Probabilistic Analysis of Buffer Starvation in Markovian Queues

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

Probabilistic Analysis of Buffer Starvation in Markovian Queues

Yuedong Xu†, Eitan Altman‡, Rachid El-Azouzi†, Salaheddine Elayoubi*, Majed Haddad‡, Tania Jimenez†
†University of Avignon, 339 Chemin des Meinajaries, Avignon, France
‡INRIA Sophia Antipolis, 2004 Route des Lucioles, France
*Orange Labs, Paris, France
Email: yuedong.xu@gmail.com, eitan.altman@inria.fr, Rachid.Elazouzi@univ-avignon.fr,
majed.haddad@inria.fr, salaheddine.elayoubi@orange-ftgroup.com, Tania.Jimenez@univ-avignon.fr

***Abstract*—Our purpose in this paper is to obtain the *exact distribution* of the number of buffer starvations within a sequence of N consecutive packet arrivals. The buffer is modeled as an M/M/1 queue.When the buffer is empty, the service restarts after a certain amount of packets are *prefetched*. With this goal, we propose two approaches, one of which is based on *Ballot theorem*, and the other uses recursive equations. The Ballot theorem approach gives an explicit solution, but at the cost of the high complexity order in certain circumstances. The recursive approach, though not offering an explicit result, needs fewer computations. We further propose a fluid analysis of starvation probability on the file level, given the distribution of file size and the traffic intensity. The starvation probabilities of this paper have many potential applications. We apply them to optimize the quality of experience (QoE) of media streaming service, by exploiting the tradeoff between the start-up delay and the starvation.**

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 real time voice as well as 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 propose two approaches (that give the same result) to compute the starvation probabilities and the distribution of the number of starvations for a single file. The first approach gives an explicit result based on the *Ballot theorem* [1]. The second approach provides a recursive computation algorithm. Both are done in an M/M/1 queue on a *packet level*. Using Ballot theorem, we can compute in a simple way the exact distribution of the number of starvations explicitly. As the file size approaches infinity, we present the asymptotic starvation probability using Gaussian (interchangeable with Normal) approximation as well as an approximation of the Riemann integral. Whereas the Ballot Theorem provides an explicit solution, we propose an alternative approach which constitutes a recursive algorithm for computing starvation probabilities. Although the recursive approach does not generate an explicit solution, it does perform better than the Ballot Theorem in terms of complexity under certain circumstances.

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 delay, while a shorter one might result in starvations. The user perceived media quality (or QoE equivalently) is impaired by either the large start-up delay or the undesirable starvations. This problem 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? We propose a set of QoE metrics for both the finite and the infinite file size. The optimal QoE is achieved by configuring the start-up threshold in packets. Recently, the similar QoE issue is studied in the important works [5], [6], [7]. Liang et al. [6] studies the bounds of start-up delay, given the *deterministic* playout and arrival curves. Authors in [7] present a minimum prefetching threshold for an M/D/1 queue, other than an exact solution. They further extend their method to consider the arrival process depicted by a two-state Markov chain. Luan et al. [5] adopts diffusion approximation to investigate the time-dependent starvation behavior. Their technique is inadequate to provide insights on starvation in a media file with small number of units (in packets or chunks). Compared to state of the art, our approaches target at the exact solution, and can analyze the starvation of small files. This is particularly important in the evaluation of adaptive streaming where the entire media file is subdivided into many chunks encoded by multiple playback rates. The starvation is more likely to happen when the packets from one or several high-definition video chunks are being played.

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. 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 and discusses the future work.

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. The exact result is expressed as an infinite sum of modified Bessel functions. The starvation analysis of this work is different from the transient queueing analysis in two aspects. First, the former aims to find the probability generating function of starvation events while not the queue size. Second, the former does not assume a stationary arrival process.

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 with batch arrivals and departures. 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 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. In the application field, our paper differs from state of the art works in the following ways: i) we present new theories that yield an *exact* probability of starvation, and the probability generating function of starvation events; ii) we study of 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 queue with finite number of arrivals. The analytical method is 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. 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 protocols. 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. Denote by λ the Poisson arrival rate of the packets, and by μ the Poisson service rate. 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. Let T_1 be the start-up delay, in which x_1 packets are accumulated in the buffer. Once 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: Before proving this theorem, we iterate the classical Ballot theorem first.

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 $\frac{N_A-N_B}{N_A+N_B}$.*

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 seen by packet N is not a starvation. When the buffer is empty at the end of the service of the k^{th} 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. $m \le 2k-x_1$), the number of packets that have been played is always greater than the number of arrivals. Otherwise, the empty buffer already happens before the k^{th} 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). ∎

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. Denote by k_l the l^{th} 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

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

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, and cannot happen after all N packets have been served. 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. 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

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

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 $\mathbf{P}_{\mathcal{E}}$ 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 $i = 1, 2, \cdots, N$.

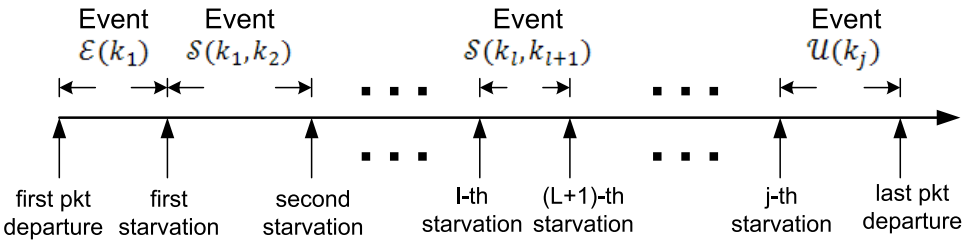

Fig. 1. A path with j starvations

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 l^{th} 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 $(l+1)^{th}$ cannot happen at $k_{l+1} \in [k_l + 1, k_l + x_1 - 1]$. 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 l^{th} 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

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

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

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

Since the starvation event takes non-negative integer values, we can write the probability generating function $G(z)$ by

$$G(z) = E(z^j) = \sum_{j=0}^{J} P_s(j) z^j = \mathbf{P}_{\mathcal{E}} \Big(\prod_{l=1}^{j-1} \mathbf{P}_{\mathcal{S}_l} \Big) \mathbf{P}_{\mathcal{U}_j}^T \cdot z^j. \quad (7)$$

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 Appendix in the technical report [22]). The Gaussian approximation significantly reduces the computational complexity of binomial distributions. The approximated probability of no starvation computed by the complement of eq.(1) has a complexity $O(N)$ obviously. The probability of having only one starvation is a product of two vectors, which also yields a complexity $O(N)$. 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 $\mathbf{O}(N^3)$ so that the direct computation of eq.(7) is extremely difficult for large N. Recall that $\mathbf{P}_{\mathcal{S}_l}$ satisfies i) an upper triangle matrix, ii) first $lx_1 - 1$ rows being 0, iii) last x_1 rows being 0 and iv) $\mathbf{P}_{\mathcal{S}_l}(k_l, k_{l+1}) = \mathbf{P}_{\mathcal{S}_l}(k_l + 1, k_{l+1} + 1)$ if they are not zero. These properties facilitate us to compute the product of the upper triangle matrices with much less effort. Due to the properties i) and iv), the product of two upper triangle matrices has a complexity order $O(N^2)$. Detailed analysis is provided in the Appendix. When there are $j (j \geq 3)$ starvations, the number of matrix production is $j-2$, resulting in a complexity order $O(N^{2(j-2)})$ for multiplying all the matrices. To obtain $P_s(j)$, we still need to compute the product of the vector $P_{\mathcal{E}(k_1)}$ and the matrix. To sum up, the total complexity is $O(N^{2j-2})$ for $j \geq 2$.

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

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

The detailed analysis can be found in the Appendix of [22].

The asymptotic analysis reveals that the probability of starvation has nothing to do with 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. 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.

$$E[T_s] = \frac{x_1}{\lambda(1-\rho)}. \quad (9)$$

IV. Starvation Analysis Via a Recursive Approach

In this section, we present a recursive approach to compute the starvation probability based on [9]. Compared with the one using Ballot theorem, the recursive approach has less computational complexity, though without an explicit expression.

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 $n = N - x_1$ and $i = x_1 - 1$. *Here, the expression $i = x_1 - 1$ means that the service starts when the packet x_1 sees $x_1 - 1$ packets accumulated in the buffer.* 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

$$Q_i(k) = \rho\big(\frac{1}{1+\rho}\big)^{k+1} = pq^k, \quad 0 \le k \le i-1, \quad (10)$$

$$Q_i(i) = \big(\frac{1}{1+\rho}\big)^{i} = q^i. \quad (11)$$

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

$$P_i(1) = 0, \quad \forall i \ge 1. \quad (12)$$

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

$$P_0(n) = 1, \quad \forall n. \quad (13)$$

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

$$P_i(n) = \sum_{k=0}^{i+1} Q_{i+1}(k) P_{i+1-k}(n-1), \quad 0 \le i \le N-1. \quad (14)$$

We explain (14) 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 (14) 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 $n = N - x_1$.

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

$$P_i(j,1) = \begin{cases} 0 & \forall i = 1, 2, \cdots, N-1, \text{ and } j \ge 1; \\ 1 & \forall i = 1, 2, \cdots, N-1, \text{ and } j = 0, \end{cases} \quad (15)$$

and

$$P_0(j,1) = \begin{cases} 0 & j = 0 \text{ or } j \ge 2; \\ 1 & j = 1. \end{cases} \quad (16)$$

The equation (15) means that the probability of no starvation is 1 conditioned by $i \ge 1$ 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 (16) 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

$$P_i(j, n) = 0, \quad \text{if } i + n \ge N \quad (17)$$

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

$$P_0(0, n) = 0, \quad \forall\, 1 \le n \le N - b; \quad (18)$$

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 only one starvation event exists, there has

$$P_0(1, n) = 1, \quad \forall\, 1 \le n \le b, \quad (19)$$

where $b := x_1 - 1$ is denoted to be the prefetching threshold. 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,

$$P_0(1, n) = P_b(0, n - b), \quad \forall\, b < n \le N - b. \quad (20)$$

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

$$P_0(j, n) = 0, \quad \forall\, j > 1 \text{ and } 1 \le n \le b. \quad (21)$$

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.

$$P_0(j, n) = P_b(j - 1, n - b), \quad \forall\, j > 1 \text{ and } n > b. \quad (22)$$

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:

$$\begin{aligned} P_i(j, n) &= \sum_{k=0}^{i+1} Q_{i+1}(k) P_{i+1-k}(j, n-1), \\ &= \sum_{k=0}^{i} pq^k P_{i+1-k}(j, n-1) + q^{i+1} P_0(j-1, n-1), \end{aligned} \quad (23)$$

for $0 \le i \le N-1$. The eq.(23) 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.

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. In **step 1**, the computation of $P_i(0,2)$ incurs up to N summations for each i, resulting in at most N^2 sums in total. The **Step 2** compute $P_i(0,n)$ repeatedly for each n and the **Step 3** repeats **Step 1&2** for each j. Therefore, the total complexity has a order $O\big((j+1)N^3\big)$.

Remark 1: The complexity orders of the Ballot approach with Gaussian approximation and the recursive approach are $O\big(N^{2j-2}\big)$ for $j \geq 2$ and $O\big((j+1)N^3\big)$ respectively. When $j \geq 3$, the recursive approach may have less computational complexity than the Ballot approach.

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 media servers are more interested in the QoE evaluation scaled to a large quantity of files they supply. 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. 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 $\frac{x_1}{\mu-\lambda}$. Let N_p be the total number of packets that are served until a starvation happens,

$$N_p = x_1\big(1+\frac{\lambda}{\mu-\lambda}\big) = \frac{x_1\mu}{\mu-\lambda}. \tag{24}$$

If the size of a file 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] reveals that the music, entertainment, comedy and sports videos have different distributions of file size. In this section, we compare the starvation probability of several commonly used distributions, given the start-up threshold. Note that these distributions possess the same mean file size. We further 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

$$P_s^{(1)} = \text{Prob } (N > N_p) = \exp(-\frac{\theta x_1 \mu}{\mu-\lambda}). \tag{25}$$

ii) *Pareto distribution:* It is frequently adopted to model the file size distribution of Internet traffic using TCP protocol. 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

$$P_s^{(2)} = \text{Prob } (N > N_p) = \begin{cases} \big(\frac{N_m(\mu-\lambda)}{\mu x_1}\big)^{\upsilon} & \forall N_m \leq \frac{x_1\mu}{\mu-\lambda}; \\ 1 & \text{otherwise }, \end{cases} \tag{26}$$

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

$$P_s^{(3)} = \text{Prob } (N > N_p) = \frac{1}{2} - \frac{1}{2}\text{erf}\big[\frac{\log\frac{x_1\mu}{\mu-\lambda} - \varrho}{\sqrt{2}\sigma}\big], \tag{27}$$

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

Equations (25),(26) and (27) 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-like Service

This section presents three scenarios in streaming-like service in which our analyses can be utilized to optimize the quality of experience. Here, we focus on the M/M/1 system.

The cost of a user reflects the tradeoff between the start-up delay and the starvation behaviors (either the starvation probability or the continuous playback interval). We first let the starvation probability be one of the QoE metrics. Let $g(\cdot)$ be a strictly increasing but convex function of the expected start-up delay $E[T_1]$. We denote by $C_1(x_1)$ the cost of a user watching the media stream,

$$C_1(x_1) = P_s + \gamma g(E(T_1)), \tag{28}$$

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 purpose 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 $g(E(T_1)) := (E(T_1))^2 = \big(\frac{x_1}{\lambda}\big)^2$.

[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.

We apply our models to optimize 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 delay locally. In the third scenario, the media server can set up one start-up threshold for all the streams that it distributes. To avoid malfunctions in realistic scenarios, a user can configure lower and upper bounds for the start-up delay. Once the upper bound 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 stream with finite size. The eq.(1) and eq.(28) yield

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

The starvation probability decreases and the start-up delay increases strictly as x_1 grows. In the 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 using the binary search method. 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

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

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

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

The optimal threshold x_1^* is solved by

$$x_1^* = LambertW\big((\frac{(2p-1)\lambda}{2pq})^2 \cdot \frac{1}{2\gamma}\big) \cdot \frac{2pq}{2p-1}, \tag{29}$$

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 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. We differentiate $C_2(x_1)$ over x_1, and let the derivative be 0, then the optimal start-up threshold is

$$x_1^* = LambertW\big(\frac{\delta^2}{2\gamma(1-\rho)^2}\big) \cdot \frac{\lambda(1-\rho)}{\delta}. \tag{30}$$

C. Optimal 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

$$x_1^* = LambertW\big((\frac{\theta\mu\lambda}{\mu-\lambda})^2 \cdot \frac{1}{2\gamma}\big) \cdot \frac{\mu-\lambda}{\mu\theta}. \tag{31}$$

VII. Numerical Examples

A. Starvation of M/M/1 Queue

This set of experiments compare the probability of starvations with the event driven simulations using MATLAB. We simulate up to 5000 samples of the M/M/1 queue 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∼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 2 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 2 also shows that our analytical results match the simulation well. Figure 3 exhibits the similar results when the start-up threshold is 40 pkts. The comparison between figure 2 and 3 manifests that a larger x_1 is very effective in reducing starvation probability.

Figure 4 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 4 also validates the asymptotic probability of no starvation obtained from Gaussian and Riemann integral approximations etc. Figure 5 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 5 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 4 and 5, simulation results validate the correctness of our analysis. Hence, in the following experiments, we only illustrate the analytical results.

B. Starvation in the File Level

This set of numerical experiments show the relationship between the starvation probability and the distribution of file size. The traffic intensity ρ is set to 0.95. Let θ be 1/2000 in the exponential distribution. Then, the average file size is 2000 pkts. For the Pareto distribution, we set the minimum file size to be 300 pkts so that the exponent υ is 1.1765. The parameters ϱ and σ of the Log-normal distribution are set to 5.0 and 2.2807. We plot the CDF curves of the file size and the starvation probabilities in figure 6. The left-side subfigure illustrates the distribution of file size with the parameters configured above. The Pareto and the Log-normal distributions exhibit heave-tail property. In the right-side subfigure, we plot the starvation probability of different file distributions when x_1 increases from 10 to 150. The starvation probability of the Pareto distribution is very high with small x_1. This is because the files have a minimum size (i.e. $N_p < N_m$). The log-normal distribution demonstrates a small starvation probability. In addition, increasing the threshold x_1 does not having a significant impact on the starvation probability when x_1 is greater than 90. Therefore, as the take-home message of fluid analysis, the configuration of x_1 relies on the distribution of file size to a great extent. To obtain a better QoE, the media servers can set different x_1 for different classes of media files.

C. QoE Optimization in the File Level

We investigate the cost minimization problem at the media server side numerically. 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). 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 7 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.

VIII. Conclusion, Discussion and Future Work

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. Both of them have pros and cons; the former providing an explicit expression, but with high complexity order in general, while the latter is more computationally efficient, but without an explicit result. In order to analyze the behavior from a media service provider's point of view, we perform a fluid level analysis that computes the probability of starvation among many files. We further apply the theoretical results to perform QoE optimization for media streaming services. Our work can be extended to study the QoE metrics in a more general network with multiple bottlenecks between the server and the user. In this situation, the arrival process can be modeled as a phase-type renewal process.

In terms of future works, we aim at extending the analytical methods to perform QoE optimization in adaptive streaming services. Another important extension is the starvation analysis in a wireless environment where the wireless link is shared by multiple connections. In such a case, the arrival rate to a user is time varying due to the arrivals and departures of other calls.

Acknowledgements: The work of the authors from INRIA and from Univ of Avignon was supported by a contract with Orange Lab, Issy Les Moulineaux.

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.

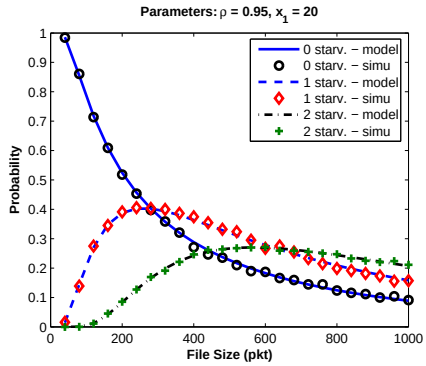

Fig. 2. Probability of 0, 1, and 2 starvations with $\rho = 0.95$ and $x_1 = 20$

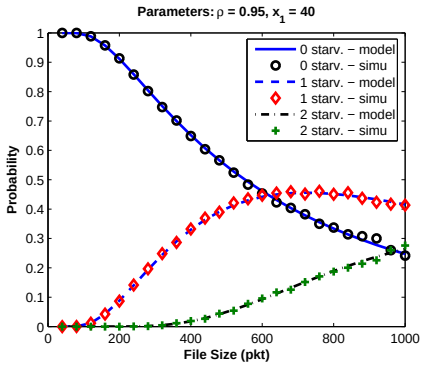

Fig. 3. Probability of 0, 1, and 2 starvations with $\rho = 0.95$ and $x_1 = 40$

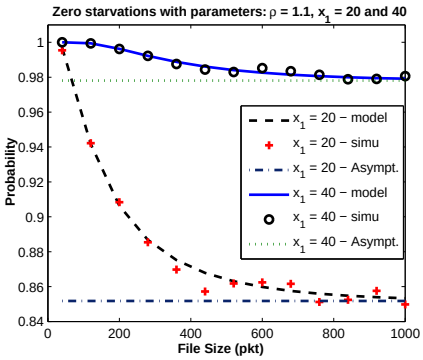

Fig. 4. Probability of no starvation with $\rho = 1.1$: $x_1 = 20$ and $x_1 = 40$

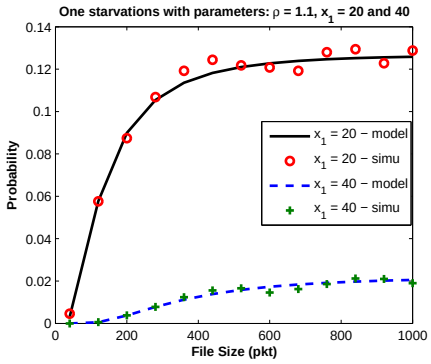

Fig. 5. Probability of one starvation with $\rho = 1.1$: $x_1 = 20$ and $x_1 = 40$

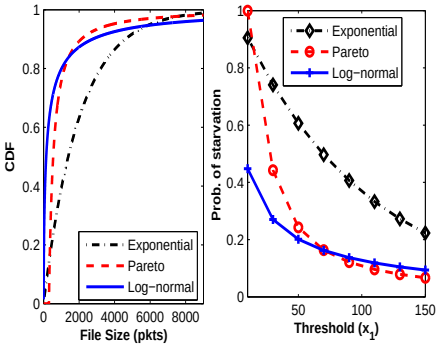

Fig. 6. Fluid analysis: prob. of starvation versus the threshold x_1

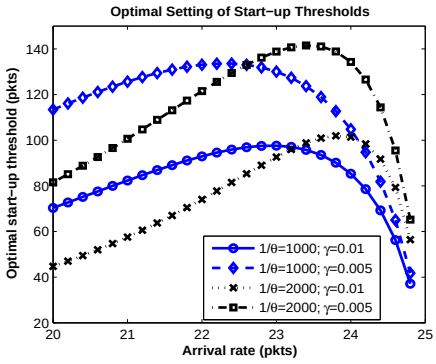

Fig. 7. Optimal threshold x_1^* for QoE enhancement at the file level: $\mu = 25$ pkts/s

[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", Technical report http://arxiv.org/abs/1108.0187

APPENDIX

A. *Complexity Analysis of* (6)

We focus on the cases with more than two starvations ($j \geq 2$). Recall that $\mathbf{P}_{\mathcal{S}_l}$ satisfies i) an upper triangle matrix, ii) first $lx_1 - 1$ rows being 0, iii) last x_1 rows being 0 and iv) $\mathbf{P}_{\mathcal{S}_l}(k_l, k_{l+1}) = \mathbf{P}_{\mathcal{S}_l}(k_l+1, k_{l+1}+1)$ if they are not zero. For ease of understanding, we only give an example with $N = 8$ and $j = 3$. The method can be extended to any N and j. Given the above parameters, there have

$$\mathbf{P}_{\mathcal{S}_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & a_4 \\ 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\mathbf{P}_{\mathcal{S}_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_1 & b_2 & b_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_1 & b_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where a_i, b_i are the variables to denote the probabilities in a simple way.

In order to obtain the probability of having 3 starvations, we need to compute $\mathbf{P}_{\mathcal{S}_1} \times \mathbf{P}_{\mathcal{S}_2}$ first, $\mathbf{P}_{\mathcal{S}_1} \times \mathbf{P}_{\mathcal{S}_2} =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1b_1 & a_1b_2+a_2b_1 & \sum_{k=1}^{3} a_k b_{4-k} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_1b_1 & a_1b_2+a_2b_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_1b_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Here, it is observed that the third row is obtained if we shift the second row to the right by 1 digit. Thus, we only need to multiply the second row of $\mathbf{P}_{\mathcal{S}_1}$ with the matrix $\mathbf{P}_{\mathcal{S}_2}$. The product is also an upper triangle matrix, where the elements exhibit the same structure as those of $\mathbf{P}_{\mathcal{S}_2}$. The complexity order is thus upper bounded by $O(N^2)$. Given that there are $j(> 2)$ starvations, the complexity order of matrix product is $O(N^{2(j-2)})$. Combined with the products of $\mathbf{P}_{\mathcal{E}}$, the total complexity has the order $O(N^{2j-2})$.