma 1 is omitted due to the page limit and can be found in [23]. This lemma gives a much simpler relation between $s_j(i)$'s and $p_j(i)$'s for the rarest-first policy, which is vital in the current model.

[2] A detailed description of the simulation settings that we have used will be presented in Section V.

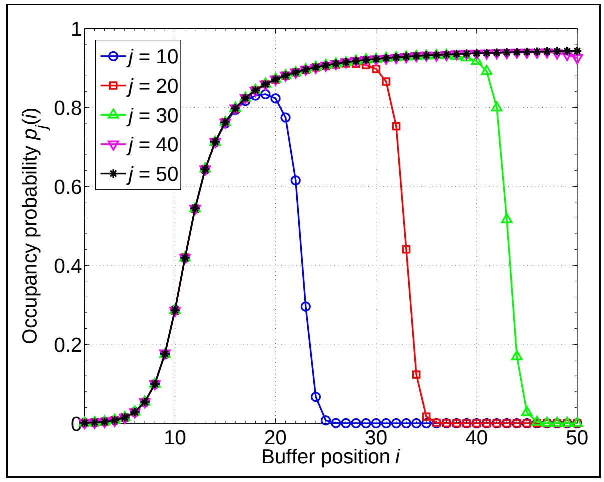



Fig. 3. Occupancy probability for the rarest-first policy

We are interested in the case when the playout probability for the last class, $p_K(m)$, is non-negligible. Thus, for consistency in the assumption $p_j(i) = p_K(i)I_{\{i \le j+c\}}$, it must be the case that

$$m < K + c \quad (6)$$

for the expression $p_K(i) = p_K(i)I_{\{i \le K+c\}}$ to make sense. Hence using (1), we have the following result.

*Theorem 1:* Assuming that the buffer occupancy probabilities for the rarest-first policy can be approximated by $p_j(i) = p_K(i)I_{\{i \le j+c\}}$ for some non-negative constant $c$, and that the playout probability of the peers in class $K$ is non-negligible, we have $m < 1/\lambda + c$. $\diamond$

The importance of this result is that it tells us that for a fixed amount of peer churn, there is a limit on the growth of the buffer size that we can have if we want to get a non-negligible playout probability.

Another implication of (6) and the approximation of $p_j(i)$ is that the expression of $\bar{p}(i)$ can be simplified too:

$$\bar{p}(i) = p_K(i)(1 - \max\{i - c, 1\}\lambda). \quad (7)$$

Using Lemma 1, (6), and (7) we then get

$$p_K(i+1) = p_K(i) + (1 - p_K(i))^2 p_K(i)(1 - \max\{i - c, 1\}\lambda). \quad (8)$$

From this set of equations we can then develop a fluid model of the rarest-first policy to bound the minimum buffer requirement and maximum allowable churn rate for a given target playout probability. To do that, we approximate the difference equations by the following differential equation

$$\frac{\mathrm{d}p_K(i)}{\mathrm{d}i} = (1 - p_K(i))^2 p_K(i)(1 - \max\{i - c, 1\}\lambda) \quad (9)$$

with initial condition $p_K(1) = 1/N$.

*Theorem 2:* From the analysis of the fluid model equation (9) we have

$$m \ge c + \frac{1}{\lambda} - \frac{1}{\lambda}\sqrt{-(2c-1)\lambda^2 - 2\left(f(p_K(m), N) - c\right)\lambda + 1}$$

and

$$\lambda \leq \frac{1}{\sqrt{(f(p_K(m),N)-c)^2+2c-1}+f(p_K(m),N)-c},$$

where

$$f(p_K(m),N) = \ln\frac{(N-1)p_K(m)}{1-p_K(m)} + \frac{1}{1-p_K(m)} - \frac{1}{N-1}.$$

⋄

The proof of Theorem 2 is omitted due to the page limit and can be found in [23]. The results of Theorem 2 are illustrated in Fig. 1. The parameter $c$ is estimated to be $13$ in the figure by the observation of Fig. 3.

*Remark 1:* As $p_K(m) \to 1$ and $N \to \infty$, Theorem 2 asymptotically becomes

$$m \geq \ln N + 1/(1-p_K(m))$$

and

$$\lambda \leq \left(\ln N + 1/(1-p_K(m))\right)^{-1}.$$

Note that the lower bound of the buffer size agrees with the result in [15]. ⋄

The significance of Theorem 1 and Theorem 2 lies in the following aspects: First, unlike in the case without peer churn where the buffer size $m$ has only a lower bound, there are now both a lower and an upper bound for $m$ in the presence of peer churn, given the playout probability requirement $p_K(m)$ and the peer population $N$. Second, with fixed $\lambda$, the lower bound of $m$ increases as $p_K(m)$ or $N$ increases, while the upper bound remain unchanged. Third, the higher the churn rate $\lambda$ is, the closer the two bounds are and thus, the stricter $m$ should be designed. Finally, for given $N$, the feasible region of $m$ vanishes as $\lambda$ becomes sufficiently large, or in other words, there is a maximum churn rate under which the target playout probability can be achieved.

## IV. UPPER BOUNDS OF THE PLAYOUT PROBABILITY

We now present two upper bounds of the playout probability for any two-sided pull-based chunk selection policy $\mu$ and any buffer size $m$. Although quantitatively the same, the first bound is due to peer departures, while the second one arises from peer arrivals. Note that we still use random peer selection policy in this section.

*Theorem 3:* Due to random departures, the playout probability for peers in Class $K$ under any chunk selection policy is upper-bounded by $p_K(m) \leq 1-\lambda$. ⋄

*Theorem 4:* Due to arrivals and random peer selection, the playout probability for peers in Class $K$ under any chunk selection policy is upper-bounded by $p_K(m) \leq 1-\lambda$. ⋄

The proofs of Theorems 3 and 4 are omitted due to the page limit and can be found in [23]. The idea of the proof for the bound due to departures in Theorem 3 is that the peer with the unique newly generated chunk leaves the network immediately with probability $\lambda$, before the chunk ever gets copied. The idea of the proof of the bound due to arrivals and random peer selection in Theorem 4 is that by the random peer selection policy any peer contacts a newcomer with probability $\lambda$ and cannot find any useful chunk. Compared to the result in [15], the maximum possible playout probability is now limited by $1-\lambda$ in the presence of peer churn, regardless of the buffer size.

To see the subtle difference between the two upper bounds, one can consider the following two systems. System A is described by the chunk availability matrix $X^{\mathrm{A}}(t)$ which is the same as $X(t)$ except that peers only contact those with at least one chunk they need, and System B is described by the chunk availability matrix $X^{\mathrm{B}}(t)$ which is the same as $X(t)$ in our model except that departures do not happen to peers with a unique chunk. In System A the first bound remains to make the playout performance suffer, while the second bound is eliminated by a smart (though complex) peer selection policy. System B is well-defined as long as $m \leq N(1-\lambda)$, since there can be at most one unique chunk at each buffer position and thus there are at most $m$ peers with unique chunks in the system. Then one can see that the first upper bound might not be true, while the second upper bound is still valid for the playout probability of System B since newcomers are still possible to be contacted. As a result, we conjecture that the two bounds are independent and a combined bound of roughly $1-2\lambda$ could be achieved with careful analysis.

## V. MODEL VALIDATION AND SIMULATION ANALYSIS

In this section we validate the accuracy of the theoretical model used for the upper and lower bounds for the rarest-first policy in Section III by simulations. We also perform simulation analysis on the rarest-first, the greedy and the hybrid policies to evaluate the playout performance.

The default settings are given as follows. We simulate a system where we have 2000 peers in the network. At the beginning of the simulations, 1000 peers are active and 1000 peers are inactive. During each time slot, a peer changes its state (from active to inactive or vice versa) with probability $1/200$. When a peer becomes inactive, it empties its buffer. So there are $1000$ peers in the network on average, and the average churn rate is $\bar{\lambda} = 1/200$. The size of the buffer for all peers is $m = 50$, and we run the simulations for $100,000$ time slots and ignore the first $50,000$ time slots so that the system is in steady state. We look for a positive integer $K$ in each simulation such that peers with age at least $K$ have roughly the same playout probability, and find that we can empirically estimate $K = m$ for the rarest-first, $K = 2m$ for the greedy, and $K = 5m$ for the hybrid. Note that we intentionally reused the letter $K$ since it is the simulation counterpart of the parameter $K$ defined in Section II. We call the peers with ages at least $K$ the steady class or Class $K$ peers. Hence, in the simulations we do not make the assumption that the peer population is always the same since the numbers of arrivals and departures are random. Moreover, as the simulations follow the original Markov chain $X(t)$ in Section II, the assumptions made for the theoretical analysis of the rarest-first policy (including the $c$ parameter approximation, the assumption that departures

TABLE I
ASSUMPTIONS USED IN ANALYSIS FOR RAREST-FIRST POLICY IN SECTION III AND IN SIMULATIONS IN SECTION V

| Assumptions | Theoretical analysis | Simulations |
|---|---|---|
| Unlimited upload capacity | ✓ | ✗ |
| Global knowledge | ✓ | ✓ |
| The $c$ parameter | ✓ | ✗ |
| Fixed peer population $N$ | ✓ | ✗ |
| Departure only in Class $K$ | ✓ | ✗ |
| Independence assumption | ✓ | ✗ |
| Mean field assumption | ✓ | ✗ |

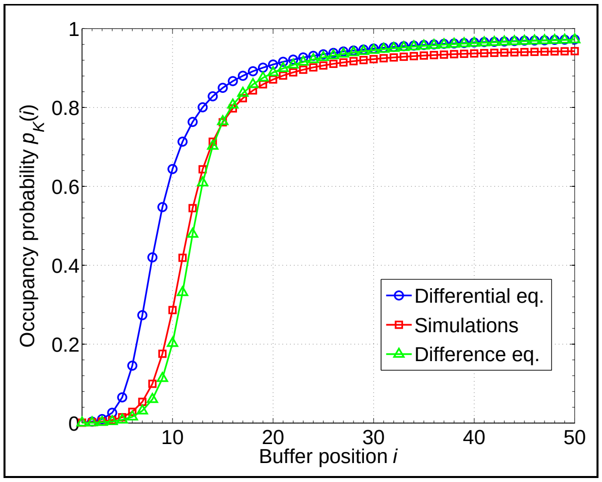



Fig. 4. Comparison of the buffer occupancy probability for class $K$ peers

only happen in Class $K$, the independence assumptions across chunks and peers, and the mean field assumption) are all relaxed in the simulations. We also relax the unlimited upload capacity assumption by using a long term upload bandwidth constraint, i.e., the cumulative number of uploads provided by a peer is at most its current age. We summarize the assumption difference between our theoretical model for the rarest-first and the simulations in Table I.

### A. Model Validation

For the rarest-first policy, we have made a number of simplifying assumptions (as listed in Table I) to obtain the difference equation (8) and the differential equation (9), which describe the evolution of the buffer occupancy probabilities under the rarest-first policy. It is difficult to validate these assumptions one by one. Instead, we compare the buffer occupancy probabilities obtained from the difference and the differential equations, which were derived based on simplifying assumptions, with the buffer occupancy probabilities obtained under a realistic simulation model.

In Fig. 4 we compare the network evolution model given by the difference equation (8), the fluid model given by the differential equation (9), and the results of our simulations. We see that both the difference and differential equations give good estimates of the playout probability despite the simplifying assumptions. Also, the slightly overestimated playout probability by the equations makes the lower bound of the buffer size in Theorem 2 still valid.

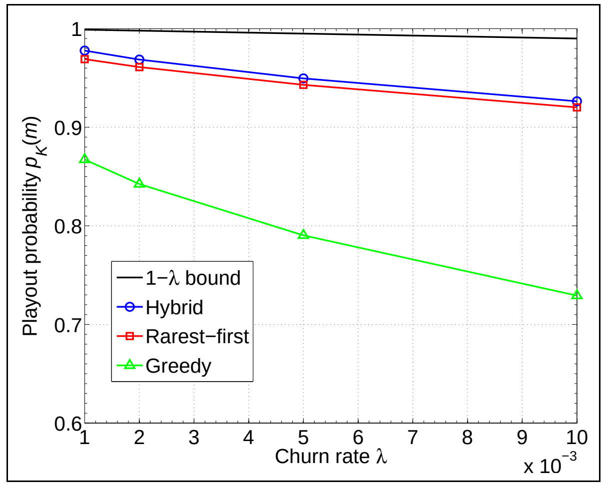



Fig. 5. Policy comparison for different values of expected churn rates

### B. Simulation Analysis

In this section we present a simulation-based study to compare the performance of the rarest-first policy against two other policies: the greedy policy and the hybrid policy. The simulation settings we have used are default to the previous settings unless otherwise specified.

Compared to the rarest-first policy, which gives priority to the chunks farthest from playout, the greedy policy gives priority to the chunks closest to the playout, so the most urgent chunks get duplicated first. The hybrid policy [13], on the other hand, tries to use the best aspects of both the greedy and rarest-first policies in order to enhance the playout probability: a peer first tries to download a useful chunk in the lowest buffer position (rarest-first) up to a buffer position threshold $\theta$; if no chunk can be exchanged then the peer tries to download a chunk in the highest buffer position (greedy) that it can get. The hybrid can also be thought of as a generalization of both the greedy ($\theta = 0$) and the rarest-first policies ($\theta = m - 1$).

*1) Playout Probability versus Churn Rate:* In Fig. 5 we compare the performance of the policies against the bound of Theorem 4, assuming that the hybrid uses the *optimal threshold*[3] for every churn rate. As can be seen, both the hybrid and the rarest-first policies have a performance close to our upper bound, while the greedy performs poorly. We also note that the hybrid policy is only slightly better than the rarest-first policy.

*2) Playout Probability versus Buffer Size:* We have shown an upper bound of the buffer size $m$ for the rarest-first policy under peer churn in Theorem 1. Thus, it is also interesting to study how both the greedy and hybrid perform under different buffer sizes. In Fig. 6 we see that the playout probabilities of the three policies increase along the buffer size for $m \leq 100$, then the playout probability of the greedy policy gradually decreases as the buffer size further increases beyond $100$, while the playout probabilities of the other

[3]We simulated the hybrid policy under all possible thresholds, and chose the best one.

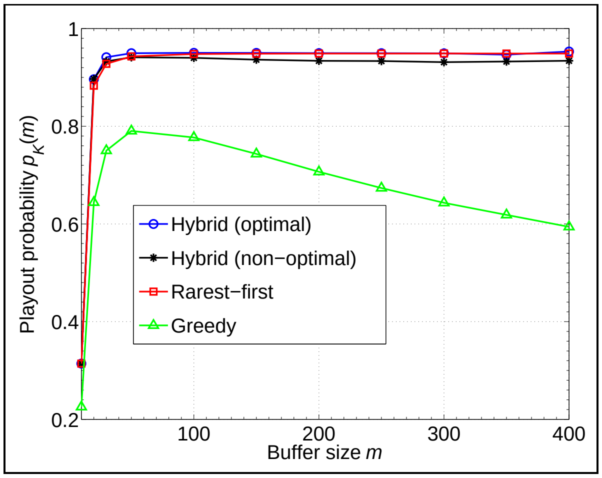



Fig. 6. Policy comparison for different values of buffer size

two policies remain basically unchanged. This is due to the fact that the greedy policy gives priority to chunks close to playout, instead of those recently generated chunks, so the probability that a chunk departs before being duplicated increases significantly, which is only exacerbated by larger buffer sizes. On the other hand, the rarest-first and the hybrid policies give priority to newly generated chunks such that they are more likely to "survive" the departures.

We also note that the playout probabilities of the rarest-first and the hybrid policies become "saturated" and do not get further improvement when the buffer sizes are large, which is in contrast to the static peer scenario where the playout probability tends to 1 as the buffer size increases for any of the three chunk selection policies [15]. This agrees with Theorem 2 in that there is a highest allowable churn rate for a given target playout probability, and conversely there is a highest possible playout probability under a given churn rate. Since larger buffers also lead to smaller fraction of peers in the steady class, the overall playout probability for the peers could reduce.

Therefore, we conclude that a good chunk selection policy focuses its efforts on duplicating recently generated chunks, and the buffer size should be neither too small nor too large.

*3) The Impact of Threshold Values on the Hybrid Policy:* Since the rarest-first policy is a special case of the hybrid policy with threshold set to be $m-1$, the latter performs better than the former when the threshold of the hybrid policy is carefully tuned. In reality, however, it is difficult to find the optimal threshold for the hybrid policy due to the lack of knowledge about peer population and churn rate. Therefore, we need to understand the impact of threshold value on the hybrid policy to determine whether the hybrid policy should be preferred than the rarest-first policy in practice.

Consequently, we study the sensitivity of the optimal threshold with respect to the churn rate and peer population in Figs. 7 and 8. We see that the optimal threshold varies along with the churn rate as well as the peer population. Furthermore, in Fig. 6 we have both the hybrid with optimal threshold (the threshold is 12 for all buffer size) and the hybrid with non-optimal threshold (the threshold is 8). Thus, when the threshold is not set optimally, it could happen that the rarest-first policy performs better than the hybrid policy.

From the above analysis we conclude that the rarest-first policy is a simple, conservative, yet robust algorithm in the presence of peer churn. The hybrid algorithm has a better performance when the threshold can be carefully tuned, which may be very difficult in practice.

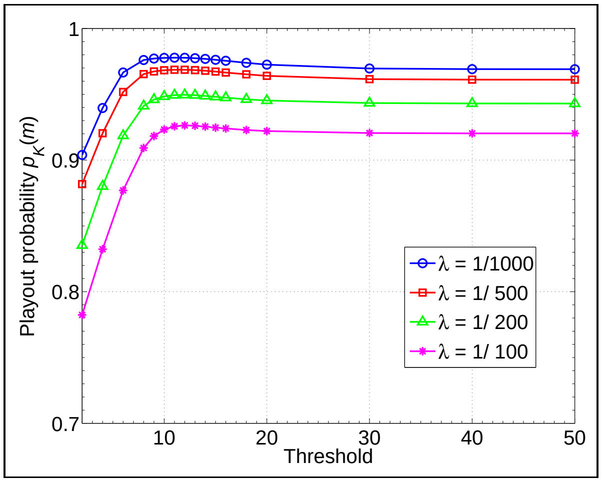
*4) Summary of Simulation Analysis:* The rarest-first and the hybrid policies give much better playout probabilities than the greedy policy does for various level of peer churn, and although the hybrid is slightly better than the rarest-first, the low complexity and easy tuning makes rarest-first policy a reasonable choice in practice.

## VI. CONCLUSIONS

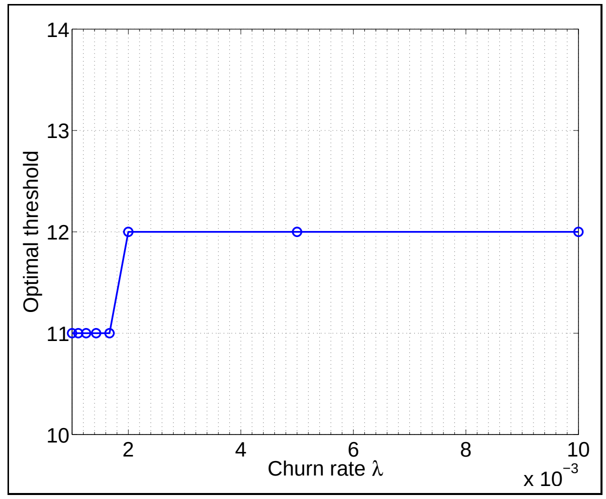
In this paper we have studied the impact of peer churn on P2P streaming networks. We have developed an analytical model that allows us to prove that the level of peer churn in the system limits the playout probabilities that can be achieved. We showed that under some assumptions we can present a model of the rarest-first policy that is tractable and that allows us to prove that in order to get non-negligible playout probabilities there is both an upper bound and a lower bound on the buffer size that can be allowed. Furthermore, we have showed that for a given target playout probability the rarest-first policy has a capacity on the maximum churn rate that it can support. We have also showed two quantitatively equal upper bounds of the playout probability due to peer churn for any chunk selection policy. Using simulations we have validated the theoretical model for the rarest-first policy and shown that the rarest-first policy is a robust alternative for the hybrid policy.

## REFERENCES

[1] "Cisco visual networking index: Forecast and methodology, 2010–2015," White Paper, Cisco, Jun. 1, 2011. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf
[2] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," in *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, ser. NOSSDAV '02, Miami, FL, 2002, pp. 177–186.
[3] G. Dan, V. Fodor, and I. Chatzidrossos, "On the performance of multiple-tree-based peer-to-peer live streaming," in *INFOCOM 2007, IEEE*, Anchorage, AK, May 2007, pp. 2556–2560.
[4] J. Mundinger, R. Weber, and G. Weiss, "Optimal scheduling of peer-to-peer file dissemination," *Journal of Scheduling*, vol. 11, pp. 105–120, 2008.
[5] N. Magharei, R. Rejaie, and Y. Guo, "Mesh or multiple-tree: A comparative study of live P2P streaming approaches," in *INFOCOM 2007, IEEE*, Anchorage, AK, May 2007, pp. 1424–1432.
[6] X. Zhang, J. Liu, B. Li, and T. S. P. Yum, "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming," in *INFOCOM 2005, IEEE*, vol. 3, Miami, FL, Mar. 2005, pp. 2102–2111.
[7] A. Vlavianos, M. Iliofotou, and M. Faloutsos, "BiToS: Enhancing BitTorrent for supporting streaming applications," in *INFOCOM 2006, IEEE*, Barcelona, Spain, Apr. 2006, pp. 1–6.
[8] F. Pianese, D. Perino, J. Keller, and E. Biersack, "PULSE: An adaptive, incentive-based, unstructured P2P live streaming system," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1645–1660, Dec. 2007.
[9] "PPTV," http://download.pptv.com/en/.
[10] "PPStream," http://www.pps.tv/en/.

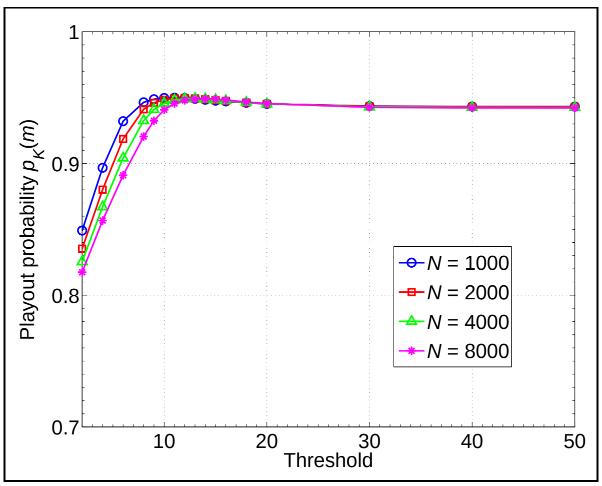



(a) Playout probability for different churn rates

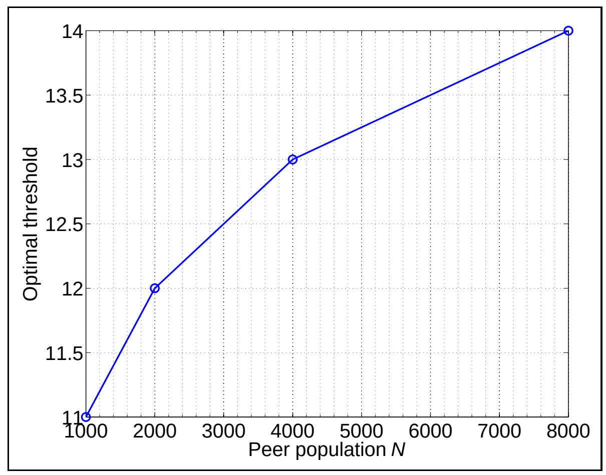



(b) Optimal threshold

Fig. 7. Sensitivity of the threshold choice with respect to the churn rate




(a) Playout probability for different peer populations




(b) Optimal threshold

Fig. 8. Sensitivity of the threshold choice with respect to the peer population


[11] M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Suel, and D. Yao, "Optimal peer selection for P2P downloading and streaming," in *INFOCOM 2005, IEEE*, vol. 3, Miami, FL, Mar. 2005, pp. 1538–1549.

[12] B. Zhao, J. Lui, and D. M. Chiu, "Exploring the optimal chunk selection policy for data-driven P2P streaming systems," in *Peer-to-Peer Computing, 2009. P2P '09. IEEE Ninth International Conference on*, Seattle, WA, Sep. 2009, pp. 271–280.

[13] Y. Zhou, D. M. Chiu, and J. C. Lui, "A simple model for analyzing P2P streaming protocols," in *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*, Beijing, China, Oct. 2007, pp. 226–235.

[14] N. Parvez, C. Williamson, A. Mahanti, and N. Carlsson, "Analysis of BitTorrent-like protocols for on-demand stored media streaming," in *Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '08, Annapolis, MD, 2008, pp. 301–312.

[15] S. Shakkottai, R. Srikant, and L. Ying, "The asymptotic behavior of minimum buffer size requirements in large P2P streaming networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 928–937, May 2011.

[16] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: Optimal performance trade-offs," in *Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '08, Annapolis, MD, Jun. 2008, pp. 325–336.

[17] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, ser. IMC '06, Rio de Janeriro, Brazil, 2006, pp. 189–202.

[18] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *INFOCOM 2007, IEEE*, Anchorage, AK, May 2007, pp. 919–927.

[19] S. Sanghavi, B. Hajek, and L. Massoulie, "Gossiping with multiple messages," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4640–4654, Dec. 2007.

[20] B. Li, S. Xie, G. Keung, J. Liu, I. Stoica, H. Zhang, and X. Zhang, "An empirical study of the Coolstreaming+ system," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1627–1639, Dec. 2007.

[21] B. Li, G. Keung, S. Xie, F. Liu, Y. Sun, and H. Yin, "An empirical study of flash crowd dynamics in a P2P-based live video streaming system," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, New Orleans, LA, Dec. 2008, pp. 1–5.

[22] F. Liu, B. Li, L. Zhong, B. Li, and D. Niu, "How P2P streaming systems scale over time under a flash crowd?" in *Proceedings of the 8th international conference on Peer-to-peer systems*, ser. IPTPS'09, Boston, MA, 2009, pp. 5–5.

[23] X. Kang, J. J. Jaramillo, and L. Ying, "Impacts of peer churn on P2P streaming networks," Iowa State University, Tech. Rep., 2012. [Online]. Available: http://home.engineering.iastate.edu/~leiying/Publications/p2pstreamarvix.pdf