Analysis of Buffer Starvation With Application to Objective QoE Optimization of Streaming Services

Yuedong Xu, Eitan Altman, Rachid El-Azouzi, Majed Haddad, Salah Eddine Elayoubi, Tania Jimenez

HAL Id: hal-01005726
https://inria.hal.science/hal-01005726

Submitted on 13 Jun 2014

Analysis of Buffer Starvation with Application to Objective QoE Optimization of Streaming Services

Yuedong Xu, Eitan Altman, *Fellow, IEEE,* Rachid El-Azouzi, Majed Haddad,

Salaheddine Elayoubi, and Tania Jimenez

Abstract

Our purpose in this paper is to characterize buffer starvations for streaming services. The buffer is modeled as a FIFO queue with exponential service time and Poisson arrivals. When the buffer is empty, the service restarts after a certain amount of packets are *prefetched*. With this goal, we propose two approaches to obtain the *exact distribution* of the number of buffer starvations, one of which is based on *Ballot theorem*, and the other uses recursive equations. The Ballot theorem approach gives an explicit result. We extend this approach to the scenario with a constant playback rate using Tàkacs Ballot theorem. The recursive approach, though not offering an explicit result, allows us to obtain the distribution of starvations with non-independent and identically distributed (i.i.d.) arrival process in which an ON/OFF bursty arrival process is considered in this work.We further compute the starvation probability as a function of the amount of prefetched packets for a large number of files via a fluid analysis. Among many potential applications of starvation analysis, we show how to apply it to optimize the objective quality of experience (QoE) of media streaming, by exploiting the tradeoff between startup/rebuffering delay and starvations.

Index Terms

Starvation, Start-up Delay, Quality of Experience, Ballot Theorem

EDICS: 8-MSAT, **3-QAUE**

I. Introduction

The starvation probability of a buffer is an important performance measure for protocol design of telecommunication networks, as well as in storage systems and ecological systems (e.g. dams). Starvation is said to occur when the buffer is empty. Various applications use buffering in order to control the rate at which packets are served at the destination. As long as there are packets in the buffer, packets arrive at the destination regularly, i.e. they are spaced by the service time of the buffer. Once the buffer empties packets may arrive at the destination separated by larger times, as the spacing between packets now depends also on the inter-arrival times at the queue. Starvation is in particular undesirable in video streaming applications.

The time till starvation of a queue is related to the busy period which has been well studied under the assumption of a stationary arrival process (see [2], [3] and their references). In contrast to this assumption, we consider a finite number of arrivals as we are interested in statistics of starvation when a file of fixed size is transferred.

The main goal of this paper is to find the *distribution of the number of starvations* within a file of N packets. We first model the buffer as an M/M/1 queue, and then extend it to incorporate the bursty packet arrival that is modeled by an *interrupted Poisson process (IPP)*. In this system, a fixed amount of packets are *prefetched* (also called "prefetching threshold") before the service begins or resumes after a starvation event.

In this paper, we mainly propose two approaches (that give the same result) to compute the distribution of the number of starvations for a single file. The first approach gives an explicit result based on the famous *Ballot theorem* [1]. But it is in general suitable for independent and identically distributed (i.i.d.) arrival process. This motivates us to propose the second approach that yields a more flexible recursive algorithm. Although it has no explicit result, it can be used to compute the starvation probabilities for more complicated arrival processes. In terms of complexity, the recursive approach can compute the starvation probabilities for different combinations of the initial start-up threshold and the file size in one round, which means an overall smaller complexity than the Ballot theorem approach.

The key feature of Ballot theorem is its simple expression to compute the probability that a counting process (e.g. arrival process) is strictly ahead of another counting process (e.g. departure process). Using Ballot theorem, we can compute in a simple way the exact distribution of the number of starvations explicitly. When the file size is large enough, we present the asymptotic starvation probability using Gaussian (interchangeable with Normal) approximation as well as an approximation of the Riemann

integral. As a special case that the playout buffer is modeled by an M/D/1 queue, the probabilities of the number of starvations can be obtained on the basis of a discrete version of Ballot theorem. Furthermore, unlike the Ballot theorem that usually requires the i.i.d. packet arrivals, the recursive approach enables us to compute the probability of starvations with continuous-time ON/OFF bursty packet arrivals. The media server transfers packets during the ON state and pauses during the OFF state. It switches from one state to the other after residing an exponentially distributed time period.

We further propose a fluid analysis of starvation behavior on the file level. This approach, instead of looking into the stochastic packet arrivals and departures, predicts the starvation where the servers manage a large quantity of file transfers. Given the traffic intensity and the distribution of file size, we are able to compute the starvation probability as a function of the prefetching threshold. The fluid analysis, though simple, offers an important insight on how to control the probability of starvation for many files, instead of for one particular file.

The probabilities of starvations developed in this work have various applications in the different fields. A prominent example is the media streaming service. This application demonstrates a dilemma between the prefetching process and the starvation. A longer prefetching process causes a larger start-up/rebuffering delay, while a shorter one might result in starvations. The user perceived media quality (or QoE equivalently) is impaired by two major factors, the large start-up/rebuffering delay and the frequent starvations, according to the measurement studies in [27], [28]. These two factors are thus defined as quality metrics of streaming service. The QoE study becomes increasingly important in the epoch that web video hogs up to more than 37% of total traffic during peak hours in USA [18]. In contrast to the rapid growth of traffic load, the bandwidth provision usually lags behind. In this context, media providers and network operators face a crucial challenge of maintaining a satisfactory QoE of streaming service. With the results developed in this work, we are able to answer the fundamental question: *How many packets should the media player prefetch to optimize the users' quality of experience?*

To answer this question, we first use objective QoE costs to model the subjective human unhappiness for both finite and infinite file size. An objective QoE cost is the weighted sum functions on the start-up/rebuffering delay and the starvation behaviors. The weight reflects an individual user's relative impatience on the delay than on the starvations. We then formulate the objective QoE optimization problems in a variety of scenarios. After solving them, we obtain the optimal algorithms to configure the start-up/rebuffering thresholds in packets. Lastly, we discuss the possible implementation of the proposed algorithms. A streaming user can dynamically configure the rebuffering threshold. When a starvation happens, the packet arrival rate can be measured at the user side. Given the knowledge the arrival

rate, service rate and the remaining file size, the user can set the rebuffering threshold to balance the tradeoffs: i) delay vs starvation probability (section VI-A) for a small remaining file size and ii) delay vs starvation interval (section VI-B) for a large remaining file size. The media server can configure the start-up threshold for each category of videos (e.g. music, sports, and TV news) beforehand. Given the distributions of file size and the user throughput, it can use the algorithm in section VI-C to compute the optimal start-up threshold. Simulation studies in section VI demonstrate the relationships between the objective QoE costs and the optimal prefetching thresholds in a variety of scenarios.

We summarize the main results as follows.

- We propose a Ballot theorem approach to compute the distribution of the number of starvations for a file of finite size in an M/M/1 queue. It provides an explicit solution and is generalized to an M/D/1 queue.

- We propose a recursive approach to compute the distribution of the number of starvations. We further extend it to the ON/OFF bursty arrival process.

- We present a fluid model to compute the starvation probability, given the start-up threshold and the file size distributions of representative content types.

- Our study provides the important understandings on how the starvation probabilities are impacted by the prefetching threshold. We propose a set of strategies to configure better prefetching thresholds so as to balance the tradeoff between the delay and the starvations for video streaming services.

The rest of this paper is organized as follows. Section II reviews the related work. We propose a Ballot approach in Section III. Section IV presents the recursive approach for an M/M/1 queue and extents it to the ON/OFF bursty arrival process. Section V performs a fluid analysis for a large number of files. Section VI presents the QoE metrics and their optimization issues. Our theoretical results are verified in section VII. Section VIII concludes this paper.

II. RELATED WORK

The analysis of starvation is close to that of busy period in transient queues. In [2], [3] authors solve the distribution of the buffer size as a function of time for the M/M/1 queue. Ballot theorem and recursive equations have been used to analyze the packet loss probability in a finite buffer when the forward error-correcting technique is deployed. Citon et al. [9] propose a recursive approach that enables them to compute the packet loss probability in a block of consecutive packet arrivals into an M/M/1/K queue. Based on their recursive approach, Altman and Jean-Marie in [10] obtain the expressions for the multidimensional generating function of the packet loss probability. The distribution of message delay

is given in an extended work [11]. Dubea and Altman in [12] analyze the packet loss probability with the consideration of random loss in incoming and outgoing links. In [14], Gurewitz et al. introduce the powerful Ballot theorem to find this probability within a block of packet arrivals into an M/M/1/K queue. They consider two cases, in which the block size is smaller or greater than the buffer limit. Another example of applying Ballot theorem to evaluate networking system is found in [13]. Humblet et al. present a method based on Ballot theorem to study the performance of nD/D/1 queue with periodical arrivals and deterministic service time. In [16], He and Sohraby use Ballot theorem to find the stationary probability distribution in a general class of discrete time systems. Privalov and Sohraby [17] study the underflow behavior of CBR traffic in a time-slotted queueing system. However, they do not provide the insights of having a certain number of starvations.

In the related applications related to our work, Stockhammer et al. [15] specify the minimum start-up delay and the minimum buffer size for a given video stream and a deterministic variable bit rate (VBR) wireless channel. Recently, [6] presents a deterministic bound, and [7] provides a stochastic bound of start-up delay to avoid starvation. Authors in [5] model the playout buffer as a G/G/1 queue. By using diffusion approximation, they obtain the closed-form starvation probability with asymptotically large file size. Xu et.al [21] study the scheduling algorithms for multicast streaming in multicarrier wireless downlink. Authors in [23] studied the QoE metrics of a persistent video streaming in cellular networks. They further presented a new method in [24] to compute the QoE metrics for a cellular network with arrivals and departures of streaming flows. Our paper differs from some of the above works in the following ways: i) we present new theories that yield an *exact* probability of starvation, and the p.g.f. of starvation events; ii) we study the asymptotic behavior with error analysis; iii) we perform a macroscopic starvation analysis using a fluid model; iv) we configure optimal prefetching thresholds to optimize the QoE metrics.

III. STARVATION ANALYSIS USING BALLOT THEOREM

In this section, we study the starvation behavior of an M/M/1 (also extended to M/D/1) queue with finite number of arrivals based on the powerful Ballot theorem.

A. System Description

We consider a single media file with finite size N. The media content is pre-stored in the media server (e.g. video on demand (VoD) service). When a user makes a request, the server segments this media into packets, and transfers them to the user by use of TCP or UDP. When packets traverse the wired or

wireless links, their arrivals to the media player of a user are not deterministic due to the dynamics of the available bandwidth. The Poisson assumption is not the most realistic way to describe packet arrivals, but it reveals the essential features of the system, and is the first step for more general arrival processes. After the streaming packets are received, they are first stored in the playout buffer. The interval between two packets that are served is assumed to be exponentially distributed so that we can model the receiver buffer as an M/M/1 queue. The maximum buffer size is assumed to be large enough so that the whole file can be stored. This simplification is justified by the fact that the storage space is usually very large in the receiver side (e.g. several GB).

The user perceived media quality has two measures called *start-up delay* and *starvation*. As explained earlier, the media player wants to avoid the starvation by prefetching packets. However, this action might incur a long waiting time. In what follows, we reveal the relationship between the start-up delay and the starvation behavior, with the consideration of file size.

B. A Packet Level Model

We present a packet level model to investigate the starvation behavior. We denote by λ the Poisson arrival rate of the packets, and by μ the Poisson service rate. We define $\rho := \lambda/\mu$ as the traffic intensity.

In a non-empty M/M/1 queue with everlasting arrivals, the rate at which either an arrival or a departure occurs is given by $\lambda + \mu$. This event corresponds to an arrival with probability p, or is otherwise to an end of service with probability q, where

$$p = \frac{\lambda}{\lambda + \mu} = \frac{\rho}{1 + \rho}; \quad q = \frac{\mu}{\lambda + \mu} = \frac{1}{1 + \rho}.$$

The buffer is initially empty. We let T_1 be the start-up delay, in which x_1 packets are accumulated in the buffer. Our analysis of the probability of starvation is built on the famous Ballot theorem:

Ballot Theorem: *In a ballot, candidate A scores N_A votes and candidate B scores N_B votes, where $N_A > N_B$. Assume that while counting, all the ordering (i.e. all sequences of A's and B's) are equally alike, the probability that throughout the counting, A is always ahead in the count of votes is $\frac{N_A - N_B}{N_A + N_B}$.*

After the service begins, the probability of starvation is given by Theorem 1.

Theorem 1: For the initial queue length x_1 and the total size N of a file, the probability of starvation is given by:

$$P_s = \sum_{k=x_1}^{N-1} \frac{x_1}{2k - x_1} \binom{2k - x_1}{k - x_1} p^{k - x_1} (1 - p)^k. \tag{1}$$

Proof: We define E_k to be an event that the buffer becomes empty for the first time when the service of packet k is finished. It is obvious that all the events $E_k, k = 1, \cdots N$, are mutually exclusive. Then, the event of starvation is the union $\cup_{k=x_1}^{N-1} E_k$. This union of events excludes E_N because the empty buffer after the service of N packets is not a starvation. When the buffer is empty at the end of the service of the kth packet, the number of arrivals is $k - x_1$ after the prefetching process. The probability of having $k - x_1$ arrivals and k departures is computed from a binomial distribution, $\binom{2k-x_1}{k-x_1}p^{k-x_1}(1-p)^k$. We next find the necessary and sufficient condition of the event E_k. If we have a backward time axis that starts from the time point when the buffer is empty for the first time, the number of departure packets is always more than that of arrival packets. As a result, the Ballot Theorem can be applied. For example, among the last m events (i.e. $m \leq 2k - x_1$), the number of packets that have been played is always greater than the number of arrivals. Otherwise, the empty buffer already happens before the kth packet is served. According to the Ballot theorem, the probability of event E_k is computed by $\frac{x_1}{2k-x_1}\binom{2k-x_1}{k-x_1}p^{k-x_1}q^k$. Therefore, the probability of starvation, P_s, is the probability of the union $\cup_{k=x_1}^{N-1} E_k$, given by eq.(1). ∎
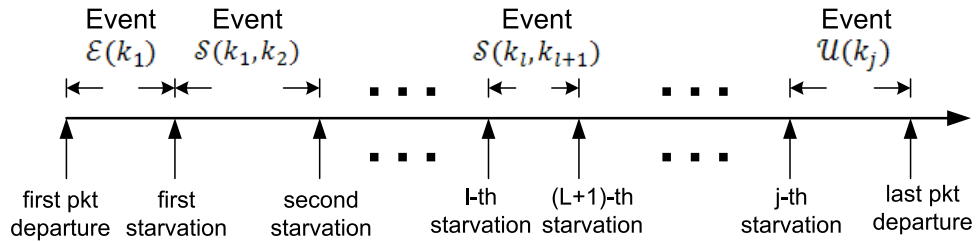

Fig. 1. A path with j starvations

The starvation event may happen for more than once during the file transfer. We are particularly interested in the probability distribution of starvations, given a finite file size N. The maximum number of starvations is $J = \lfloor \frac{N}{x_1} \rfloor$ where $\lfloor \cdot \rfloor$ is the floor of a real number. We define *path* as a complete sequence of packet arrivals and departures. The probability of a path depends on the number of starvations. We illustrate a typical path with j starvations in Figure 1. To carry out the analysis, we start from the event that the first starvation takes place. We denote by k_l the lth departure of a packet that sees an empty queue. We notice that the path can be decomposed into three types of mutually exclusive events as follows:

- Event $\mathcal{E}(k_1)$: the buffer becoming empty for the first time in the entire path.
- Event $\mathcal{S}_l(k_l, k_{l+1})$: the empty buffer after the service of packet k_{l+1} given that the previous empty buffer happens at the departure of packet k_l.
- Event $\mathcal{U}_j(k_j)$: the last empty buffer observed after the departure of packet k_j.

Obviously, a path with j starvations is composed of a succession of events

$$\mathcal{E}(k_1), \mathcal{S}_1(k_1, k_2), \mathcal{S}_2(k_2, k_3), \cdots, \mathcal{S}_{j-2}(k_{j-2}, k_{j-1}), \mathcal{S}_{j-1}(k_{j-1}, k_j), \mathcal{U}_j(k_j).$$

We let $P_{\mathcal{E}(k_1)}$, $P_{\mathcal{S}_l(k_l, k_{l+1})}$ and $P_{\mathcal{U}_j(k_j)}$ be the probabilities of events $\mathcal{E}(k_1)$, $\mathcal{S}_l(k_l, k_{l+1})$ and $\mathcal{U}_j(k_j)$ respectively. The main difficulty to analyze the probability mass function is that the media player pauses for x_1 packets upon starvation. In what follows, we analyze the probabilities of these events step by step. The event $\mathcal{E}(k_1)$ can happen after the departure of packet $k_1 \in [x_1, N-1]$. According to the proof of Theorem 1, the probability distribution of event $\mathcal{E}(k_1)$ can be expressed as

$$P_{\mathcal{E}(k_1)} := \begin{cases} 0 & \text{if } k_1 < x_1 \text{ or } k_1 = N; \\ \frac{x_1}{2k_1 - x_1} \binom{2k_1 - x_1}{k_1 - x_1} p^{k_1 - x_1} q^{k_1} & \text{otherwise .} \end{cases} \tag{2}$$

The first starvation cannot happen at the departure of first $(x_1 - 1)$ packets because of the prefetching of x_1 packets. It cannot happen after all N packets have been served because this empty buffer is not a starvation. We next solve the probability distribution of the event $\mathcal{U}_j(k_j)$. Suppose that there are j starvations after the service of packet k_j. The extreme case is that these j starvations take place consecutively. Thus, k_j should be greater than $jx_1 - 1$. Otherwise there cannot have j starvations. The starvation event cannot take place after the departure of packet N because the whole file has now been transferred. If k_j is no less than $N - x_1$, the media player resumes until all the remaining $N - k_j$ packets are stored in the buffer. Then, starvation will not appear afterwards. In the remaining cases, the event $\mathcal{U}_j(k_j)$ is equivalent to the event that no starvation happens after the service of packet k_j. We can take the complement of starvation probability as the probability of no starvation. Hence, the probability distribution of event $\mathcal{U}_j(k_j)$ is given by

$$P_{\mathcal{U}_j(k_j)} := \begin{cases} 0, & \text{if } k_j < jx_1 \text{ or } k_j = N; \\ 1, & \text{if } N - x_1 \le k_j < N; \\ 1 - \sum_{m=x_1}^{N-k_j-1} \frac{x_1}{2m - x_1} \binom{2m - x_1}{m} p^{m - x_1} q^m, & \\ & \text{otherwise .} \end{cases} \tag{3}$$

We denote by $P_s(j)$ the probability of having j starvations. The probability $P_s(0)$ can be obtained from Theorem 1 directly. For the case with one starvation, $P_s(1)$ is solved by

$$P_s(1) = \sum_{i=1}^{N} P_{\mathcal{E}(i)} P_{\mathcal{U}_1(i)} = \mathbf{P}_{\mathcal{E}} \cdot \mathbf{P}_{\mathcal{U}_1}^T \tag{4}$$

where ^{T} denote the transpose. Here, $\mathbf{P}_{\mathcal{E}}$ is the row vector of $P_{\mathcal{E}(i)}$, and $\mathbf{P}_{\mathcal{U}_1}$ is the row vector of $P_{\mathcal{U}_1(i)}$, for $i = 1, 2, \cdots, N$.

To compute the probability of having more than one starvations, we need to find the probability of event $\mathcal{S}_l(k_l, k_{l+1})$ beforehand. Solving $P_{\mathcal{S}_l(k_l, k_{l+1})}$ is non-trivial due to that the probability of this event depends on the remaining file size and the number of starvations. After packet k_l is served, the lth starvation is observed. It is clear that k_l should not be less than lx_1 in order to have l starvations. Given that the buffer is empty after serving packet k_l, the $(l+1)^{th}$ starvation cannot happen at $k_{l+1} \in [k_l + 1, k_l + x_1 - 1]$ because of the subsequent prefetching process. Since there are j starvations in total, the $(l+1)^{th}$ starvation must satisfy $k_{l+1} < N - (j - l - 1)x_1$. We next compute the remaining case that the lth and the $(l+1)^{th}$ starvations happen after packets k_l and k_{l+1} are served. Then, there are $(k_{l+1} - k_l)$ departures, and $(k_{l+1} - k_l - x_1)$ arrivals after the prefetching process. According to the Ballot theorem, a path without starvation between the departure of packet $(k_l + 1)$ and that of packet (k_{l+1}) is expressed as $\frac{x_1}{2k_{l+1} - 2k_l - x_1}$. Therefore, we can express $P_{\mathcal{S}_l(k_l, k_{l+1})}$ as

$$
\begin{cases}
\frac{x_1}{2k_{l+1} - 2k_l - x_1} \binom{2k_{l+1} - 2k_l - x_1}{k_{l+1} - k_l - x_1} p^{k_{l+1} - k_l - x_1} q^{k_{l+1} - k_l}, \\
\quad \text{if } k_l \geq lx_1, k_l + x_1 \leq k_{l+1} < N - (j - l - 1)x_1; \\
0, \qquad \text{otherwise .}
\end{cases}
\tag{5}
$$

We denote by $\mathbf{P}_{\mathcal{S}_l}$ the matrix of $P_{\mathcal{S}_l(k_l, k_{l+1})}$ for $k_l, k_{l+1} \in [1, N]$. Here, $\mathbf{P}_{\mathcal{S}_l}$ is an upper triangle matrix where all the elements in the first $(lx_1 - 1)$ rows, and the last x_1 rows are 0. The probability of having $j (j \geq 2)$ starvations is given by

$$
P_s(j) = \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} \cdots \sum_{k_{j-1}=1}^{N} \sum_{k_j=1}^{N} P_{\mathcal{E}(k_1)} \cdot P_{\mathcal{S}_1(k_1, k_2)} \cdots P_{\mathcal{S}_{j-1}(k_{j-1}, k_j)} \cdot P_{\mathcal{U}_j(k_j)} = \mathbf{P}_{\mathcal{E}} \left(\prod_{l=1}^{j-1} \mathbf{P}_{\mathcal{S}_l} \right) \mathbf{P}_{\mathcal{U}_j}^{T}.
\tag{6}
$$

The probability of no starvation, $P_s(0)$, is computed as $1 - P_s$ where P_s is obtained from eq.(1). Since the starvation event takes non-negative integer values, we can write the probability generating function (p.g.f.) $G(z)$ by

$$
G(z) = E(z^j) = \sum_{j=0}^{J} P_s(j) \cdot z^j.
\tag{7}
$$

In $\mathbf{P}, \mathbf{P}_{\mathcal{S}_l}$ and $\mathbf{P}_{\mathcal{U}_j}$, the binomial distributions can be approximated by the corresponding Normal distributions with negligible errors (see technical report [22]). The Gaussian approximation significantly reduces the computational complexity of binomial distributions.

We next analyze the complexity of matrix (including vector) operations in eq. (6). A matrix operation consists of floating-point operations where one floating-point operation can be an addition, subtraction, multiplication or division of two float type matrix elements [29]. In the complexity analysis of matrix operations, the lower-order terms are usually ignored. The approximated probability of starvation in eq.(1) consists of N additions, thus having a complexity order $O(N)$. The probability of one starvation is a

product of two vectors, which consists of N multiplications and $N-1$ additions. Hence, the complexity order in still $O(N)$. If there are only two starvations, we need to compute the product of two vectors and one matrix, which has a complexity order $O(N^2)$. When $j \geq 3$, the computation of $P_s(j)$ involves the product of two matrices. In general, multiplying two matrices has a complexity order $O(N^3)$. In eq.(6), we should multiply a vector and a matrix each time, instead of multiplying two matrices inside the bracket. Then, the matrix operations in eq.(6) only contain a set of multiplications between a vector and a matrix. This yields the complexity order $O(N^2)$. To sum up, the computation of the p.g.f. of starvations in eq.(7) has a complexity order $O(N^2)$, given the start-up/rebuffering threshold x_1 and the file size N.

Asymptotic Property:

We want to know whether the starvation event yields simple implications as the file size N approaches ∞. The asymptotic behavior of the starvation probability is given by

$$\lim_{N \to \infty} P_s := \begin{cases} 1 & \text{if } \rho < 1; \\ \exp\left(\frac{x_1(1-2p)}{2pq}\right) & \text{otherwise} . \end{cases} \tag{8}$$

The detailed analysis can be found in the technical report [22].

The asymptotic starvation probability is irrelevant to the start-up threshold when $\rho < 1$. Under this situation, it is necessary to know how frequent the starvation event happens. Here, we compute the average time interval between two starvations. We let T_s be the duration of starvation interval. Its expectation $E[T_s]$ is the expected busy period of an M/M/1 queue with x_1 customers in the beginning [4], i.e. $E[T_s] = \frac{x_1}{\lambda(1-\rho)}$.

C. Extension to Discrete-time Systems

In general, the playback rate of video streaming has a much smaller variance than the arrival rate. Hence, the playback of streaming packets is sometimes regarded as a time-slotted process (e.g. [6], [7]) where only one packet is served at the beginning of a time slot. We consider a playout buffer modeled as an M/D/1 queue. We denote by d the duration of a slot.

In this subsection, we introduce a discrete Ballot theorem named Takács Ballot Theorem.

Theorem 2: (Takács Ballot Theorem [1]) If $X(1), X(2), \cdots, X(l)$ are cyclically interchangeable r.v.s taking on nonnegative integer values summing to k, then

$$\mathbb{P}\left\{ \sum_{s=1}^{t} X(s) < t, \forall\, t \in [1, l] \right\} = \frac{[l-k]^+}{l}. \tag{9}$$

The Takács Ballot Theorem presents a probability that the number of departures is larger than that of arrivals in all l slots. If the arrival process $\{X(s)\}$ is Poisson, $X(s)$ is i.i.d. at different slots, and thus cyclically interchangeable. Suppose that the starvation event happens after t packets have been served, $(t \geq x_1)$. The total number of arrivals is $t-x_1$. We create a backward time axis where the starvation event happens at slot 1. The number of departures is always greater than that of arrivals. Otherwise, the starvation event has already taken place. Hence, according to Takács Ballot theorem, the probability of the departure always leading the arrival is

$$\mathbb{P}\{\sum_{s=1}^{t} X(s) < t, \ \forall t \in [1,l]\} = \frac{x_1}{l}. \tag{10}$$

Therefore, the probability that the first starvation takes place after the service of the lth packet (i.e. starvaton event happening at slot $(l+1)$)

$$P_s(l) = \frac{x_1}{l} \cdot \mathbb{P}\{\sum_{s=1}^{l} X(s) = l-x_1\}, \quad \forall l \geq x_1. \tag{11}$$

For the Poisson process $\{X(s)\}$, the probability of $l-x_1$ packet arrivals in l slots (i.e. the duration ld) is obtained by

$$\mathbb{P}\{\sum_{s=1}^{l} X(s) = l-x_1\} = \frac{(\lambda ld)^{l-x_1}}{(l-x_1)!} \exp(-\lambda ld). \tag{12}$$

Given the file size N and the prefetching threshold x_1, the starvation might happens upon the departure of packets from x_1 to $N-1$. Then the starvation probability is obtained by

$$P_s = \sum_{l=x_1}^{N-1} \frac{x_1}{l} \frac{(\lambda ld)^{l-x_1}}{(l-x_1)!} \exp(-\lambda ld). \tag{13}$$

We next show how the p.g.f. of starvation events can be derived using the Takács Ballot theorem. The path with j starvations is the same as that in Fig.1. With certain abuse of notations, we reuse $\mathcal{E}(k_1)$, $\mathcal{U}_j(k_j)$ and $\mathcal{S}_l(k_l, k_{l+1})$ to denote the first, the last and the other starvation events. According to eq.(12), there has

$$P_{\mathcal{E}(k_1)} := \begin{cases} 0 & \text{if } k_1 < x_1 \text{ or } k_1 = N; \\ \frac{x_1}{k_1} \frac{(\lambda k_1 d)^{k_1-x_1}}{(k_1-x_1)!} \exp(-\lambda k_1 d) & \text{otherwise}. \end{cases} \tag{14}$$

Since there exist j starvations in total, the last starvation event will not happen at the departure of packets less than jx_1. Given the last starvation happening as soon as the k_j^{th} packet is served, the probability of

no starvation afterwards can also be solved using eq.(12).

$$P_{\mathcal{U}_j(k_j)} := \begin{cases} 0, & \text{if } k_j < jx_1 \text{ or } k_j = N; \\ 1, & \text{if } N - x_1 \le k_j < N; \\ 1 - \sum_{s=x_1}^{N-k_j-1} \frac{x_1}{s} \frac{(\lambda sd)^{s-x_1}}{(s-x_1)!} \exp(-\lambda sd), \\ & \text{otherwise .} \end{cases}$$

When the lth and the $l+1^{th}$ starvation events appear at the departure of packet k_l and k_{l+1}, the probability $P_{\mathcal{S}_l(k_l, k_{l+1})}$ is given by

$$\begin{cases} \frac{x_1}{k_{l+1}-k_l} \frac{(\lambda(k_{l+1}-k_l)d)^{(k_{l+1}-k_l-x_1)}}{(k_{l+1}-k_l-x_1)!} \exp(-\lambda(k_{l+1}-k_l)d) \\ \quad \text{if } k_l \ge lx_1, k_l + x_1 \le k_{l+1} < N - (j-l-1)x_1; \\ 0, \quad \text{otherwise .} \end{cases} \quad (15)$$

Then, the p.g.f. of starvation events can be solved using eq.(7) in the same way.

IV. STARVATION ANALYSIS VIA A RECURSIVE APPROACH

In this section, we present a recursive approach to compute the p.g.f. of starvations based on [9]. Compared with the one using Ballot theorem, the recursive approach can handle more complicated arrival process.

A. Probability of Starvation

The probability of starvation and the p.g.f can be analyzed all in once. However, we compute them separately because the analysis of the starvation probability provides an easier route to understand this approach.

We denote by $P_i(n)$ the probability of starvation with a file of n packets, given that there are i packets in the system just before the arrival epoch of the first packet of this file. In the original system, our purpose is to obtain the starvation probability of a file with the size N when x_1 packets are prefetched before the service begins. This corresponds to $P_i(n)$ with $n = N+1-x_1$ and $i = x_1 - 1$. *Here, the expression $i = x_1 - 1$ means that the service starts when the x_1-th packet sees $x_1 - 1$ packets accumulated in the buffer. When the service begins, there are already x_1 packets in the queue.* To compute $P_i(n)$, we will introduce recursive equations. We define a quantity $Q_i(k)$, $i = 0, 1, \cdots, n$, $0 \le k \le i$, which is the probability that k packets out of i leave the system during an inter-arrival period. This probability is equivalent to the probability of k Poisson arrivals with rate μ during an exponentially distributed period

with parameter $1/\lambda$. According to [8], we obtain

$$Q_i(k) = \rho\left(\frac{1}{1+\rho}\right)^{k+1} = pq^k, \quad 0 \le k \le i-1, \tag{16}$$

$$Q_i(i) = \left(\frac{1}{1+\rho}\right)^i = q^i. \tag{17}$$

To carry out the recursive calculation, we start from the case $n = 1$.

$$P_i(1) = 0, \quad \forall i \ge 1. \tag{18}$$

When the file size is 1 and the only packet observes a non-empty queue, the probability of starvation is 0 obviously. If i is 0, the starvation happens for sure, thus yielding

$$P_0(n) = 1, \quad \forall n. \tag{19}$$

For $n \ge 2$, we have the following recursive equations:

$$P_i(n) = \sum_{k=0}^{i+1} Q_{i+1}(k) P_{i+1-k}(n-1), \quad 0 \le i \le N-1. \tag{20}$$

We explain (20) as the following. When the first packet of the file arrives and sees i packets in the system, the starvation does not happen. However, the starvation might happen in the service of remaining $n-1$ packets. Upon the arrival of the next packet, k packets out of $i+1$ leave the system with probability $Q_{i+1}(k)$. We next add constraints to the recursive equation (20) for a file of size N. Since the total number of packets is N, the starvation probability must satisfy $P_i(n) = 0$ for $i + n > N$.

B. P.G.F. of Starvations

To compute the p.g.f. of starvation, we use the same recursive approach, despite of the more complicated structure. With certain reuse of notation, we denote by $P_i(j, n)$ the probability of j starvation of a file with size n, given that the first packet of the file sees i packets in the system upon its arrival. Our final purpose is to compute the probability of starvation for a file of size N. It can be obtained from $P_i(j, n)$ with $i = x_1 - 1$ and $n = N + 1 - x_1$. When the first packet of remaining n packets arrives at the buffer, it sees $x_1 - 1$ packets. At this time point, there are x_1 packets in the buffer and the service of packets begins.

In order to compute $P_i(j, n)$ recursively, we provide the initial conditions first:

$$P_i(j,1) = \begin{cases} 0 & \forall i = 1, 2, \cdots, N-1, \text{ and } j \ge 1; \\ 1 & \forall i = 1, 2, \cdots, N-1, \text{ and } j = 0, \end{cases} \tag{21}$$

and

$$P_0(j,1) = \begin{cases} 0 & j = 0 \text{ or } j \geq 2; \\ 1 & j = 1. \end{cases} \tag{22}$$

The equation (21) means that the probability of no starvation is 1 conditioned by $i \geq 1$ and $n = 1$. Thus, the probability of having one or more starvations is 0 obviously if the only packet sees a nonempty system. The equation (22) reflects that the starvation happens for sure when the only packet observes an empty queue. However, there can only have one starvation event due to $n = 1$. Another practical constraint is

$$P_i(j,n) = 0, \quad \text{if } i + n > N \tag{23}$$

because of the finite file size N.

To compute $P_i(j,n)$, we need to know what will happen if the buffer is empty, i.e. $i = 0$. One intuitive observation is

$$P_0(0,n) = 0, \quad \forall \, 1 \leq n \leq N - b; \tag{24}$$

where $b := x_1 - 1$ is denoted to be the prefetching threshold. Eq.(25) holds because an empty queue means at least one starvation event. For a more general probability $P_0(j,n)$, we begin with the case $j = 1$. If $n \leq b$ and the first packet of n sees an empty buffer, there has only one starvation, that is,

$$P_0(1,n) = 1, \quad \forall \, 1 \leq n \leq b, \tag{25}$$

If $n > b$, b packets will be prefetched. Thus, the remaining file size is $n - b$. We see b packets in the system upon the arrival of the first packet in the remaining file. Given that the only one starvation event has taken place, there will be no future starvations. Therefore, the following equality holds,

$$P_0(1,n) = P_b(0,n-b), \quad \forall \, b < n \leq N - b. \tag{26}$$

Using the similar method, we can solve $P_0(j,n)$ for $j > 1$. However, the property of $P_0(j,n)$ with $j > 1$ is quite different

$$P_0(j,n) = 0, \quad \forall \, j > 1 \text{ and } 1 \leq n \leq b. \tag{27}$$

This means that the probability of having > 1 starvations is 0 if the file size is no larger than b. If n is greater than b, then b packets are prefetched, leaving $n - b$ packets in the remaining file. The remaining $n - b$ packets encounter $j - 1$ starvations, given that the first packet sees b packets in the system upon arrival, i.e.

$$P_0(j,n) = P_b(j-1,n-b), \quad \forall \, j > 1 \text{ and } n > b. \tag{28}$$

So far, we have computed a critical quantity $P_0(j,n)$, the probability of meeting an empty buffer. Next, we construct recursive equations to compute $P_i(j,n)$ as the following:

$$P_i(j,n) = \sum_{k=0}^{i+1} Q_{i+1}(k) P_{i+1-k}(j, n-1) = \sum_{k=0}^{i} p q^k P_{i+1-k}(j, n-1) + q^{i+1} P_0(j, n-1), \qquad (29)$$

for $0 \le i \le N-1$. The eq.(29) contains two parts. The former expression reflects the cases that the next arrival sees an *non-empty* queue. The latter one characterizes the transition of the system to a prefetching process that is computed by (28).

We are interested in how efficient the recursive method is. Hence, we present the roadmap to compute $P_i(j,n)$ and its complexity:

- **Step 1:** Solving $P_i(0,2)$, for $i = 1$ to $N-2$;
- **Step 2:** Solving $P_i(0,n)$, for $i = 1$ to $N-2$, and $n = 3$ to $N-x_1+1$ based on *Step 1*;
- **Step 3:** Adding j by 1 and computing $P_i(j,n)$ based on *Step 1* and *Step 2*.

The complexity analysis is carried out from this roadmap. An operation in this recursive algorithm refers to an addition. In **step 1**, the computation of $P_i(0,2)$ incurs $i+2$ additions for each i according to eq.(29). Hence, the total number of additions for all i from 1 to N is around $N^2/2$. **Step 2** computes $P_i(0,n)$ repeatedly for each $n \in \{2, \cdots, N\}$. The **Step 3** repeats **Step 1&2** for each j, but not augmenting the complexity order in N. Therefore, the total complexity has the order $O(N^3)$. The recursive algorithm obtains the starvation probabilities for all j and n, ($j \le J$, $n \le N$) and all initial start-up threshold i ($1 \le i \le N$).

Remark 1: We compare the complexity of the Ballot approach and the recursive approach. First, the standard Ballot approach contains factorial terms (e.g. $\binom{2k-x_1}{k-x_1}$ in eq.(1)) that are of high computational burdens. Second, after Gaussian approximation of factorial terms, the Ballot approach has a complexity order $O(N^2)$ given the file size and the start-up threshold. The recursive approach has a complexity $O(N^3)$ for all combinations of initial start-up threshold i and file size $n(1 \le i, n \le N)$. Thus, the recursive approach has an overall smaller complexity than the Ballot approach.

C. ON/OFF Bursty Traffic

In this section, we model the arrival process as an *interrupted Poisson process (IPP)*, which is commonly used to characterize the bursty and correlated arrivals. The source may stay for relatively long durations in ON and OFF states. The ON/OFF arrival model also has direct applications. For example, the Youtube servers use a simple ON/OFF rate control algorithm to transfer streaming packets

to the users [20]. Our objective is to understand the interaction between the parameters of arrival process and the probability of starvation.
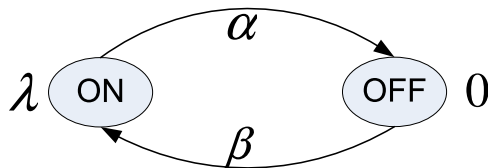

Fig. 2. Two-state Markov process to model bursty traffic

We illustrate the bursty traffic model in figure 2 with the state transition rates α and β. We denote by $Q_i(k)^{ON}$, $0 \le i \le N - 1$, $0 \le k \le i$, the probability that k packets out of i leave the system upon an arrival at the ON state (i.e. no arrival during the OFF period). According to [9], the following proposition holds.

Proposition 1: [9] The probability $Q_i(k)^{ON}$ is expressed as

$$
\begin{aligned}
Q_i(k)^{\mathrm{ON}} &= c_1 \Big(\frac{1}{a_1}\Big)^k + c_2 \Big(\frac{1}{a_2}\Big)^k, \quad 0 \le k \le i - 1, \\
Q_i(i)^{\mathrm{ON}} &= c_1 \frac{(1/a_1)^i}{1 - 1/a_1} + c_2 \frac{(1/a_2)^i}{1 - 1/a_2},
\end{aligned}
\tag{30}
$$

where a_1, a_2, c_1 and c_2 are solved by

$$
\begin{aligned}
\Delta &= (\lambda + \alpha + \beta)^2 - 4\lambda\beta, \\
a_{1,2} &= 1 + \frac{\lambda + \alpha + \beta}{2\mu} \pm \frac{\sqrt{\Delta}}{2\mu}, \\
c_1 &= \frac{\lambda(\beta + \mu) - \lambda\mu a_1}{a_1(a_2 - a_1)}, \quad c_2 = \frac{\lambda(\beta + \mu) - \lambda\mu a_2}{a_2(a_1 - a_2)}.
\end{aligned}
$$

We next show how the starvation probability $P_i(j, n)$ is obtained. The starvation event can happen in both the ON and OFF states. However, the starvation event at the OFF state is equivalent to the event that the first new packet arrival at the ON state sees an empty queue. Therefore, we can use (27) to compute the p.g.f. of starvations with bursty arrivals, simply replacing $Q_i(k)$ by $Q_i^{\mathrm{ON}}(k)$.

Remark 2: The standard Ballot theorem cannot be used to study the starvations of the ON/OFF arrival process. In the presence of bursty traffic, the packet arrival process has two states, ON and OFF. The packet arrivals of two consecutive ON states are separated by an OFF state. Hence, the counting of the arrival arrivals is not equally probable.

V. FLUID MODEL ANALYSIS OF STARVATION PROBABILITY

So far we have studied the starvation behavior of a single file, which is concerned by either the media servers or the users. In fact, the streaming providers are more interested in the QoE evaluation scaled

to a large quantity of videos. They cannot afford the effort of configuring each file a different start-up delay. In this section, we present a fluid analysis of starvation probability, given the distribution of file size.

In the fluid model, the arrival and departure rates are deterministic. We let λ be the number of packet arrivals *per second*, and μ be the number of departures *per second*. Here, μ depends on the encoding rate that the media files use. We focus on the setting $\mu \geq \lambda$ because no starvation will happen with $\mu < \lambda$ in the fluid model. We let x_1 be the start-up threshold. The start-up delay T_1 is simply computed by x_1/λ. Once the media packets are played, the queue length decreases at a rate $\mu - \lambda$. The time needed to empty the queue is thus $\frac{x_1}{\mu-\lambda}$. We let N_p be the total number of packets that are served until a starvation happens,

$$N_p = x_1\big(1 + \frac{\lambda}{\mu - \lambda}\big) = \frac{x_1\mu}{\mu - \lambda}. \tag{31}$$

If the file size is less than N_p, there will be no starvation event.

The distribution of media file size depends on the types of contents. A measurement study in [19] shows the distributions of Youtube video duration for four most popular categories: music, entertainment, comedy and sports videos. The authors find that most of the entertainment, comedy and sports videos are short. They are likely to follow exponential distribution or lognormal distribution with large standard deviations. The lengths of music files in playback time are usually between 180 and 240 seconds (the file size in Bytes is the product of duration and the default bit-rate on Youtube). Hence, with the help of the measurements in [19], we speculate that music video files on Youtube follow lognormal distribution with a small standard deviation. The Pareto distribution adds practical restrictions to the video file size. The file size needs to be greater than a certain value, and a very small fraction of video files can be very large. Today, Youtube allows some users to upload some long movies (more than the previous maximum of 10 min) without copyright issues. The distribution of movie file size may have a heavy-tail. Then, Pareto distribution can serve as a good approximation.

We compare the starvation probabilities of exponential, log-normal and Pareto distributions, given the start-up threshold. Note that these distributions possess the same mean file size. We assume that the users are homogeneous so that λ and μ are the same for different types of file size distributions.

i) *Exponential distribution:* Suppose that the file size N follows an exponential distribution with parameter θ. The probability of starvation, $P_s^{(1)}$, is obtained by

$$P_s^{(1)} = \text{Prob}\ (N > N_p) = \exp(-\frac{\theta x_1\mu}{\mu - \lambda}). \tag{32}$$

ii) *Pareto distribution:* We let N_m be the minimum possible value of the file size, and υ be the exponent in the Pareto distribution. The probability of starvation is computed by

$$P_s^{(2)} = \text{Prob } (N > N_p) = \begin{cases} \left(\frac{N_m(\mu-\lambda)}{\mu x_1}\right)^\upsilon & \forall N_m \leq \frac{x_1\mu}{\mu-\lambda}; \\ 1 & \text{otherwise} , \end{cases} \tag{33}$$

where the expectation of the Pareto distribution is equal to that of the exponential distribution, i.e. $\frac{\upsilon N_m}{\upsilon-1} = \frac{1}{\theta}$.

iii) *Log-Normal distribution:* We suppose that the file size follows a log-normal distribution $\ln \mathcal{N}(\varrho, \sigma)$, where ϱ and σ are the mean and the standard deviation of a natural normal distribution. Given that N_p packets can be served without an interruption, the starvation probability $P_s^{(3)}$ is computed by

$$P_s^{(3)} = \text{Prob } (N > N_p) = \frac{1}{2} - \frac{1}{2}\text{erf}\Big[\frac{\log \frac{x_1\mu}{\mu-\lambda} - \varrho}{\sqrt{2}\sigma}\Big], \tag{34}$$

where its expectation $\exp(\varrho + \frac{\sigma^2}{2})$ equals to $\frac{1}{\theta}$.

Equations (32),(33) and (34) show that the probability of starvation can be controled by setting x_1, if the distribution of file size, the arrival and departure rates are pre-knowledge[1].

VI. APPLICATION TO STREAMING SERVICE

This section presents four scenarios in streaming service in which our analysis can be utilized to optimize the objective QoE. Here, we focus on the M/M/1 system.

The QoE reflects the human perception of the streaming service. A common practice to evaluate QoE is called Mean Opinion Score (MOS). The video watchers give scores according to their subjective opinions. The start-up delay and the starvation behaviors are explicitly defined as quality metrics related to user perception in [27], [28] and [7]. To remove the confusion, we designate the direct human perception as subjective QoE, and designate the objective measure as objective QoE. The human tests implicitly map the objective QoE metrics into a single subjective value. This QoE value, though revealing the user perception statistically, is usually unreliable to report the QoE for each individual watching. At the same time, the subjective test is done after the watching, which cannot be utilized to tune the prefetching online. Therefore, a rising trend is to evaluate the objective QoE metrics and to balance the tradeoff among them (e.g. [27], [28] and [7]).

---

[1]Because the starvation probabilities $P_s^{(1)}$, $P_s^{(2)}$ and $P_s^{(3)}$ take complicated forms, we will compare their dependency on x_1 numerically in section VII. Both Pareto and Log-normal distributions have two parameters. In the comparison, we fix one of them, and solve the other according to the property of identical expectations.

Our purpose is to use content prefetching as a way to achieve the optimal tradeoff between the start-up delay and the starvation behaviors (either the starvation probability or the continuous playback interval) for a user. In [7], the authors configure a start-up threshold to guarantee that the starvation probability is less than a certain value. The bound of the starvation probability is deemed as a parameter obtained from human tests. In this paper, we adopt a more flexible method by defining an objective QoE cost function for a user. A user-defined weight λ is introduced to indicate his/her preference to one type of objective QoE metrics. We first let the starvation probability be one of the QoE metrics. We let $g(\cdot)$ be a strictly increasing but convex function of the expected start-up delay $E[T_1]$. The larger the start-up delay, the higher the QoE cost. The convexity of $g(\cdot)$ means that streaming users are more and more impatient to large start-up delays. We denote by $C_1(x_1)$ the cost of a user watching the media stream,

$$C_1(x_1) = P_s + \gamma g(E(T_1)), \tag{35}$$

where γ is a positive constant. A large γ represents that the users are more sensitive to the start-up delay, and a smaller γ means a higher sensitivity to the starvation. Our goal is to find the optimal start-up threshold x_1^* to minimize $C_1(x_1)$.

The choice of $C_1(x_1)$ should satisfy three basic principles. First, it is convex in x_1 so that only one optimal threshold x_1^* exists. Second, $C_1(x_1)$ is bounded even if ρ is close to 1. Otherwise, the configuration of x_1 is extremely sensitive to ρ. Third, though x_1^* is not required to be a decreasing function of the arrival rate λ, it cannot grow unbounded when λ is large enough. In what follows, we simply let $g(E(T_1)) := (E(T_1))^2 = \left(\frac{x_1}{\lambda}\right)^2$. Note that building a completely convincible cost function is very difficult for a particular user. Even the measurement studies in [27] and [28] only quantify the influence of P_s and $E(T_1)$ statistically and indirectly (using user engagement). We choose $C_1(x_1)$ to be the sum of the measures of two types of objective QoE metrics for two reasons. One is the reflection of QoE tradeoff. The other is its simplicity. Other forms of $C_1(x_1)$ can be optimized in the same way. The more realistic cost function is subject to our future study.

We apply our models to optimize objective QoE in three scenarios: i) finite media streaming, ii) everlasting media streaming and iii) file level. The scenarios i) and ii) are designed for a single stream, while iii) is designed for a large number of streams. When the streaming file has a finite size, the congested bottlenecks such as the 3G base station or the wifi access point can configure or suggest a start-up threshold before the media stream is played. If the steaming file is large enough (e.g. realtime sport channel), a user can measure the arrival/service processes, and then configure the rebuffering threshold. In the third scenario, the media server can set up one same start-up threshold for all the videos that

it distributes. To avoid malfunctions in realistic scenarios, a user can configure the maximum and the minimum start-up/rebuffering delay. Once maximum value is reached, the media player starts to play regardless of the prefetching threshold.

A. Finite Media Size

We hereby consider the adaptive buffering technique for a streaming with finite size. The eq.(1) and eq.(35) yield

$$C_1(x_1) = \sum_{k=x_1}^{N-1} \frac{x_1}{2k - x_1} \binom{2k - x_1}{k - x_1} p^{k-x_1}(1-p)^k + \gamma(\frac{x_1}{\lambda})^2. \tag{36}$$

The starvation probability decreases and the start-up delay increases strictly as x_1 grows. In the objective QoE optimization of finite media size, there does not exist a simple expression of the optimal threshold x_1^*. To find x_1^* numerically, we need to compare the costs from all possible thresholds. The complexity order is low if the binomial distribution in eq.(1) is replaced by the Gaussian distribution. If a user can tolerate up to 1 starvations, P_s will be replaced by the probability $(P_s(0) + P_s(1))$ according to eq.(4).

B. Infinite Media Size

We revisit the user perceived streaming quality in two scenarios: 1) $\rho \geq 1$ and 2) $\rho < 1$.

Case 1: $\rho \geq 1$. The starvation probability converges to a fixed value when the file size approaches infinity. We adopt the same QoE metric as that of the finite media size. Note that P_s can be directly replaced by its asymptotic value in eq.(8). Submitting P_s to $C_1(x_1)$, we have the following cost function

$$C_1(x_1) = \exp\left(\frac{x_1(1 - 2p)}{2pq}\right) + \gamma(\frac{x_1}{\lambda})^2.$$

Letting the derivative $\frac{dC_1}{dx_1}$ be 0, we obtain

$$x_1 \cdot \exp\left(\frac{x_1(2p - 1)}{2pq}\right) = \frac{(2p - 1)\lambda^2}{4\gamma pq}.$$

The optimal threshold x_1^* is solved by

$$x_1^* = LambertW\left(\left(\frac{(2p - 1)\lambda}{2pq}\right)^2 \cdot \frac{1}{2\gamma}\right) \cdot \frac{2pq}{2p - 1},$$

where $LambertW(\cdot)$ is the Lambert W-function.

Case 2: $\rho < 1$. When $\rho < 1$, P_s is 1 for an infinite media size. If we adopt the QoE metric C_1 directly, the optimal start-up delay is always 0. This requires a new objective QoE metric for the case $\rho < 1$.

Since the starvation happens many times, the continuous playback interval can serve as a measure of users' satisfaction. We denote by $C_2(x_1)$ the cost function for an infinite media size with $\rho < 1$,

$$C_2(x_1) := \exp(-\frac{\delta x_1}{\lambda(1-\rho)}) + \gamma(\frac{x_1}{\lambda})^2,$$

where δ is a user defined weighting factor to the expected playback duration ($\delta := 1$ in our numerical examples). We differentiate $C_2(x_1)$ over x_1, and let the derivative be 0, then the optimal start-up/rebuffering threshold is

$$x_1^* = LambertW\big(\frac{\delta^2}{2\gamma(1-\rho)^2}\big) \cdot \frac{\lambda(1-\rho)}{\delta}.$$

C. Optimal Objective QoE in the File Level

Unlike the above QoE optimizations, the threshold x_1 for many files is configured by the media server, instead of the users. The objective is still to balance the tradeoff between the start-up delay and the starvation probability. Here, only the exponentially distributed file size is considered. We choose the cost function $C_1(x_1)$ that yields $C_1(x_1) = \exp(-\frac{\theta x_1 \mu}{\mu - \lambda}) + \gamma\big(\frac{x_1}{\lambda}\big)^2$. The optimal threshold x_1^* can be easily found as

$$x_1^* = LambertW\big((\frac{\theta \mu \lambda}{\mu - \lambda})^2 \cdot \frac{1}{2\gamma}\big) \cdot \frac{\mu - \lambda}{\mu\theta}.$$

VII. NUMERICAL EXAMPLES

A. Starvation of M/M/1 Queue

This set of experiments compares the probability of starvations with the event driven simulations using MATLAB. The M/M/1 queue is tested for up to 5000 times with arrivals from files of different sizes. We deliberately consider four combinations of parameters: $\rho = 0.95$ or 1.1, and $x_1 = 20$ or 40 pkts. The departure rate μ is normalized as 1 if not mentioned explicitly. The choice of the start-up thresholds coincides with the playout of audio or video streaming services in roughly a couple of seconds (e.g. 200$\sim$400kbps playback rate on average given the packet size of 1460 bytes in TCP). The file size in the experiments ranges between 40 and 1000 in terms of packets. Figure 3 displays the probability of 0,1, and 2 starvations with parameters $\rho = 0.95$ and $x_1 = 20$. When the file size grows, the probability of no starvation decreases. We observe that the probabilities of 1 and 2 starvations increase first, and then decline after reaching the maximum values. The reason lies in that the traffic intensity ρ is less than 1. Figure 3 also shows that our analytical results match the simulation well. Figure 4 exhibits the similar results when the start-up threshold is 40 pkts. The comparison between figure 3 and 4 manifests that a larger x_1 is very effective in reducing starvation probability.

Figure 5 plots the probability of no starvation with the traffic intensity $\rho = 1.1$. The probability of no starvation is improved by more than 10% (e.g. $N \geq 300$) when x_1 increases from 20 to 40. Figure 5 also validates the asymptotic probability of no starvation obtained from Gaussian and Riemann integral approximations etc. Figure 6 plots the probability of one starvation with the same parameters. Recall that the probability of one starvation decreases to 0 as N increases in the case $\rho = 0.95$. While figure 6 exhibits a different trend along with the increase of file size. This probability becomes saturated, instead of decreasing to 0. When ρ is greater than 1, the probability of having a particular number of starvations approaches a constant. In both figure 5 and 6, simulation results validate the correctness of our analysis. Hence, in the following experiments, we only illustrate the analytical results.
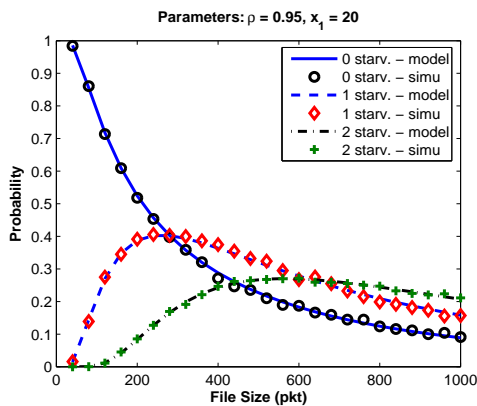
Fig. 3.   Probability of 0, 1, and 2 starvations with $\rho = 0.95$ and $x_1 = 20$
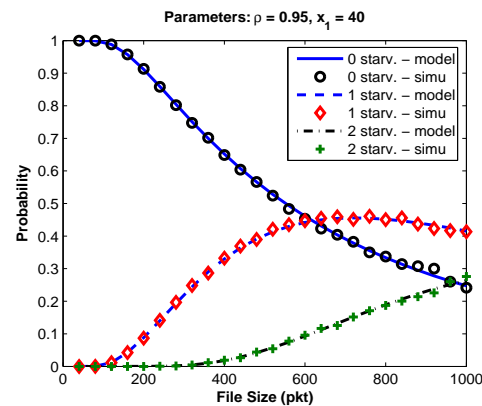
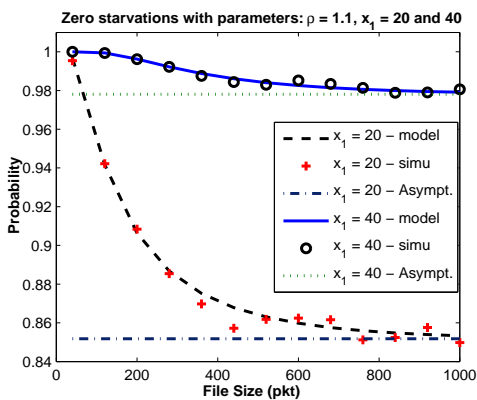Fig. 4.   Probability of 0, 1, and 2 starvations with $\rho = 0.95$ and $x_1 = 40$

Fig. 5.   Probability of no starvation with $\rho = 1.1$: $x_1 = 20$ and $x_1 = 40$
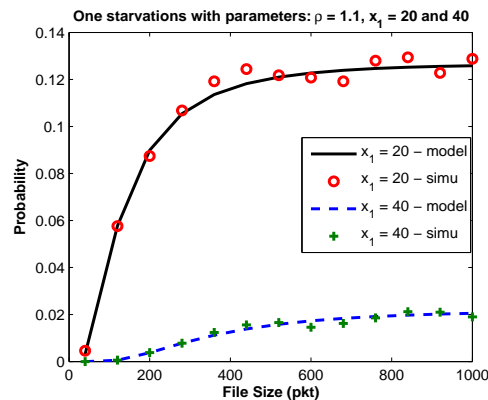
Fig. 6.   Probability of one starvation with $\rho = 1.1$: $x_1 = 20$ and $x_1 = 40$
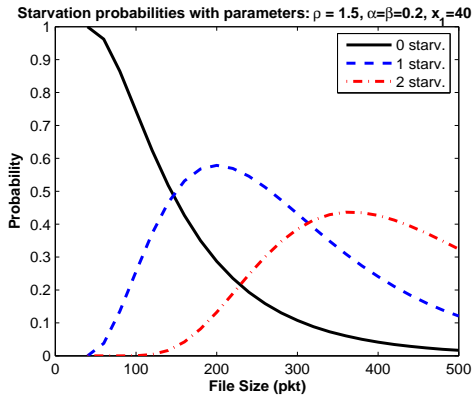
Fig. 7. ON/OFF traffic: probability of 0, 1, and 2 starvations with $\rho = 1.5$ and $x_1 = 40$
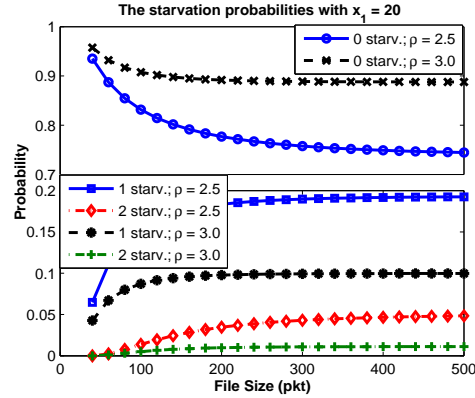
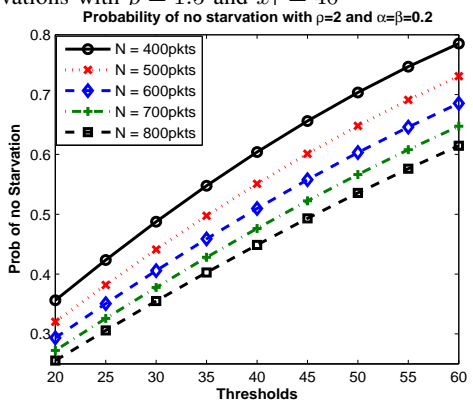Fig. 8. ON/OFF traffic: prob. of 0, 1, and 2 starvations for $x_1 = 20$: $\rho = 2.5$ and 3.0

Fig. 9. ON/OFF traffic: probability of no starvation with $\rho = 2$ versus the threshold x_1
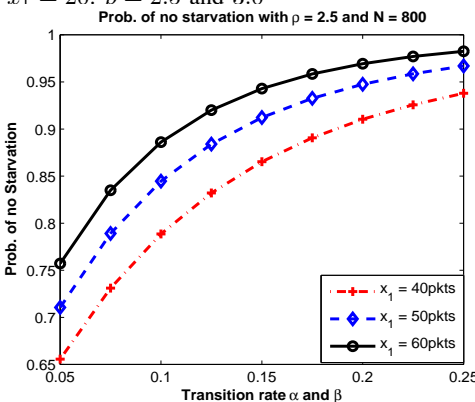
Fig. 10. ON/OFF traffic: probability of no starvation with $\rho = 2.5$ and $N = 800$ versus the state transition rates
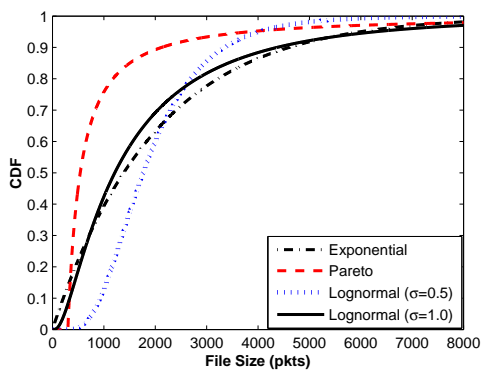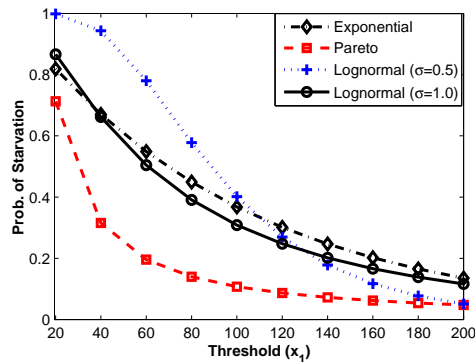

Fig. 11. Fluid analysis: CDF of media file size

Fig. 12. Fluid analysis: prob. of starvation versus the threshold x_1

B. Starvation of Bursty Traffic

We consider the ON/OFF bursty arrival of packets into an M/M/1 queue. For the ease of comparison, we let the transition rates α and β be both 0.2. The file size ranges from 40 to 500 pkts. In figure 7, we

plot the probabilities of having no more than two starvations with $\rho = 1.5$ and $x_1 = 40$. As the file size increases, the probability of no starvation decreases. The probabilities of 1 and 2 starvations increases first, and then decreases to 0. This means that the starvation is for sure when the file size approaches infinity. In figure 8, we plot the starvation probabilities for $\rho = 2.5$ and 3.0 where the start-up threshold is set to 20. In contrast to figure 7, the probability of no starvation converges to a positive constant as N is large enough.

Figure 9 illustrates the impact of x_1 on the probability of no starvation. In this set of experiments, ρ is set to 2. The start-up threshold x_1 increases from 20 to 60 pkts, and the file size increases from 400 to 800 pkts. It is clearly shown that a slight increase in x_1 can greatly improve the starvation probability. In figure 10, we plot the probability of no starvation with $\rho = 2.5$ and $N = 800$ pkts. The transition rates α and β increases from 0.05 to 0.25. It can be seen that the probability of no starvation increases monotonically with the symmetric transition rates α and β.

C. Starvation in the File Level

This set of numerical experiments shows the relationship between the starvation probability and the distribution of file size. The traffic intensity ρ is set to 0.95. We let θ be 1/2000 in the exponential distribution. Then, the average file size is 2000 pkts. This setting is in accord with the recent measurement that most of mobile streaming files are short (i.e. a median size of 1.68MBytes) [30]. For the Pareto distribution, we set the minimum file size to be 300 pkts so that the exponent υ is 1.1765. The parameters of the Log-normal distribution are set to $(\varrho, \sigma) = (7.476, 0.5)$ and $(7.101, 1.0)$ respectively. We plot the CDF curves of the file sizes in Figure 11. In this setting, the Pareto distribution exhibits an obvious heavy tail property. For the log-normal distribution, more than 30% percent of files are less than 1000 packets with $\sigma = 1$ and most of the files are less than 2000 packets with $\sigma = 0.5$. The exponential distribution and the log-normal distribution with $\sigma = 1.0$ exhibit similar CDF of file sizes. We evaluation the starvation probabilities in Figure 12 by increasing the start-up threshold from 20 to 200. The starvation probability of the Pareto distributed files has a sharp reduction in the beginning of increasing x_1. However, as x_1 is more than 80, there is only a slight improvement in the starvation probability. Hence, the streaming service providers need to configure different start-up thresholds for the short files and the tail files in the Pareto distribution. For the log-normal distribution with $\sigma = 0.5$, the starvation probability is high with small x_1. There have significant reductions of starvation probability when x_1 increases from 40 to 140. This is because the file sizes have a small standard deviation. Very small thresholds do not help to reduce the starvation probability and very large thresholds do not further reduce the starvation probability. When

x_1 increases from 20 to 200, we can observe the noticeable reduction of starvation probability in the exponential file size distribution. As the take-home message of fluid analysis, the choice of x_1 relies on the distribution of file size to a great extent. To obtain a better objective QoE, the media service providers can set different x_1 for different categories of media files.

D. Optimizing Quality of Experience

Objective QoE optimization of finite media size:

We illustrate the total QoE cost (including the starvation cost and the start-up delay cost) in figure 13 with $\lambda = 16, 20, 24$ and $\mu = 25$. The file size is set to $N = 1000$ and the weight γ is 10^{-3}. We find that the total QoE looks neither "concave" nor "convex" w.r.t. the start-up threshold. For example, when x_1 is less than 300 with $\lambda = 16$, the increase in the start-up delay cost cannot be compensated by the reduction of the starvation probability. We further plot the optimal start-up threshold obtained from the maximum of eq.(36) in figure 14. When $\gamma = 10^{-4}$ and $\gamma = 10^{-3}$, the optimal start-up threshold x_1^* decreases when λ increases. We also observe that for each λ x_1^* of the case $\gamma = 10^{-4}$ is higher than that of $\gamma = 10^{-3}$ because the former user is more sensitive to the starvation. In the extreme scenario $\gamma = 0$, the streaming user will download the whole media file before watching it. In the case $\gamma = 5 \times 10^{-3}$, x_1^* are always 1 if the arrival rate λ is less than 20. The reason lies in that the total cost is always greater than 1 in those situations. The start-up threshold $x_1^* = 1$ will induce numerous consecutive starvations, which definitely degrades the streaming QoE. To mitigate this malfunction, we can introduce a minimum playback delay that works independent of the QoE optimization.
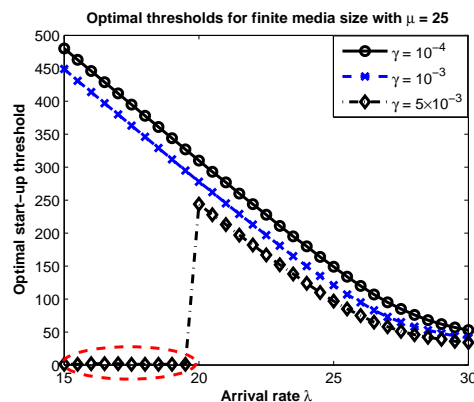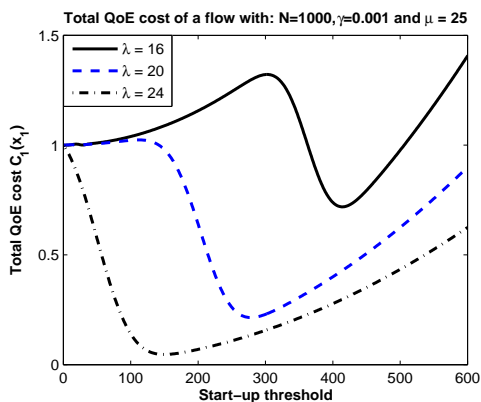

Fig. 13.   Finite media size: total cost with $\mu = 25$ and $\gamma = 0.001$

Fig. 14.   Finite media size: optimal thresholds with $\gamma = 10^{-4}, 10^{-3}$, and 5×10^{-3}

QoE optimization of infinite media size:

We plot the optimal prefetching thresholds x_1 for the case $\rho > 1$ in figure 15 and the case $\rho < 1$ in figure 16. As λ increases, the optimal prefetching threshold x_1^* reduces. Unlike figure 14, there do not exist an abrupt change in x_1^*. This is because the cost function $C_1(x_1)$ is a convex function of x_1 with both $\rho > 1$ and $\rho < 1$. Furthermore, x_1^* decreases as γ increases (the user putting more weight to the prefetching delay).
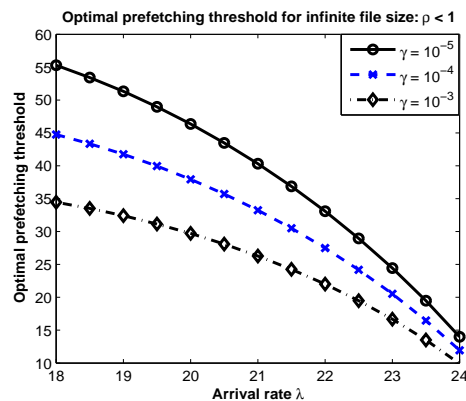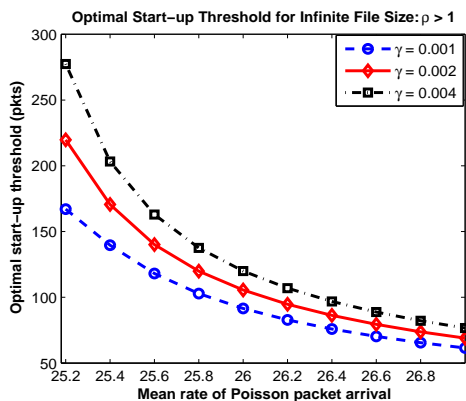

Fig. 15.   Optimal threshold x_1^* of infinite file size: $\rho > 1$ Fig. 16.   Optimal threshold x_1^* of infinite file size: $\rho < 1$

QoE optimization in the flow level:

We investigate the cost minimization problem at the media server side numerically. We let $\mu := 25$ which means that 25 packets are served per second. Given the packet size of 1460 bytes, this service rate is equivalent to 292Kbps (without considering protocol overheads). We let the mean file size $1/\theta$ be 1000 and 2000 packets respectively (equivalent to the playback time of 40 and 80 seconds). The sensitivity γ is set to 0.01 or 0.005. Figure 17 illustrates the choice of the optimal start-up thresholds when λ increases from 20 to 25 (i.e. $\rho \leq 1$). We evaluate four combinations of θ and γ numerically. Our observations are summarized as follows. First, for the same file size distribution, a smaller γ causes a higher optimal start-up threshold. Second, x_1^* is not a strictly decreasing function of λ. When λ is small (e.g. 20pkts/s), a large start-up threshold does not help much in reducing the starvation probability, but causes impatience of users of waiting the end of prefetching. If λ increases, the adverse impact of setting a larger x_1 on the start-up delay can be compensated by the gain in the reduction of starvation probability. Third, with the same sensitivity γ, the optimal x_1^* of a long video stream can be smaller than that of a short one in some situations. This is caused by the fact that the large threshold might not significantly improve the starvation probability for a file of large size. We further show the starvation probability in figure 18. A larger mean file size, or a smaller γ result in a larger probability of starvation. Unlike the start-up threshold x_1^*, the starvation probability is shown to be strictly decreasing as λ increases.
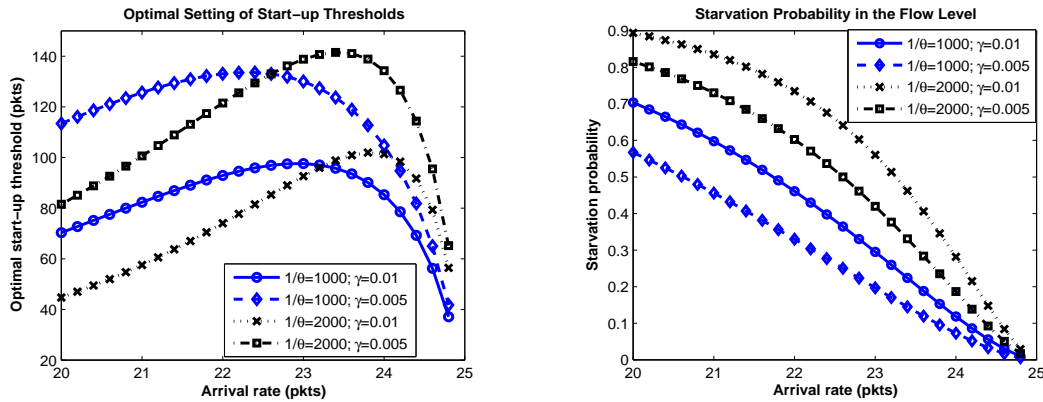
Fig. 17. Optimal threshold x_1^* for QoE enhancement at the file level: $\mu = 25$

Fig. 18. Starvation probability at the file level for the optimal start-up threshold x_1^*: $\mu = 25$

VIII. CONCLUSION AND DISCUSSION

We have conducted an *exact* analysis of the starvation behavior in Markovian queues with a finite number of packet arrivals. We perform a packet level analysis and a fluid level analysis. The packet level study is carried out via two approaches, the Ballot theorem and the recursive equations. The former provides an explicit expression, but is usually limited to i.i.d. packet arrivals. The latter can handle bursty packet arrival process, but without an explicit result. From the perspective of a media service provider, we perform a fluid level analysis that computes the probability of starvation for a large number of video files. We further apply the theoretical results to tune the prefetching thresholds in order to optimize the objective QoE for media streaming services.

REFERENCES

[1] L. Takacs, "Ballot problems", *Prob. Theory Related Fields*, Vol. 1, No.2, pp:154-158, 1962.

[2] F. Baccelli and W.A. Massey, "A Sample Path Analysis of the M/M/1 Queue", *Journal of Applied Probability*, Vol.26, No.2, pp:418-422, 1989.

[3] W. Ledermann and G. Reuter, "Spectral Theory for the Differential Equations of Simple Birth and Death Processes", *Phi. Trans. Roy. Soc. London*, Vol.246, No.914, pp:321-369, 1954.

[4] L. Liu and D.H. Shi, "Busy period in GI(X)/G/∞", *J. Appl. Prob.*, Vol.33, pp:815-829, 1996.

[5] Hao Luan, Lin X. Cai, and Xuemin (Sherman) Shen, "Impact of network dynamics on users' video quality: analytical framework and QoS provision" *IEEE Trans. on Multimedia*, Vol.12, No.1, pp:64-78, 2010.

[6] G. Liang and B. Liang, "Effect of delay and buffering on jitter-free streaming over random VBR channels", *IEEE Trans. on Multimedia*, Vol.10, No.6 pp:1128-1141, 2008.

[7] A. ParandehGheibi, M. Medard, A. Ozdaglar, S. Shakkottai, "Avoiding Interruptions a QoE Reliability Function for Streaming Media Applications", *IEEE Journal on Selected Area in Communications*, Vol.29, No.5, pp:1064-1074, 2011.

[8] A. Papoulis, "Probability, Random Variables and Stochastic Processes", McGraw-Hill Publisher, pp:360-361, 1984.

[9] I. Citon, A. Khamisy, and M. Sidi, "Analysis of packet loss processes in high-speed networks", *IEEE Trans. Info. Theory*, Vol.39, No.1, 1993.

[10] E. Altman, A. Jean-Marie, "Loss probabilities for messages with redundant packets feeding a finite buffer", *IEEE J. Sel. Area. Comm.*, Vo.16, No.5, pp:778-787, 1998.

[11] E. Altman, A. Jean-Marie, "The distribution of delays of dispersed messages in an M/M/1 queue", *Proc. IEEE Infocom*, Boston, 1995.

[12] P. Dubea, O. Ait-Hellal, E. Altman, "On loss probabilities in presence of redundant packets with random drop", *Elsevier Perf. Eval.*, Vol.53, pp:147-167, 2003.

[13] P. Humblet, A. Bhargava, M.G. Hluchyj, "Ballot theorems applied to the transient analysis of nD/D/1 queues", *IEEE Trans. Networking*, Vol.1, No.1, pp:81-95, 1993.

[14] O. Gurewitz, M. Sidi, I. Cidon, "The Ballot Theorem Strikes Again: Packet Loss Process Distribution ", *IEEE Trans. Info. Theory*, Vol.46, No.7, 2000.

[15] T. Stockhammer, H. Jenkac, and G. Kuhn, "Streaming video over variable bit-rate wireless channels," *IEEE Trans. Multimedia*, Vol.6, No.2, pp:268-277, 2002.

[16] J.F. He and K. Sohraby, "New Analysis Framework for Discrete Time Queueing Systems with General Stochastic Sources", *Proc. of IEEE Infocom 2001*, pp:1075-1084, Anchorage, 2001.

[17] A.Y. Privalova and K. Sohraby, "Playout in Slotted CBR Networks: Single and Multiple Nodes", *Problems of Information Transmission*, Vol.43, No.2, pp:143-166, 2007.

[18] http://techcrunch.com/2010/11/19/web-video-37-percent-internet-traffic/

[19] X. Cheng, C. Dale, J.C. Liu, "Statistics and Social Network of YouTube Videos", *Proc. of IEEE IWQoS*, pp:229-238, Enschede, 2008

[20] S. Alcock, R. Nelson, "Application flow control in YouTube video streams", *ACM Comp. Commun. Review*, Vol.41, No.2, pp:25-30, 2011.

[21] Y.D. Xu, X.X. Wu, J.C.S. Lui, "Cross-Layer Qos Scheduling for Layered Multicast Streaming in OFDMA Wireless Networks", *Wireless Pers. Commun.*, Vol.51, No.3, pp:565-591, 2009.

[22] Y.D. Xu, E. Altman, et al., "Probabilistic Analysis of Buffer Starvation in Markovian Queues", *Proc. of IEEE Infocom 2012*. Technical report available at http://arxiv.org/abs/1108.0187

[23] Y.D. Xu, E. Altman, R. Elazouzi, S.E. Elayoubi and M. Haddad. "QoE Analysis of Media Streaming in Wireless Data Networks", *Proc. of IFIP Networking 2012*.

[24] Y.D. Xu, S.E. Elayoubi, E. Altman and R. Elazouzi. "Impact of flow-level dynamics on QoE of video streaming in wireless networks", *Proc. of IEEE Infocom 2013*.

[25] M. Abramowitz and I.Stegun (Ed.), "Handbook of Mathematical Functions", 1970.

[26] G.E. Box, W.G. Hunter and J.S. Hunter, "Statistics for Experimenters". Wiley Publisher. pp:130, 1978.

[27] F. Dobrian, A. Awan, I. Stoica, V. Sekar, A. Ganjam, D. Joseph, J. Zhan, and H. Zhang, "Understanding the Impact of Video Quality on User Engagement", *Proc. of ACM Sigcomm 2011*, Vol.41, No.4, pp:362-373.

[28] S. Krishnan and R.K. Sitaraman, "Video Stream Quality Impacts Viewer Behavior: Inferring Causality using Quasi-Experimental Designs", *ACM/USENIX Internet Measurement Conference*, 2012.

[29] G.F. Golub and C.F. van Loan. *Matrix Computations*, Johns Hopkins University Press, 1986.

[30] Y. Liu, F. Li, L. Guo, B. Shen and S. Chen. "A Servers Perspective of Internet Streaming Deliver to Mobile Devices", *Proc. of IEEE Infocom 2012*.
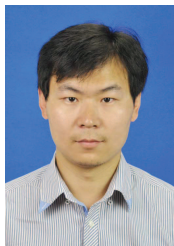
Yuedong Xu is a tenure-track associate professor in the School of Information Science and Technology, Fudan University, China. He received his PhD from The Chinese University of Hong Kong, M.S. degree from Huazhong University of Science and Technology and B.S. degree from Anhui University. From late 2009 to 2012, he was a postdoc with INRIA Sophia Antipolis and Universite d'Avignon, France. His areas of interest include performance evaluation, control, optimization, and economic analysis of communication networks.

Eitan Altman received the B.Sc. degree in electrical engineering (1984), the B.A. degree in physics (1984) and the Ph.D. degree in electrical engineering (1990), all from the Technion-Israel Institute, Haifa. In (1990) he further received his B.Mus. degree in music composition in Tel-Aviv university. Since 1990, Dr. Altman has been a researcher at INRIA (National research institute in computer science and control) in Sophia-Antipolis, France. He received the best paper award in the Networking 2006, in Globecom 2007, in IFIP Wireless Days 2009 and in CNSM 2011 (Paris) conferences. His areas of interest include Network Engineering Games, social networks and their control. He became an IEEE Fellow in 2010. He received in 2012 the Grand Prix de France Telecom from the French Academy of Sciences.

Rachid Elazouzi received the Ph.D. degree in Applied Mathematics from the Mohammed V University, Morocco (2000). He joined INRIA Sophia Antipolis for post-doctoral and Research Engineer positions. Since 2003, he has been a researcher at the University of Avignon, France. He received the best paper award in the Networking 2006, MSN 2007 and Globcom 2007 conferences. His research interests are networking games, performance evaluation, learning algorithms, radio resource management and epidemic networks.

Majed Haddad received his B.S. degree (2004) from the National Engineering School of Tunis (ENIT), Tunisia. He received the M.S. degree (2005) from Polytech Nice Sophia in France, and the PhD degree (2008) from Eurecom Institute. From 2009 to 2011, he was a postdoc fellow with France Telecom R&D and INRIA Sophia-Antipolis. Dr. Haddad is currently a research engineer at INRIA Sophia-Antipolis in France under an INRIA-ALCATEL-LUCENT Bell-labs fellowship where his main research interests are in radio resource management, heterogeneous networks, green networks, complex networks and game theory.

Salah Eddine Elayoubi received his M.S. degree from the National Polytechnic Institute at Toulouse, France, in 2001, and his Ph.D. and Habilitation in Computer science from the University of Paris VI in 2004 and 2009, respectively. Since then, he has been working in Orange Labs, the research and development division of France Telecom group. He is now a senior research engineer, involved in several EU-funded research projects. His research interests include radio resource management, modeling and performance evaluation of mobile networks.

Tania Jimenez received her Ph.D. from University of Nice Sophia-Antipolis, France in 2000. Her research interests include simulation as well as optimization and control of telecommunication networks. She is at present a research engineer at Avignon University, in the Informatics Lab.

Notation	Definitions
	Section III
λ	Packet arrival rate
μ	Packet service rate
ρ	λ/μ
$p,\ q$	$p = \lambda/(\lambda + \mu),\ q = \mu/(\lambda + \mu)$
x_1	Start-up threshold in pkts
T_1	Start-up delay
N	File size in pkts
P_s	Probability of starvation
$P_s(j)$	Probability of meeting j starvations
$\mathcal{E}(k_1)$	First empty buffer after the service of k_1 pkts
$\mathcal{S}_l(k_l, k_{l+1})$	Empty buffer after the service of pkt k_{l+1} given that the previous emptiness happens at the departure of pkt k_l
$\mathcal{U}_j(k_j)$	Last empty buffer observed after departure of packet k_j
$P_{\mathcal{E}(k_1)}$	Probability of $\mathcal{E}(k_1)$
J	Maximum number of starvations, $J = \lfloor \frac{N}{x_1} \rfloor$
d	Duration of a service slot in M/D/1 queue
	Section IV
$P_i(n)$	Starvation probability of a file with n packets, given that there are i packets in the buffer
$Q_i(k)$	Probability that k pkts out of i leave the system during an inter-arrival period
$P_i(j, n)$	Probability of j starvations with a file of size n, given that the first pkt sees i pkts already there
α	Transition rate from ON to OFF
β	Transition rate from OFF to ON
$Q_i(k)^{ON}$	Probability that k pkts out of i leave the system during an inter-arrival period at ON state
	Section V
N_p	Total number of packets that are served
$1/\theta$	Mean of exponential file size distribution
N_m	Minimum file size for Pareto distribution
υ	Exponent for Pareto distribution
ϱ	Mean of Normal distribtion for log-normal
σ	Standard deviation of Normal distribtion for log-normal
	Section VI
$C_1(x_1)$	QoE cost function for general situations
$C_2(x_1)$	QoE cost function for infinite media size with $\rho < 1$
$LambertW$	Lambert W-function

TABLE I

GLOSSARY OF MAIN NOTATION