Flow-Level QoE of Video Streaming in Wireless Networks

Yuedong Xu, Salaheddine Elayoubi, Eitan Altman, Rachid El-Azouzi, Yinghao Yu

Flow-level QoE of Video Streaming in Wireless Networks

Yuedong Xu, Salaheddine Elayoubi, Eitan Altman,
Rachid El-Azouzi, and Yinghao Yu *

October 12, 2016

Abstract

The Quality of Experience (QoE) of streaming service is often degraded by frequent playback interruptions. To mitigate the interruptions, the media player prefetches streaming contents before starting playback, at a cost of initial delay. We study the QoE of streaming from the perspective of flow dynamics. Firstly, a framework is developed for QoE when streaming users join the network randomly and leave after downloading completion. We model the distribution of prefetching delay using partial differential equations (PDEs), and the probability generating function of playout buffer starvations using ordinary differential equations (ODEs) for constant bit-rate (CBR) streaming. The explicit form starvation probabilities and mean start-up delay are obtained by use of a matrix function approach. Secondly, we extend our framework to characterize the throughput variation caused by opportunistic scheduling at the base station, and the playback variation of variable bit-rate (VBR) streaming. Our study reveals that the flow dynamics is the fundamental reason of playback starvation. The QoE of streaming service is dominated by the first moments such as the average throughput of opportunistic scheduling and the mean playback rate. While the variances of throughput and playback rate have very limited impact on starvation behavior in practice.

Keywords: Quality of Experience, Start-up Delay, Buffer Starvation, Flow Dynamics, Video Streaming

1 Introduction

Streaming services are witnessing a rapid growth in mobile networks. According to Cisco VNI white paper, mobile video traffic exceeded 50 percent of total mobile data traffic for the first time in 2012 [1]. This presents new challenges for operators that are used to classify services into real-time (voice-like) and elastic (data-like) services. Indeed, classical quality of service (QoS) metrics in mobile networks are blocking rates for real-time traffic and average user throughput for elastic one, and operators dimension their networks for satisfying targets on those metrics [2]. However, the particular nature of streaming applications, halfway between real-time and elastic services, is raising the following difficult questions in wireless environments. First, which QoS metrics best represent the QoE perceived by users. Second, how to predict these QoE metrics for a given traffic intensity and to dimension the network accordingly.

The first step towards defining QoE and predicting it is to understand how streaming is played. In general, media players at the devices are equipped with a playout buffer that stores arriving packets. As long as there are packets in the buffer, the video is played smoothly. Once the buffer empties, the spacing between packets does not follow the original one. These *starvations* cause

*Yuedong Xu and Yinghao Yu are with Research Center of Smart Networks and Systems, School of Information Science and Engineering, Fudan University, Shanghai, China. Rachid El-Azouzi are with LIA, Universite d'Avignon, 339 Chemin des Meinajaries, Avignon, France. Salaheddine Elayoubi is with Orange Labs, Moulineaux, France. Eitan Altman is with University of Côte d'Azur, INRIA, 2004 route des Lucioles, Sophia Antipolis, France, as well as at LINCS, 23 Ave. d'Italie, Paris, France. Email: ydxu@fudan.edu.cn, salaheddine.elayoubi@orange-ftgroup.com, rachid.elazouzi@univ-avignon.fr, eitan.altman@inria.fr, 10300700057@fudan.edu.cn.

large *jitters* and are particularly annoying for end users that see frozen images. One feasible way to avoid starvations is to introduce a start-up (also called prefetching) delay before playing the stream, and a rebuffering delay after each starvation event. Then after a number of media frames accumulate in the buffer, the media player starts to work. This leads to two important sets of QoE metrics: starvation properties (probability, frequency, etc.) and startup/re-buffering delays.

Once the behavior of media streaming service is understood, the particularity of offering it over wireless networks is considered. Indeed, wireless channel is subject to a large variability due to fading, mobility etc. On top of this, it is a shared channel where multiple users are served simultaneously. Each flow delivers a file of finite size and leaves the system after the completion of downloading. The arrival and departure of flows cause a stochastic number of concurrent flows sharing the same wireless channel. We call this phenomena the *flow dynamics*. The flow dynamics leads to the variation of per-flow throughput that cannot be simply taken as a random variable. In a sense, with flow dynamics, the wireless channel consists of two time-scale variabilities: flow-level (tens of seconds) variability driven by the departures/arrivals of flows and wireless channel variability (milliseconds) driven by fast fading. In addition, the variable bit-rate (VBR) streaming leads to a variable service rate at the time scale of tens of milliseconds.

1.1 Related Literature

Starting from the mid-nineties, many works focused on performance analysis for real time video delivery over wireless networks. A large attention was given to enhance video coding in order to combat errors introduced by the wireless channel variability. [3] derived a theoretical framework for the picture quality after video transmission over lossy channels, based on a 2-state Markov model describing burst errors on the symbol level. Authors in [4] and [5] proposed methods for estimating the channel distortion and its impact on performance. These works mainly focused on ensuring robustness of video delivery over a variable wireless channel but did not consider the impact of flow-level dynamics. A more recent set of works considered flow-level performance in cellular networks delivering real time video. Authors in [2] proposed a queuing theory model for deriving QoS when integrating elastic and video traffic in cellular networks; video QoS was expressed by a blocking rate, while average throughputs and delays represent QoS for elastic traffic. Authors in [6] derived the Erlang-like capacity region for a traffic mix including real time video, the aim being to dimension the network for ensuring a target QoS. [7] derived the stability region of the network and showed how it is impacted by real-time video traffic.

With the increased popularity of streaming services over wireless systems, more attention has been dedicated to deriving QoE performance metrics for this new streaming service, knowing the initial buffering period and its relationship with starvation. QoE issue has been addressed in the important works [12, 13, 14, 8]. These works adopt different methodologies and assumptions for deriving QoE metrics. [12] considered a general G/G/1 queue where the arrival and service rates are characterized by their first two moments, while [13] considered a particular wireless channel model where the channel oscillates between *good* and *bad* states following the extended Gilbert model [15]. Authors in [14] considered a particular P2P video streaming based on random linear network coding; this simplifies the packet requests at the network layer and allows to model the receiver buffer as an M/D/1 queue. Finally, an M/M/1 queue model has been adopted in [8], allowing to derive explicit formula for QoE metrics. The playback interruption behaviors for an M/D/1 queue are considered in [11]. The authors further propose a set of mathematical models to characterize the distribution of start-up delay and buffer starvation when multiple streaming flows with finite duration share the same wireless bottleneck [10].

As of the tools used in the literature for deriving QoE metrics, they differ in the adopted system models. [12] adopted a diffusion approximation where the discrete buffer size is replaced with a Brownian motion whose drift and diffusion coefficients are calculated based on the first two moments of the arrival and service rates. [13] presented a probabilistic analysis based on an a priori knowledge of the playback and arrival curves. [14] calculated bounds on the playback interruption probability based on the adopted M/D/1 buffer model. Explicit formula of the exact distribution of the number of starvations has been obtained in [8] based on a Ballot theorem approach [16].

Authors in [8] also proposed an alternative approach for computing QoE metrics based on a recursive algorithm that performs better than the Ballot Theorem in terms of complexity. They further studied the QoE metrics of a persistent video streaming in cellular networks in [9].

The above-described works on QoE estimation are very useful for catching the impact of variability of the wireless channel due to fast fading or even user's mobility. However, the underlying models fail to capture the large variations due to flow dynamics. For instance, the diffusion approximation in [12] supposes that the drift and diffusion coefficients are constant over time, which is not true when the number of concurrent flows changes during playback in wireless environments. The assumption of Poisson packet arrivals in [14, 8] also fails to take into account these flow dynamics. Note that the analysis of [14] has been generalized to a two-state Markovian arrival process, but this corresponds more to a bursty traffic due to a Gilbert channel model than to flow dynamics.

1.2 Main Contributions and Organization

To the best of our knowledge, this paper is the first attempt to assess the impact of flow dynamics on the QoE of streaming. We model the system as two queues in tandem. The first queue, representing the scheduler of the base station, is modeled as a processor sharing queue, while the second represents the playout buffer whose arrival rates are governed by the output process of the base station queue. We first consider a static channel (no fast fading) with Constant Bit Rate (CBR) streaming, and derive the prefetching delay distribution and the starvation probability generation function using Partial Differential Equations (PDEs) as well as Ordinary Differential Equations (ODEs) constructed over the Markov process describing the flow dynamics. We then extend the model to the Variable Bit Rate (VBR) streaming using diffusion approximation. We next extend the model to include a fast fading channel and show that the impact of flow dynamics is preponderant over the variability of the channel due to fast fading. Extensive simulations show that our models are accurate enough to be used in QoE prediction.

Our study provides important insights in the design and optimization of video streaming services. Firstly, we bring a new understanding to the QoE of Internet streaming in cellular networks. We rigorously show that the QoE metrics are mostly influenced by the dynamic arrival and departure of streaming flows. While before our study, a number of related works mainly focus on the impact of variance of packet arrivals in a slot on the starvation behavior (e.g. [8, 12, 13, 14]), and their models as well as proposals may not be effective in shared wireless channel. Therefore, in order to improve QoE, network operations need to design new resource allocation strategies, and streaming users need to configure the prefetching scheme with the consideration of flow dynamics. Secondly, our models shed light on the explicit relationship between prefetching and QoE metrics. As the prefetching thresholds increase, the starvation probabilities decrease exponentially. The proposed models enable us to identify critical scenarios that the starvation probabilities are very sensitive to the prefetching thresholds. Thirdly, our explicit form models provide an easy way for online/offline QoE prediction and enhancement. With the pre-knowledge of throughput variation caused by flow dynamics, a streaming user can choose a better start-up/rebuffering threshold that guarantees the starvation probability below a certain level. Our models can help network operators to determine how many video streams can be admitted in a cell, and serve as a benchmark for network operators to design QoE-aware scheduling algorithms.

The main contributions of this work are summarized as follows:

1. We develop an analytical framework for assessing the impact of flow dynamics on streaming QoE in bandwidth shared wireless networks. The most striking result is that to handle this new issue, we propose a series of interdisciplinary approaches inherited from ruin analysis in actuarial science.

2. We present explicit form expressions for the QoE metrics (i.e. the start-up delay and the starvation behaviors), in which only some diagonalization steps of low-dimensional matrices are needed.

3. We evaluate the QoE metrics of both CBR and VBR streaming. Among three types of variations that may influence QoE metrics, our analyses show that the variation of throughput due to flow dynamics dominates the variations due to fast channel fading and variable sizes of video frame.

The remainder of this paper is organized as follows. Section 2 describes the system model and the QoE metrics. Section 3 presents the analytical framework for analyzing QoE taking into account flow dynamics. VBR streaming is analyzed in Section 4. The analytical model is verified through simulations in section 5 and a perfect match is demonstrated. Section 6 extends the QoE analysis framework to include the impact of fast fading. We discuss the potential limitations and their justifications in Section 7. Section 8 eventually concludes the paper.

2 Problem Description and Model

In this section, we first describe our motivation and the network settings. We then define the metrics of quality of experience for media streaming service, and present a queueing model for the playout buffer at a user.

2.1 Motivation and Network Description

We consider a wireless data network that supports a number of flows. When a new flow "joins" the network, it requests the streaming service from a media server. After the connection has been built, the streaming packets are transmitted through the base station (BS). The streaming flows have *finite* sizes, which means that a flow "leaves" the network once the transmission completes. Note that each active user cannot watch more than one streams at the mobile device simultaneously. Hence, we use the terms "flow" and "user" interchangeably. The flows are competing for finite capacity (or resource). When the number of flows sharing the bottleneck increases, network congestion occurs. The decrease of per-flow throughput may result in the undesirable playback interruption of video streaming services. The arrival and departure of flows further cause throughput fluctuation of the concurrent flows. To summarize, with the dynamics of co-existing flows in the bottleneck, per-flow throughput is not a random variable that has been studied in the literature, but a continuous time stochastic process.

In wireless data networks, a streaming flow may traverse both wired and wireless links, whereas the BS is the bottleneck for the sake of limited channel capacity. Most of Internet streaming providers such as Youtube and Youku use TCP/HTTP protocols to deliver streaming packets. TCP congestion control scheme tries to send as more packets as possible to the user (by consistently increasing the congestion window) in order to explore the available bandwidth. The sender reduces the congestion window when packet drops have been detected. The packet drops happen when the number of backlogged packets exceeds a certain threshold, e.g. the maximum queue length in the DropTail and the minimum threshold in the random early detection (RED) active queue management scheme. Therefore, it is reasonable to assume that the queue of an active flow is always backlogged at the BS. The packet losses rarely happen under adversary wireless channel conditions. The reason lines in that the adaptive coding and modulation in the physical layer, and ARQ scheme at the MAC layer can effectively avoid TCP packet loss caused by channel variation. A recent measurement study validates that TCP packet loss rate is usually less than 0.1% in 3G/4G channels [36].

Streaming flows may experience fast fading and normalized signal-to-noise ratio (NSNR) scheduling is usually adopted to achieve multiuser diversity with the consideration of fairness [18, 19]. The scheduling duration is commonly around 2ms [17]. NSNR selects the user that has the largest ratio of signal-to-noise ratio (SNR) compared with its mean SNR. It is similar to the well-known proportional fair (PF) scheduler in that they both attempt to achieve channel access-time fairness. We consider NSNR instead of PF for two reasons. First, the moments of throughput of PF do not have explicit results, even asymptotic ones (see [19] and references therein) when the channel capacity is computed according to the Shannon theorem. Second, NSNR needs the knowledge of

the average SNR that can be obtained from the history information. When a flow join the network, its throughput process is stationary as long as the number of active flows does not change. However, the throughput of PF scheduler is not stationary, but is a dynamic function of time t (see [20] for the ODE throughput model with two users). It relies on the configuration of the average throughput at time 0. The initial average throughput may influence the start-up delay, and cause the whole system intractable. Here, we make a declaration that our analytical framework applies to any wireless scheduling algorithm whose first two moments of throughput per-slot can be derived.

At the user side, incoming bits are reassembled into video *frames* step by step. These video frames are played with a deterministic rate, e.g. 25 frames per second (fps) in the TV and movie-making businesses. The size of a frame is determined by the video codec, i.e. a high definition video streaming or a complex video scenario require more bits to render each frame. We consider two modes of streaming services: constant bit-rate (CBR) and variable bit-rate (VBR). In CBR, the rate at which a codec's output data should be consumed is constant (i.e. the same size of frames). The VBR streaming has a variable frame size so as to deliver a more efficiently encoded and consistent watching experience. The frame size roughly follows Erlang/Gamma distributions [21].

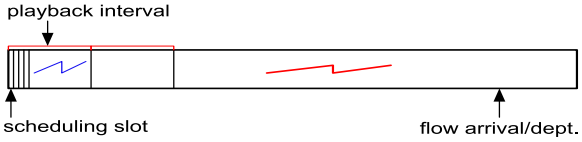

Figure 1: Illustration of three different time scales

We highlight the properties of the streaming system briefly to facilitate the mathematical modeling. In our system, there exist three time scales shown in Fig.1: i) the scheduling duration (e.g. 2ms); ii) playback interval (e.g. 40ms for a video frame rate of 25fps), and iii) duration of flow dynamics (lasting about tens of seconds). The scheduler and the media player do not work at the same granularity of time scale and job size.

The video streaming system is then deemed as a queueing process. The arrival to the queue contains two type of variations: one is due to flow arrival and departure, the other is due to fast fading and scheduled transmission. The service of the queue comprises only the variation of video frame size. In what follows, we build a mathematical framework to show the impact of these variations on the QoE of streaming services.

2.2 QoE Metrics

There exist five industry-standard video quality metrics. Authors in [22] summarize them into five terms: *join time*, *buffering ratio*, *rate of buffering events*, *average bitrate* and *rendering quality*. The first three metrics reflect the fundamental tradeoff in designing the prefetching process. The last two metrics are concerned with source coding. For analytical convenience, we redefine the QoE metrics regarding "prefetching" process.

- **Start-up delay:** The start-up delay denotes the duration (measured in seconds) between the time that a user initiates a session and the time that the media player starts playing video frames. In the initial prefetching phase, the player starts until the duration of received video reaches the *start-up threshold* measured in seconds of video segment. The start-up delay depicts the user's impatience of waiting for the video playback. Once the starvation event happens, the player pauses and resumes until the rebuffered video duration reaches the *rebuffering threshold*. We use the term *rebuffering delay* to differentiate the rebuffering time from the initial start-up delay.

- **Starvation probabilities:** When the playout buffer of a user becomes empty before the video has been completely played, we call this event a *starvation*. The starvation is very annoying to users. We adopt the starvation probability to evaluate the influence of the start-up threshold. In

addition, if the rebuffering process is taken into account, we analyze the probabilities of having a certain number of starvations.

Note that the start-up delay and the starvation probabilities can be used to compute the QoE metrics in [22]. The expected number of starvations is the sum of the products of the number of starvations and its probability. The expected buffering time equals to the product of the start-up delay in each rebuffering and the mean number of starvation events (including the initial prefetching).

2.3 Basic Queueing Model of Playout Buffer

We consider a wireless cellular network that supports up to K simultaneous flows. The purpose of admission control is to avoid the overloading of the cell. We make the following assumptions:
- **Single user type and static channel:** We begin with the case where streaming users coexist in a static channel, as this provides an easier route to understand the developed QoE evaluation model. We suppose that all the flows have the same SNR, and hence, in a static channel case, identical throughput. The concurrent flows compete for finite channel capacity such that the per-flow throughput is inversely proportional to the number of active flows. The extension to a Rayleigh fading channel is presented in Section 6 where mobile users have independent and identically distributed (i.i.d) SNR in each scheduling slot. Note that the i.i.d Rayleigh fading serves as a benchmark scenario for the analysis of channel capacity in wireless systems, e.g. [32, 33, 34, 35] among a large body of similar type of works.
- **Exponentially distributed video duration:** The video duration, measured in seconds, is exponentially distributed with mean $1/\theta$. Though the exponential distribution is not the most realistic way to describe video duration, it reveals the essential features of the system, and is the first step for more general distributions.
- **Processor sharing at the BS:** The scheduling slot is very small (e.g. $\leq$2ms in 3G LTE) compared with the service interval between two video frames (e.g. 40ms at 25fps) in the playout buffer. This property enables us to treat the BS as an egalitarian processor sharing queue where all the flows are served simultaneously. Hence, the per-flow throughput, depicted in continuous time, is a deterministic step-wise function of the number of active users in the static channel (e.g. [27]).
- **Continuous time playback:** The service of video contents is regarded as a continuous process, instead of a discrete rendering of adjacent video frames spaced by a fixed interval. This assumption is commonly used (see [28]) and is validated by simulations in this work. We assume that all the flows have identical video bit-rate.

All the notations are summarized in the Appendix. We denote by λ the arrival rate of new video streams. Let $Bitrate$ be the playback speed of video streams in bits per-second, and C (in bps) be the capacity of the static wireless channel. Given the exponential distribution of video duration, the file size F (measured in bits) is also exponentially distribution with mean $1/\theta_F = Bitrate/\theta$. Therefore, the dynamics of coexisting flows in the cell can be depicted as a continuous time Markov chain with a finite state space.

We concentrate on one "tagged" flow in order to gain the insight of dynamics of the playout buffer. At any time t, the tagged flow sees i other flows in a finite space $S := \{0, 1, \cdots, K-1\}$. Because wireless channel is a shared medium, per-flow throughput is inversely proportional to the number of coexisting flows. For the tagged flow, its throughput will change when a new flow joins or a concurrent flow completes the transmission. Given the assumption of exponentially distributed flow size, the service time of a flow is also exponentially distributed. This implies that the departure of a flow only depends on the current number of coexisting flow, while not the past history. We hereby define a stochastic process to capture the throughput dynamics of the tagged user.

Definition 1 (Environment Process) *The environmental process $\{I(t); t \geq 0\}$ is an irreducible and recurrent Markov chain that has discrete states in the set $S = \{0, 1, \cdots, K-1\}$. The transition rate from state i to $i+1$ is λ $(0 \leq i < K-1)$, and from state i to $i-1$ is ν_i $(0 < i \leq K-1)$. Denote*

by $\{\pi_i; i \in S\}$ the stationary distribution of environmental states. Note that the environment change is caused by flow arrival/departure.

The close-form expressions of ν_i and π_i ($\forall i \in S$) will be computed in the following section. At any state i, the throughput of the tagged flow is determined by the environment process as $b_i := \frac{C}{Bitrate \cdot (i+1)}$ in seconds of video contents. Let $N_e(t)$ be the number of changes in the environment by time t. Denote by A_l the time that the l^{th} environmental change takes place with $A_0 = 0$ and by $I_l := I(A_l)$ the state to which the environment changes after time A_l. When the tagged flow joins the network, we begin to study the dynamics of its playout buffer length. The entry time of the tagged flow is set to $t = 0$.

We denote by $Q(t)$ the length of playout buffer *measured in seconds* of video contents at time t. In the initial prefetching phase, the queue dynamics consists of only packet arrival, while in the playback phase, the queue dynamics contains both packet arrival and service of video frames. To differentiate these two phases, we denote by $Q_a(t)$ and $Q_b(t)$ the buffer length of the initial prefetching phase and the playback phase at time t respectively.

Initial Prefetching Phase: The queue dynamics is given by

$$Q_a(t) = \sum_{l=1}^{N_e(t)} b_{I_{l-1}}(A_l - A_{l-1}) + b_{I_{N_e(t)}}(t - A_{N_e(t)}). \tag{1}$$

Denote by q_a the start-up threshold. When $Q_a(t)$ increases from 0 until q_a, the initial prefetching phase ceases. The start-up delay, denoted by T_a, is expressed as

$$T_a = \inf\{t \geq 0 | Q_a(t) \geq q_a\}. \tag{2}$$

The cumulative distribution of T_a is given by

$$\Psi_i(t; q_a) = \mathbb{P}\{T_a < t | I(0) = i\} \tag{3}$$

if the tagged flow is in state i upon arrival.

Playback Phase: When the initial prefetching phase ceases and the playback phase begins immediately, the amount of prefetched content is exactly q_a. In the playback phase, the queueing process is expressed as

$$Q_b(t) = q_a - t + \sum_{l=1}^{N_e(t)} b_{I_{l-1}}(A_l - A_{l-1}) + b_{I_{N_e(t)}}(t - A_{N_e(t)}), \tag{4}$$

if the time axis starts at the instant of playing. Define $c_i := b_i - 1$ for all $i \in S$. Define

$$T_b = \inf\{t \geq 0 | Q_b(t) < 0\} \tag{5}$$

to be the time of observing empty buffer. Denote by $T_e (T_e < \infty)$ the completion time of downloading of the tagged flow. If T_b is less than T_e, a starvation event happens at the playout buffer. Then, the ultimate starvation probability is computed as

$$W_i(q_a) = \mathbb{P}\{T_b < T_e | I(0) = i, Q_b(0) = q_a\} \tag{6}$$

when the playback begins at state i, and stops at an arbitrary state that meets an empty queue for the first time. The ultimate starvation probability is the weighted sum of starvation probabilities at all the ergodic entry states.

3 Complete QoE analysis for CBR streaming

In this section, we model the starvation probability and the prefetching delay in a static channel where the media flows join and leave the system dynamically. The key idea is to investigate the queueing process of one "tagged" flow on the basis of differential equations.

3.1 Markov models of flow dynamics

Our purpose here is to construct two Markov chains to characterize the dynamics of the number of active flows. The first one models flow dynamics before the "tagged" flow joins in the network. Based on this Markov process, we can compute the stationary distribution of the number of active flows observed by the "tagged" flow at the instant when it is admitted. The second one describes the flow dynamics after the tagged flow is admitted. This Markov process enables us to investigate how the playout buffer of the tagged user changes.

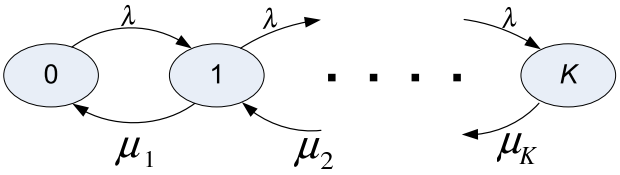

Figure 2: Markov chain before the tagged flow joins

We first look into the flow dynamics before the tagged flow joins. When the NSNR scheduling algorithm is used, the per-flow throughput is proportional to the reciprocal of flow population. The inter-arrival time of flows is exponentially distribution due to Poisson flow arrivals. The service time of a video at a state is equal to the size of the video in bits divided by its throughput. Because the video duration is exponentially distributed, the service time at a state is also exponentially distributed. Therefore, the dynamics of the number of coexisting flows can be depicted by a continuous time Markov chain (CTMC). This CTMC consists of a finite number of states for the sake of the admission control scheme at the base station. We model the change of number of coexisting flows as a CTMC $\mathbf{Z}_a := \{0, 1, \cdots, K\}$ shown in Fig.2. The transition rate from i to $i-1$ is $\mu_i := C\theta_F$. Note that the network capacity is a constant in the static channel. Hence, we let $\mu_i = \mu$ for $i = 1, 2, \cdots, K$ and $\mu_0 = 0$. Define $\rho := \frac{\lambda}{\mu}$ to be the load of the channel. Let z_i^a be the stationary probability that there exist i flows. We give the expression of z_i^a $(i \in S \cup \{K\})$ directly because it is easy to compute.

$$z_0^a = \frac{1-\rho}{1-\rho^{K+1}}; \quad z_i^a = \frac{\rho^i(1-\rho)}{1-\rho^{K+1}}, \quad \forall i = 1, \cdots, K.$$

The tagged user cannot be admitted at state K due to the admission control at the BS. Therefore, if it joins in the network successfully, it will observe i other flows with the probability π_i,

$$\pi_i = \frac{z_i^a}{1-z_K^a} = \frac{\rho^i(1-\rho)}{1-\rho^K}, \quad \forall i \in S. \tag{7}$$

After the tagged flow joins in the network, the Markov process $\mathbf{Z}_a$ has been altered. The states are the number of flows observed by the tagged user, and the transition rates are conditioned on the presence of the tagged flow. Therefore, we model the flow dynamics observed by the tagged flow through a finite-state Markov chain $\mathbf{Z}_b := \{0, 1, \cdots, K-1\}$ in Fig.3. Denoted by ν_i the transition rate from state i to $i-1$. The per-flow throughput at state i is $\frac{C}{(i+1)}$ so that there has $\nu_i := \frac{iC\theta_F}{(i+1)} = \frac{i}{i+1}\mu$ for all $i \in S$. For the simplicity of notations, we denote by λ_i the transition rate from state i to $i+1$. It is obvious to have $\lambda_i = \lambda$ for all $i \neq K-1$ and $\lambda_{K-1} = 0$.

3.2 Modeling prefetching delay distribution

Analysis of Prefetching Delay Distribution:

We want to know how long the tagged user needs to wait in the prefetching phase. Recall that q_a is the start-up threshold. Let q be the video content in seconds stored in the playout buffer. The prefetching time is only meaningful to the case that the video duration is **longer** than q_a. In

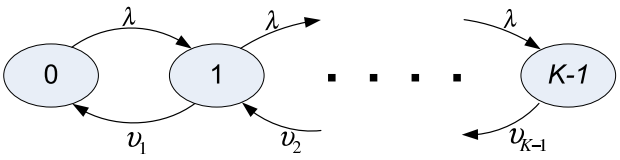

Figure 3: Flow dynamics observed by tagged flow

the prefetching phase, because the playout buffer does not serve video frames, the queue length of the tagged flow evolves in an infinitesimal time interval $[0, h]$ with $h(> 0)$

$$Q(t + h) = Q(t) + b_i h. \tag{8}$$

The distribution of the prefetching time is difficult to solve directly. We resort to the following duality problem:

> DUALITY PROBLEM: What is the starvation probability by time t if the queue is depleted with rate $b_i (i \in S)$ and the duration of prefetched contents is q_a?

In the duality problem, the queue dynamics in $[0, h]$ is modified as

$$\tilde{Q}(t + h) = \tilde{Q}(t) - b_i h. \tag{9}$$

We define $U_i(q, t)$ $(\forall i \in S)$ to be the probability of starvation before time t, conditioned on the entry state i and the initially prefetched content q. We use differential equations to obtain $U_i(q, t)$. In the infinitesimal time interval $[0, h]$, there are four possible events

- no change of the concurrent flows;
- arrival of one flow;
- departure of one flow (not the tagged one);
- occurrence of more than one events.

After an infinitesimal slot h, the queue length decreases from q to $q - b_i h$. The probability that a new streaming flow arrives is $\lambda_i h$, the probability of the departure of an on-going flow is $\nu_i h$, and the probability of no change of current flows is $(1 - \lambda_i h - \nu_i h)$, according to the principle of embedded Markov chain. Conditioned on the events occurred in $[0, h]$, we have

$$\begin{aligned} U_i(q, t) \;=\; & (1 - \lambda_i h - \nu_i h) U_i(q - b_i h, t - h) \\ & + \lambda_i h U_{i+1}(q - b_i h, t - h) \\ & + \nu_i h U_{i-1}(q - b_i h, t - h) + o(h), \;\; \forall i \in S. \end{aligned} \tag{10}$$

The above equation yields for $i \in S$

$$\begin{aligned} & \frac{1}{h}(U_i(q, t) - U_i(q - b_i h, t - h)) = \\ & -(\lambda_i + \nu_i) U_i(q - b_i h, t - h) + \lambda_i U_{i+1}(q - b_i h, t - h) \\ & + \nu_i U_{i-1}(q - b_i h, t - h) + o(h)/h. \end{aligned} \tag{11}$$

When $h \to 0$, the left side of eq.(11) is the partial differentials of $U_i(q, t)$ over q and t. In other words, eq.(11) yields a set of linear partial differential equations (PDEs)

$$\begin{aligned} \frac{\partial U_i}{\partial t} \;=\; & -b_i \frac{\partial U_i}{\partial q} - (\lambda_i + \nu_i) U_i(q, t) \\ & + \lambda_i U_{i+1}(q, t) + \nu_i U_{i-1}(q, t), \;\; \forall i \in S, \end{aligned} \tag{12}$$

with the initial condition

$$U_i(q,0) = 0, \quad \forall q > 0 \tag{13}$$

and the boundary conditions at both sides

$$U_i(0,t) = 1, \quad \forall\, t > 0, \tag{14}$$

$$\lim_{q\to\infty} U_i(q,t) = 0, \quad \forall\, t > 0. \tag{15}$$

The initial condition in eq.(13) means that the starvation cannot happen at time 0 for $q > 0$. The right-side boundary condition says that the starvation will not happen before t if the initial prefetching is large enough.

Remark: The initial condition in eq.(13) is incomplete since $U_i(q,0)$ is not a function of q. Comparing eq.(13) with eq.(14), we find that $U_i(q,t)$ is discontinuous at $(q,t) = (0,0)$. In this scenario, the direct numerical integral of the PDEs is usually unstable and the analytical solution may be inaccurate.

In what follows, we present an explicit solution to the PDEs, and its approximation. Here, the c.d.f. of start-up delay is the solution of linear PDEs by letting q be q_a. To solve these linear PDEs, we first define a matrix as

$$\mathbf{M}_S = \begin{pmatrix} \lambda_0 & -\lambda_0 & 0 & \cdots & 0 & 0 \\ -\nu_1 & \lambda_1+\nu_1 & -\lambda_1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \cdots & -\nu_{N-1} & \nu_{N-1} \end{pmatrix}. \tag{16}$$

According to the lemma in the Appendix, the tridiagonal matrix $\mathbf{M}_S$ is diagonalizable. Let D_S be an invertible matrix, and Λ_S be a diagonal matrix that contains the eigenvalues of $\mathbf{M}_S$. Then, there has $\mathbf{M}_S = D_S \Lambda_S D_S^{-1}$. Define a vector $\mathbf{G}(q,t)$ as a set of step functions that have

$$\mathbf{G}_i(q,t) = \begin{cases} 0 & \text{if } q - b_i t > 0; \\ 1 & \text{if } q - b_i t \le 0. \end{cases} \tag{17}$$

Then, the linear PDEs in Eq.(12) are solved by

$$\boxed{\mathbf{U}(q,t) = D_S \exp(-\Lambda_S t) D_S^{-1} \cdot \mathbf{G}(q,t).} \tag{18}$$

So far, we have derived the explicit c.d.f. of start-up delay, which only involves a small-scale matrix decomposition. Detailed analysis can be found in the Appendix.

Analysis of Mean Prefetching Delay:

Another important metric is the mean completion time of the start-up process. We denote by $X_i(q;q_a)$ the expected start-up delay, given the initial entry state i, the current buffer duration q and the start-up threshold q_a. Following the similar approach, we examine the events taking place at an infinitesimal slot $[0,h]$. Without restating these events in details, we notice that the the stored video length increases from q to $q + b_i h$. The expected time to reach the start-up threshold increases by h. This gives rise to the dynamics of $X_i(q;q_a)$ for $i \in S$ by

$$\begin{aligned} X_i(q;q_a) &= (1 - \lambda_i h - \nu_i h)(h + X_i(q + b_i h; q_a)) \\ &\quad + \lambda_i h X_{i+1}(q + b_i h; q_a) + \nu_i h X_{i-1}(q + b_i h; q_a). \end{aligned} \tag{19}$$

Taking h small enough, we derive the following set of ordinary differential equations (ODEs)

$$\begin{aligned} b_i \dot{X}_i(q;q_a) &= (\lambda_i + \nu_i) X_i(q;q_a) \\ &\quad - \lambda_i X_{i+1}(q;q_a) - \nu_i X_{i-1}(q;q_a) - 1, \quad \forall i \in S. \end{aligned} \tag{20}$$

The boundary condition of $X_i(q;q_a)$ is given by

$$X_i(q_a;q_a) = 0, \quad \forall i \in S. \tag{21}$$

The physical interpretation is that the start-up process ends immediately if the initial q reaches q_a. Let $\mathbf{X}(q;q_a)$ be a vector of the expected start-up delays at different states. Define a matrix $\mathbf{M}_V$ to be $\mathbf{M}_V = \mathbf{diag}\{\frac{1}{b_i}\} \cdot \mathbf{M}_S$. Then, Eq.(20) can be rewritten as

$$\dot{\mathbf{X}}(q;q_a) = \mathbf{M}_V \mathbf{X}(q;q_a) - \{1/b_i\} \tag{22}$$

where $\{1/b_i\}$ is a column vector. We have the following property w.r.t the eigenvalues of $\mathbf{M}_V$.

Lemma 1 *The matrix $\mathbf{M}_V$ has K real non-negative eigenvalues, and is similar to a diagonal matrix.*

Proof: Please refer to the Appendix. ∎

Based on the above lemma, we have $\mathbf{M}_V := D_V \Lambda_V D_V^{-1}$ where D_V is an invertible matrix and Λ_V is the diagonal matrix containing all the eigenvalues of $\mathbf{M}_V$. In the ergodic Markov matrix $\mathbf{M}_V$, one of its eigenvalues is 0. According to theory of matrix functions, we can solve the above ODEs directly by

$$\begin{aligned}\mathbf{X}(q;q_a) &= \exp(\mathbf{M}_V q) X(0;q_a) \\ &\quad - \int_0^q \exp(\mathbf{M}_V(q-t)) \cdot \{1/b_i\} dt \\ &= D_V \exp(\Lambda_V q) D_V^{-1} X(0;q_a) \\ &\quad - D_V \int_0^q \exp(\Lambda_V(q-t)) dt \cdot D_V^{-1}\{1/b_i\}. \end{aligned} \tag{23}$$

The integral part is obtained by

$$\int_0^q \exp(\Lambda_V(q-t)) dt = \mathbf{diag}\{\frac{1}{\delta_V^i}(e^{\delta_V^i q} - 1)\} \tag{24}$$

where δ_V^i is the i^{th} eigenvalue in the diagonal matrix Λ_V. The term $\frac{1}{\delta_V^i}(e^{\delta_V^i q} - 1)$ equals to 1 when δ_V^i is 0. Submitting Eq.(24) to (23), we get the solution of the mean start-up delay

$$\begin{aligned}\mathbf{X}(q;q_a) &= D_V \exp(\Lambda_V q) D_V^{-1} X(0;q_a) \\ &\quad - D_V \mathbf{diag}\{\frac{1}{\delta_V^i}(e^{\delta_V^i q} - 1)\} D_V^{-1}\{\frac{1}{b_i}\}. \end{aligned} \tag{25}$$

In the above equation, the vector $X(0;q_a)$ is still unknown. Since $X_i(q;q_a)$ is 0 for all $i \in S$ at the boundary $q = q_a$, the mean start-up delay is solved by

$$\boxed{X(0;q_a) = D_V \exp(-\Lambda_V q_a) \mathbf{diag}\{\frac{1}{\delta_V^i}(e^{\delta_V^i q_a} - 1)\} D_V^{-1}\{\frac{1}{b_i}\}.} \tag{26}$$

We next analyze the probability that the prefetching process starts at state i and ends at state j, for all $i, j \in S$. Define

$$V_{i,j}(q;q_a) := \mathbb{P}\{I(T_a) = j | I(0) = i, Q(0) = q\}. \tag{27}$$

We can use the approach of obtaining $U_i(q,t)$ to solve $V_{i,j}(q;q_a)$. Note that we now use the queueing dynamics in eq.(8) instead of eq.(9). In the time interval $[0,h]$, there exists for all $i, j \in S$

$$\begin{aligned}V_{i,j}(q;q_a) &= (1 - \lambda_i h - \nu_i h) V_{i,j}(q + b_i h; q_a) \\ &+ \lambda_i h V_{i+1,j}(q + b_i h; q_a) + \nu_i h V_{i-1,j}(q + b_i h; q_a) + o(h). \end{aligned} \tag{28}$$

It is easy to see that $V_{i,j}(q;q_a)$ is the solution of the following differential equation

$$\begin{aligned} b_i \dot{V}_{i,j}(q;q_a) &= (\lambda_i + \nu_i) V_{i,j}(q;q_a) - \lambda_i V_{i+1,j}(q;q_a) \\ &\quad - \nu_i V_{i-1,j}(q;q_a), \ \forall i, j \in S, \end{aligned} \tag{29}$$

with the boundary condition

$$V_{i,j}(q_a; q_a) := \begin{cases} 1 & \text{if } i = j; \\ 0 & \text{otherwise .} \end{cases} \tag{30}$$

We interpret the boundary condition in the following way. If there exist $I(0) = i$ and $Q(0) = q_a$, the prefetching duration is 0 and the prefetching process ends at state i. Hence, $V_{i,j}(q_a; q_a)$ is 1 iff i equals to j. Recalling $\mathbf{M}_V = \mathbf{diag}\{\frac{1}{b_i}\} \cdot \mathbf{M}_S$. Define $\mathbf{1}_j$ to be a column vector in which the j^{th} element is 1 and all other elements are 0. Eq. (28) can be rewritten as

$$\dot{\mathbf{V}}(q; q_a) = \mathbf{M}_V \mathbf{V}(q; q_a). \tag{31}$$

Then, $\mathbf{V}(q; q_a)$ is solved by

$$\mathbf{V}(q; q_a) = \exp(\mathbf{M}_V q) \cdot \mathbf{V}(0; q_a). \tag{32}$$

Eq.(32) is expressed as

$$\mathbf{V}(q; q_a) = D_V \exp(\Lambda_V q) D_V^{-1} \cdot \mathbf{V}(0; q_a). \tag{33}$$

Submitting eq.(30) to eq.(33), we yield

$$\boxed{\mathbf{V}(q; q_a) = D_V \exp(\Lambda_V (q - q_a)) D_V^{-1} \cdot \mathbf{V}(q_a; q_a).} \tag{34}$$

3.3 Modeling starvation probability

The modeling of starvation probabilities should take into account the departure of the tagged flow. Recall that the CTMC in Fig. 3 assumes the persistent tagged flow, which is not suitable for the playback process. Before solving the starvation probabilities, we first modify the original CTMC by adding an absorbing state **A** shown in Fig. 4. The state **A** denotes the event that the tagged flow completes its downloading. Because of the exponentially distributed video duration, the transition from state i to state **A** is Poisson. Denote by φ_i the transition rate from state i to **A**. At state i, the bandwidth of a flow is $\frac{C}{i+1}$, resulting in $\varphi_i := \frac{\mu}{i+1}$. Define $c_i := b_i - 1$. The queue length of the tagged flow changes in an infinitesimal interval h according to the rule

$$Q(t + h) = Q(t) + c_i h. \tag{35}$$

If $c_i > 0$, the bandwidth is sufficient for continuous playback of the tagged flow and i other flows. For mathematical convenience, we suppose that q is 0^- if buffer starvation happens.When the tagged flow enters the absorbing state, it has downloaded the whole file with a non-empty playout buffer. Thus, the starvation probability at state **A** is 0 for any $q \geq 0$. Let $W_i(q)$ be the starvation probability with q seconds of contents in the playout buffer at state i. We derive a system of

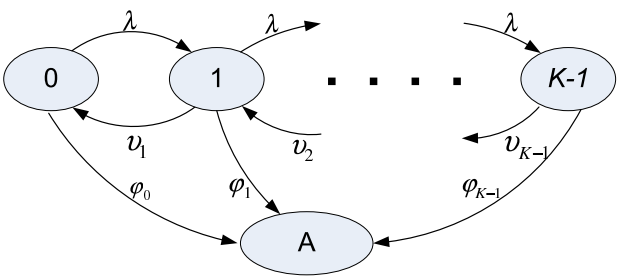

Figure 4: Markov chain for user dynamics with an absorbing state for departure of tagged flow

ordinary differential equations for $W_i(q)$. In an infinitesimal interval $[0, h]$, there are five possible events:

- no change of the concurrent flows;
- arrival of one more flow;
- departure of one flow (not the tagged flow);
- the tagged flow entering the absorbing state;
- occurrence of more than one events.

The above conditions give rise to the a set of equations

$$\begin{aligned} W_i(q) &= (1-(\lambda_i+\mu_i)h)W_i(q+c_ih) \\ &\quad +\lambda_i hW_{i+1}(q+c_ih)+\nu_i hW_{i-1}(q+c_ih)+o(h). \end{aligned} \tag{36}$$

When $h \to 0$, we obtain

$$c_i\dot{W}_i(q) = (\lambda_i+\mu_i)W_i(q) - \lambda_i W_{i+1}(q) - \nu_i W_{i-1}(q) \tag{37}$$

with the initial conditions

$$W_i(0) = 1, \quad \text{if } c_i < 0 \quad \forall i \tag{38}$$

and the boundary conditions

$$\lim_{q\to\infty} W_i(q) = 0, \quad \forall i. \tag{39}$$

Here, $\mathbf{W}(0)$ denotes the starvation probabilities with no initial prefetching. The boundary conditions are $W_i(q) = 0$ for all i as q approaches infinity. The above equations can be rewritten in the matrix form

$$\dot{\mathbf{W}}(q) = \mathbf{M}_W\mathbf{W}(q) \tag{40}$$

where $\mathbf{M}_W$ is expressed in eq.(41)

$$\begin{pmatrix} \frac{\lambda_0+\mu_0}{c_0} & -\frac{\lambda_0}{c_0} & 0 & \cdots & 0 & 0 \\ -\frac{\nu_1}{c_1} & \frac{\lambda_1+\mu_1}{c_1} & -\frac{\lambda_1}{c_1} & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \cdots & -\frac{\nu_{N-1}}{c_{N-1}} & \frac{\mu_{N-1}+\lambda_{N-1}}{c_{N-1}} \end{pmatrix}. \tag{41}$$

The solution to eq.(40) is given directly by

$$\mathbf{W}(q) = \exp(\mathbf{M}_W q)\cdot\mathbf{W}(0). \tag{42}$$

Note that $W_i(0) = 1$ holds for all i if $c_i < 0$ and $W_i(0)$ are unknowns for all i with $c_i > 0$. Using the proof of Lemma 1, we can show that $\mathbf{M}_W$ is similar to a diagonal matrix. There exists an invertible matrix D_W and a diagonal matrix Λ_W such that $\mathbf{M}_W := D_W\Lambda_W D_W^{-1}$. The starvation probabilities $\mathbf{W}(q)$ are expressed as

$$\boxed{\mathbf{W}(q) = D_W\exp(\Lambda_W q)D_W^{-1}\cdot\mathbf{W}(0).} \tag{43}$$

The eigenvalues in Λ_W are sorted in a decreasing order. According to Gershgorin circle theorem [30], the signs of eigenvalues are uncertain since the centers of the Gershgorin circles can be positive or negative. Based on the signs of c_i for $i \in \mathbf{S}$, we obtain the following corollary without proof due to its simplicity.

Corollary 1 *Suppose that c_i is positive for $0 \le i < k$ and is negative for $k \le i < K$. The matrix $\mathbf{M}_W$ has k positive eigenvalues and $K-k$ negative eigenvalues.*

The unknowns in $\mathbf{W}(0)$ can be solved subsequently. Define a vector $\bar{\mathbf{W}} := D_W^{-1} \cdot \mathbf{W}(0)$. When q is infinitely large, $\mathbf{W}(q)$ is a zero vector, resulting in $\exp(\Lambda_W q) D_W^{-1} \cdot \mathbf{W}(0) = 0$. Because the first k eigenvalues are positive in Λ_W, there must have $\bar{W}_i = 0$ for $i < k$. Hence, the unknowns $W_i(0)$ for $i < k$ can be derived.

Next, we build a bridge to interconnect the prefetching threshold and the starvation probability function $W_i(q)$. For a given prefetching threshold q_a, the starvation event takes place only when the video duration T_{video} is longer than q_a. This is to say, a flow with $T_{video} > q_a$ can be regarded as a tagged flow. When the prefetching process is finished, the tagged flow enters the playback process. Conditioned on the distribution of entry states π, the distribution of the states that the playback process begins (or the prefetching process ends) is computed by $\pi \cdot \mathbf{V}(0; q_a)$. Then, the starvation probability with the prefetching threshold q_a is obtained by

$$\begin{aligned} P_s(q_a) &= \mathbb{P}\{T_{video} > q_a\} \cdot \pi \cdot \mathbf{V}(0; q_a) \cdot \mathbf{W}(q_a) \\ &= \exp\big(-\theta q_a\big) \cdot \pi \cdot \mathbf{V}(0; q_a) \cdot \mathbf{W}(q_a). \end{aligned} \tag{44}$$

3.4 Modeling P.G.F. of starvation events

When a starvation event happens, the media player pauses until q_b seconds of video contents are re-buffered. A more interesting but challenging problem is how many starvations may happen in a streaming session. In this section, we come up with an approach to derive the probability generating function of starvation events.

We define a *path* as a sequence of prefetching and starvation events, as well as the event of completing the downloading. Obviously, the probability of a path depends on the number of starvations. We illustrate a typical path with L starvations in figure 5 that starts from a prefetching process and ends at a playback process. We denote by I_l^A the beginning state of the l^{th} prefetching, by I_l^B the beginning state of the l^{th} playback, and by I_e the end of downloading. The end of a prefetching process is exactly the beginning of a playback process. The end of a playback process is also the beginning of a subsequent prefetching process if the video has not been downloaded completely. This path contains a sequence of events happening at the states $\{I_1^A, I_1^B, I_2^A, I_2^B, \cdots, I_{L+1}^A, I_{L+1}^B, I_e\}$. The process between I_l^A and I_l^B is the l^{th} prefetching process, while that between I_l^B and I_{l+1}^A is the l^{th} playback process, $(1 \le l \le L)$. The first starvation takes place at the instant that the second prefetching process begins. The starvation event (e.g. $I_l^B,\ 1 \le l \le L$) cannot happen at the state i that has $c_i \ge 0$.

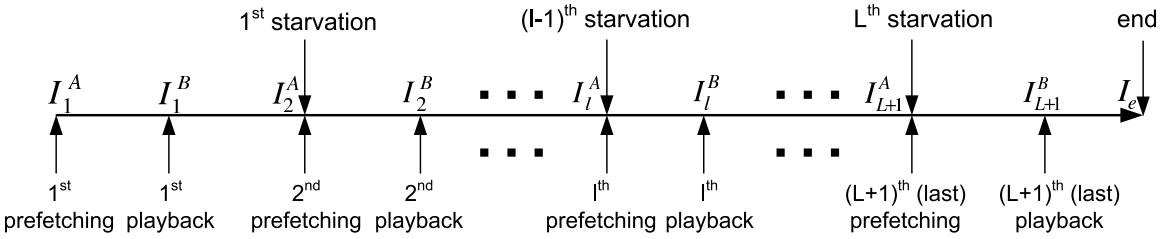

Figure 5: A path with L starvations

The sample path in figure 5 demonstrates a roadmap to find the p.g.f. of starvation events. We need to compute the transition probability along the path with all possible states. Recall that the transition probabilities from state I_l^A to I_l^B have been computed in section 3.2. The only missing part is the transition probabilities from state I_l^B to I_{l+1}^A.

Denote by $Y_{i,j}(q)$ the probability that a playback process starts at state i and meets with the empty buffer at state j with the prefetching threshold q. Define a matrix $\mathbf{Y}(q) := \{Y_{i,j}(q); i, j \in S\}$. Denote by $\mathbf{Y}_j(q)$ the vector of probabilities that the starvation takes place at state j with the prefetching threshold q, i.e. $\mathbf{Y}_j(q) := [Y_{0,j}(q), \cdots, Y_{K-1,j}(q)]^T$. Let $\mathbf{Y}_j(0) := [Y_{0,j}(0), \cdots, Y_{K-1,j}(0)]^T$ be the vector of those probabilities without the prefetching. Using the same argument, we get the

differential equation of $Y_{i,j}(q)$, $\forall i,j \in S$,

$$c_i \dot{Y}_{i,j}(q) = (\lambda_i + \mu_i) Y_{i,j}(q) - \lambda_i Y_{i+1,j}(q) - \nu_i Y_{i-1,j}(q). \tag{45}$$

The solution of eq.(45) is directly given by

$$\boxed{\mathbf{Y}_j(q) = D_W \exp(\Lambda_W q) D_W^{-1} \cdot \mathbf{Y}_j(0).} \tag{46}$$

The computation of $\mathbf{Y}_j(q)$ requires the knowledge of the boundary condition $\mathbf{Y}_j(0)$. Here, $Y_{i,j}(0) = 0,\ i \neq j$ and $Y_{i,j}(0) = 1$ if $c_i < 0$, and $Y_{i,j}(0) = 0$ if $c_{K-1} \geq 0$. The computation of remaining $Y_{i,j}(0)$ follows the same approach as that in the computation of $W_i(0)$.

When replacing q by q_a, we obtain the probability $Y_{ij}(q_a)$ that the first starvation happens at state j with i other flows observed by the tagged flow at the beginning of the playback process. The starvation probability in a rebuffering process is calculated by $Y_{ij}(q_b)$, given the rebuffering threshold q_b.

The probability of having L starvations can be expressed as the product of the probabilities from the first prefetching to the last playback. The probability vector from I_1^A to I_1^B is obtained by

$$\{\mathbb{P}_{I_1^A \to I_1^B}\} = \pi \cdot \exp\big(-\theta q_a\big) \cdot \mathbf{V}(0; q_a),\ \forall I_1^A, I_1^B \in S. \tag{47}$$

The probability vector from I_1^A to I_2^A is,

$$\begin{aligned} \{\mathbb{P}_{I_1^A \to I_2^A}\} &= \{\mathbb{P}_{I_1^A \to I_1^B}\} \cdot \mathbf{Y}(q_a) \\ &= \pi \cdot \exp\big(-q_a\theta\big) \cdot \mathbf{V}(0; q_a) \cdot \mathbf{Y}(q_a), \forall I_1^A, I_2^A \in S. \end{aligned} \tag{48}$$

Recall that the starvation happens at state I_2^A, and the rebuffering process ends at state I_2^B with the prefetched video duration q_b. We next compute the probability of having only one starvation denoted by $\mathbb{P}_{\text{1starv}}$. The possible paths include $\{I_1^A, I_1^B, I_2^A, I_e\}$ and $\{I_1^A, I_1^B, I_2^A, I_2^B, I_e\}$. The first part of $\mathbb{P}_{\text{1starv}}$ refers to the case that the remaining video duration is less than the rebuffering threshold q_b. The second part refers to the case that the remaining video duration is longer than q_b and there is no starvation after the rebuffering process.

$$\begin{aligned} \mathbb{P}_{\text{1starv}} &= \{\mathbb{P}_{I_1^A \to I_2^A}\} \cdot \mathbf{1} \cdot \big(1 - \exp(-q_b\theta)\big) \\ &\quad + \{\mathbb{P}_{I_1^A \to I_2^B}\} \cdot (1 - \mathbf{W}(q_b)) \\ &= \pi \cdot \exp\big(-q_a\theta\big) \cdot \mathbf{V}(0; q_a) \cdot \mathbf{W}(q_a) \cdot \big(1 - \exp(-q_b\theta)\big) \\ &\quad + \pi \exp\big(-(q_a + q_b)\theta\big) \mathbf{V}(0; q_a) \mathbf{Y}(q_a) \mathbf{V}(0; q_b)(1 - \mathbf{W}(q_b)). \end{aligned} \tag{49}$$

Here, the expression $(1 - \mathbf{W}(q_b))$ is the probability $I_2^A \to I_e$ in the first path and the expression $(1 - \mathbf{W}(q_b))$ is that of $I_2^B \to I_e$ in the second path. Similarly, we can deduce the probability of having $L (L > 1)$ starvations recursively by

$$\begin{aligned} \mathbb{P}_{\text{Lstarv}} &= \{\mathbb{P}_{I_1^A \to I_{L+1}^A}\} \cdot \mathbf{1} \cdot \big(1 - \exp(-q_b\theta)\big) \\ &\quad + \{\mathbb{P}_{I_1^A \to I_{L+1}^B}\} \cdot (1 - \mathbf{W}(q_b)) \\ &= \pi \cdot \exp\big(-q_a\theta\big) \cdot \mathbf{V}(0; q_a) \mathbf{Y}(q_a) \\ &\quad \cdot \Big(\exp\big(-q_b\theta\big) \mathbf{V}(0; q_b) \mathbf{Y}(q_b)\Big)^{L-1} \cdot \mathbf{1} \cdot \big(1 - \exp(-q_b\theta)\big) \\ &\quad + \pi \cdot \exp\big(-(q_a + q_b)\theta\big) \cdot \mathbf{V}(0; q_a) \mathbf{Y}(q_a) \cdot \Big(\exp\big(-q_b\theta\big) \\ &\quad \cdot \mathbf{V}(0; q_b) \mathbf{Y}(q_b)\Big)^{L-1} \cdot \mathbf{V}(0; q_b) \cdot (1 - \mathbf{W}(q_b)). \end{aligned} \tag{50}$$

Though the expression in eq.(50) looks complicated, it only involves duplicated products of matrices with dimension K that can be calculated easily.

4 VBR Streaming: Modeling QoE

In this section, we investigate the QoE of variable bit rate streaming (VBR). We introduce a diffusion process to model the variation of playback rate.

4.1 Queueing Model of VBR Streaming

In VBR, the frame size depends on the video scenario. For instance, the complex segments of video clips require more bits to render each frame than the simple segments. Then, the playback process exhibits the variation of service rate. The complex and the simple segments occur randomly, producing a mean playback rate. In this context, an important question is whether the jittering of playback rate significantly influences the starvation behavior or not.

In VBR streaming, the video file size is exponentially distributed with the mean $1/\theta_F$. Therefore, the Markovian property of flow departure still holds in Fig.2-4 and the transition rates remain the same as in Section 3. Whereas the video duration follows a general distribution. We define the mean playback rate to be $Bitrate$. The mean frame size is written as $\frac{Bitrate}{25}$ with frame rate 25fps. Denote by σ the standard deviation of video frames. The total variance of video frames is $25\sigma^2$ in one second.

We define an $Itô$ process $\{\mathcal{S}(t)\}$ to describe the total service measured in the duration of video contents by time t. The $Itô$ process $\{\mathcal{S}(t)\}$ satisfies the following stochastic differential equation

$$d\mathcal{S}(t) = \mathcal{S}(t+h) - \mathcal{S}(t) = 1 \cdot h + \bar{\sigma} d\mathcal{B}_h, \tag{51}$$

where $\mathcal{B}$ is the standard Wiener process and the subscript h denotes the duration. The process $\mathcal{B}_h$ satisfies $\mathcal{B}_h|_{h=0} = 0$, $E[\mathcal{B}_h] = 0$ and the derivative $d\mathcal{B}_h = \sqrt{h}\mathcal{N}(0,1)$ where $\mathcal{N}(0,1)$ is the standard Normal distribution. In eq.(51), the parameter $\bar{\sigma}$ denotes the standard deviation of video playback in a unit time. Hence, given the playback starting at time 0, the total variance of $\mathcal{S}(t)$ is $\mathrm{Var}[\mathcal{S}(t)] = \bar{\sigma}^2\mathrm{Var}[\mathcal{B}_t] = t\bar{\sigma}^2$. At the unit time $t = 1$ second, there has $\mathrm{Var}[\mathcal{S}(1)] = \bar{\sigma}^2$. Remember that 25 frames are served in one second. The total variance of served bits is thus $25\sigma^2$. When it is re-scaled by the video bitrate (measured in the duration of video contents), the variance is expressed as $\frac{25\sigma^2}{Bitrate^2}$. Therefore, we obtain the mapping $\bar{\sigma} = \frac{5\sigma}{Bitrate}$.

Remark: The service process of VBR differs from that of CBR in that the former has a diffusion part. Their gap, reflected by $\bar{\sigma}$, is determined by the standard deviation of video frame size σ in VBR and the mean bit-rate. However, σ and the mean bit-rate are positively correlated. When the mean bit-rate becomes small, σ is also small such that $\bar{\sigma}$ may not decrease obviously.

In this section, we integrate the playback perturbation with the fluid-level flow dynamics. The method employed here is inspired by the ruin analysis in actuarial science [24, 25]. With the continuous time assumption, we use the diffusion process $\mathcal{S}(t)$ to describe the queueing dynamics with the perturbation of playback rate. The continuous time queueing process in the prefetching phase, $\{Q_a(t); t \geq 0\}$, is defined as

$$Q_a(t) = \sum_{l=1}^{N_e(t)} b_{I_l}(A_l - A_{l-1}) + b_{I_{N_e(t)}}(t - A_{N_e(t)}) + \bar{\sigma}\mathcal{B}_t. \tag{52}$$

Similarly, the queueing process in the playback phase, $\{Q_b(t); t \geq 0\}$, is expressed as

$$Q_b(t) = q_a + \sum_{l=1}^{N_e(t)} c_{I_l}(A_l - A_{l-1}) + c_{I_{N_e(t)}}(t - A_{N_e(t)}) + \bar{\sigma}\mathcal{B}_t. \tag{53}$$

For the VBR streaming, the starvation can be caused by either the playback rate variation in small time scales or the flow dynamics in large time scales.

4.2 Starvation Probability

The computation of starvation probability uses the similar approach as the one in section 3. All possible events that take place in an infinitesimal time interval are taken into account. Conditioned on the flow dynamics and throughput perturbation in $[0, h]$, we have

$$\begin{aligned} W_i(q) &= (1-\lambda_i h-\mu_i h)W_i(q+c_i h+d\mathcal{B}_h) \\ &\quad +\lambda_i h W_{i+1}(q+c_i h+d\mathcal{B}_h) \\ &\quad +\nu_i h W_{i-1}(q+c_i h+d\mathcal{B}_h)+o(h),\ \forall i\in S. \end{aligned} \tag{54}$$

The above equations yield

$$\begin{aligned} &\frac{1}{h}\cdot\big(W_i(q+c_i h+d\mathcal{B}_h)-W_i(q)\big)=(\lambda_i+\mu_i)W_i(q+c_i h+d\mathcal{B}_h) \\ &-\lambda_i W_{i+1}(q+c_i h+d\mathcal{B}_h)-\nu_i W_{i-1}(q+c_i h+d\mathcal{B}_h)+\frac{o(h)}{h}. \end{aligned} \tag{55}$$

As $h\to 0$, the left-side of eq.(55) is expressed as

$$E[\frac{1}{h}\big(W_i(q+c_i h+d\mathcal{B}_h)-W_i(q)\big)]=c_i\dot{W}_i(q)+\frac{1}{2}\bar{\sigma}^2\ddot{W}(q), \tag{56}$$

according to [25]. Submitting (56) to (55), we obtain

$$\begin{aligned} a\ddot{W}_i(q)+c_i\dot{W}_i(q)-(\lambda_i+\mu_i)W_i(q)+\lambda_i W_{i+1}(q) \\ +\nu_i W_{i-1}(q)=0, \forall i\in S, \end{aligned} \tag{57}$$

where $\ddot{}$ denotes the second order derivative. The constant a equals to $\frac{1}{2}\bar{\sigma}^2$. The boundary conditions satisfy

$$W_i(0)=1,\ \ \forall i\in S. \tag{58}$$

$$\dot{W}_i(\infty)=0,\ \ \forall i\in S. \tag{59}$$

The starvation probability with no initial prefetching is 0 because the queueing process is oscillating very fast. The queue length will go "below" 0 immediately for sure. When q is infinitely large, the starvation probability $W_i(q)$ is 0. But $W_i(q)$ approaches 0 gradually, giving rise to the first-order derivative $\dot{W}_i(\infty)=0$. We denote by $\mathbf{Z}(q):=\{W_0(q),\cdots,W_{K-1}(q),\dot{W}_0(q),\cdots,\dot{W}_{K-1}(q)\}$. We further define two matrices, Z_3 and Z_4, that have the following forms:

$$Z_3=\mathbf{diag}\{c_i/a\}\cdot\mathbf{M}_W \quad \text{and} \quad Z_4=\mathbf{diag}\{-c_i/a\}.$$

Then, equations in (57) are rewritten in the matrix form

$$\dot{\mathbf{Z}}(q)=\mathbf{M}_Z\mathbf{Z}(q)=\begin{bmatrix}\mathbf{0} & I\\ Z_3 & Z_4\end{bmatrix}\cdot\mathbf{Z}(q). \tag{60}$$

The solution to eq.(60) is thus given by

$$\mathbf{Z}(q)=\exp\left(\mathbf{M}_Z q\right)\cdot\mathbf{Z}(0). \tag{61}$$

Since Z_3 is similar to a symmetric tridiagonal matrix and Z_4 is a diagonal matrix, we make the following conjecture.

Conjecture 1 *The matrix $\mathbf{M}_Z$ has $2K$ real eigenvalues, and can be expressed as $\mathbf{M}_Z=D_Z\Lambda_Z D_Z^{-1}$, where D_Z is an invertible matrix and Λ_Z is a diagonal matrix.*

On the basis of the above conjecture, eq.(61) is substituted by

$$\boxed{\mathbf{Z}(q)=D_Z\exp\left(\Lambda_Z q\right)D_Z^{-1}\cdot\mathbf{Z}(0).} \tag{62}$$

5 Simulation

In this section, we compare the numerical experiments with the developed framework using MATLAB. Our model exhibits excellent accuracy.

5.1 Constant bit-rate streaming

We consider a network with maximum number of ten simultaneous streaming flows and the capacity of 2.5Mbps. Flows arrive to the network with a Poisson rate $\lambda = 0.12$. Let the video duration be exponentially distributed with the mean 60 seconds. Then, there have $\mu = 0.1302$ and $\rho = 0.9216$ at the playback rate 360Kbps, and $\mu = 0.0868$ and $\rho = 1.3824$ at the playback rate 480Kbps. The simulation lasts 5×10^5 seconds.

Starvation probabilities: In this set of experiments, we will illustrate the overall starvation probability, the starvation probabilities when the playback process begins at different states, as well as the p.g.f. of starvation events.

Figure 6 shows the overall starvation probabilities with different settings of the start-up threshold. When it increases from 0 to 20s of video contents, the starvation probability decreases. The higher playback rate (e.g. 480Kbps) incurs larger starvation probabilities in comparison with the lower playback rate (e.g. 360Kbps). Our mathematical models match the simulations very well.

Figure 7 compares the starvation probabilities when the playback process begins at different states. A higher state refers to more coexisting flows (or congestions), and hence causing a larger starvation probability. Note that the arrival rates at state 7 and 9 are less than 360Kbps. Without prefetching, the starvation event happens for sure.

We further evaluate the probabilities of having one or two starvations in the whole procedure. For clarity, we choose the same value for the start-up and re-buffering thresholds. The starvation probabilities increase in the beginning and decrease afterwards when q_a (or q_b) increases from 0 to 30s of video segment. This is because there are many starvations with very small start-up threshold and few starvations with very large start-up threshold. Our analytical models predict the starvation probabilities accurately.

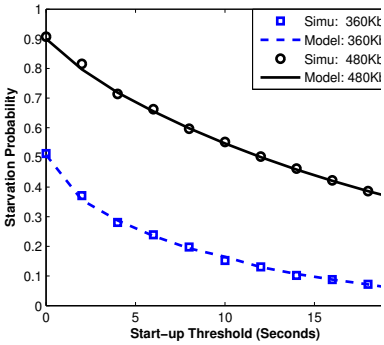

Figure 6: Overall starvation probability VS start-up threshold

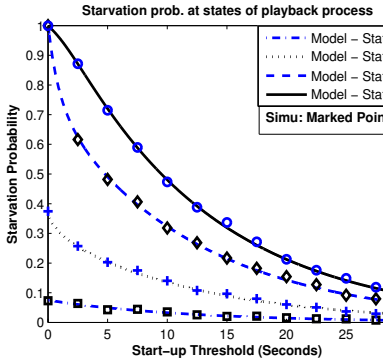

Figure 7: Starvation probabilities at different playback states with a playback rate 360Kbps

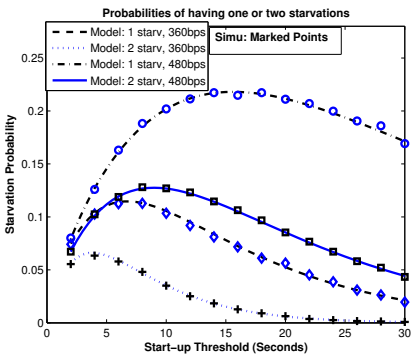

Figure 8: Probability of observing one and two starvations

Start-up delay: We illustrate the distribution of start-up delays in Fig.9 and 10. The start-up threshold is set to 10s and the playback rate is 360Kbps. We highlight the CDF curves when the tagged flow sees $3, 5, 7$ and 9 other flows respectively after entering the network. The CDF curves in Fig.9 are computed using MATLAB PDE integral function *pdepe*. For the cases $I(0) = 3, 5, 7$, the direct numerical integrals coincide with the experiments quite well. However, when the cumulative probability is close to 1, the numerical integral may oscillate (e.g. in the case $I(0) = 9$). This is because the initial condition $U_i(0, 0)$ is discontinuous in eqs.(13) and (15).

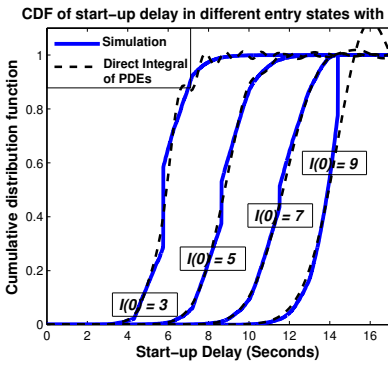

Figure 9: CDF of start-up delay with $q_a = 10$s: direct integral method

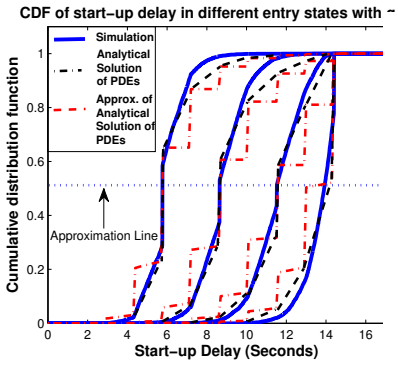

Figure 10: CDF of start-up delay with $q_a = 10$s: analytical solution and its approximation

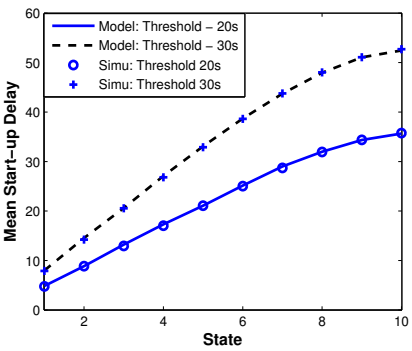

Figure 11: Mean start-up delay with $q_a = 20$s or 30s and 480Kbps playback rate.

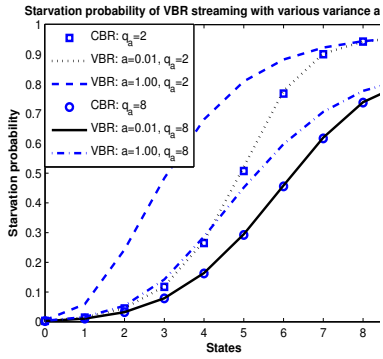

Figure 12: Starvation probabilities at all states with $d = 0.01$ and 0.5 computed by models

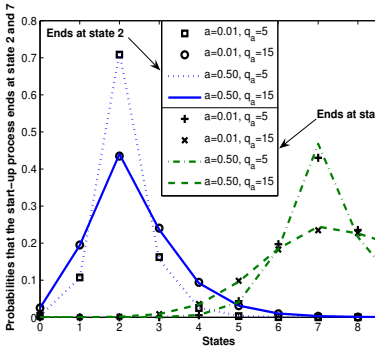

Figure 13: Probabilities that prefetching process starts from a state (from 0 to 9) and ends at state 2 or 7 with $d = 0.01$ and 0.5.

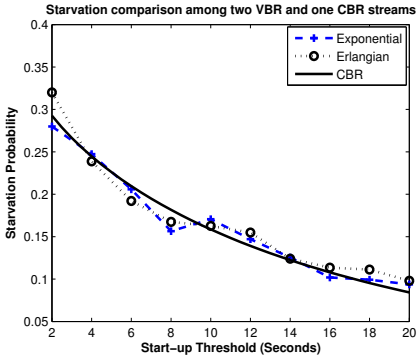

Figure 14: Starvation comparison among VBR of different frame size distributions and CBR model

In Fig.10, we compute the c.d.f. of start-up delay using our explicit model in eq.(18). Due to the discontinuous and incomplete initial condition, the explicit model results in a set of piece-wise curves. One can observe that each pair of CDF curves from the explicit model and from the experiments have a very similar trend as time t increases. However, their gap is large. Here, we propose a *heuristic* approximation technique to bridge this gap. A set of piece-wise lines are introduced to replace the CDF curves obtained from the explicit model. In the CDF curve of the explicit model, there exist a certain number of horizontal lines. Each horizontal line has two edges, the left one and the right one. If both two adjacent horizontal lines are above 0.5 (we usually refer to that the left edge is above 0.5), two left edges are connected. If both two lines are below 0.5, two right edges are connected. If one horizontal line is below 0.5 and the other is above 0.5, we connect the right edge of the lower horizontal line with the left edge of the upper horizontal line. This approximation is illustrated by the dashed lines in red. Though there lacks of a rigid justification of our approximation, we do observe that this technique is rather accurate in most of the experiments, and is also very simple.

We evaluate the mean start-up delay of a video stream at different entry states in Fig.11. The adopted start-up thresholds are 20s and 30s respectively. The video bitrate is 480Kbps. The proposed model matches the experiments well. The ratios of the mean start-up delays with different q_a are close to 2/3, which is the ratio of their start-up thresholds.

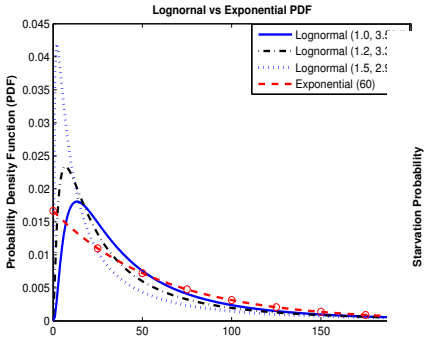

Figure 15: Comparison of probability density functions of lognormal and exponential distributions with the same expectation

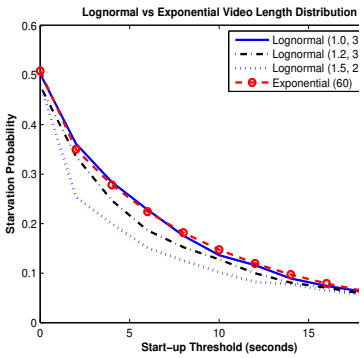

Figure 16: Starvation probability with lognormal distribution of video duration

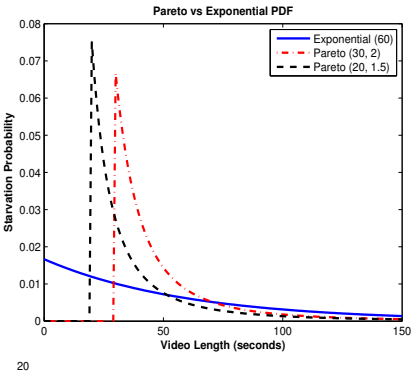

Figure 17: Comparison of probability density functions of Pareto and exponential distributions with the same expectation

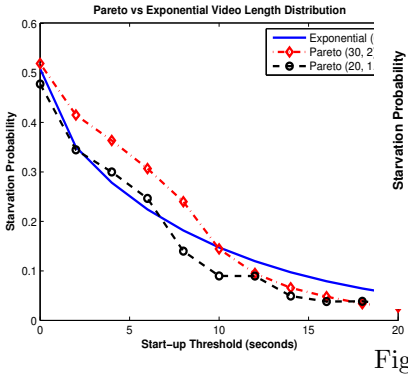

Figure 18: Starvation probability with lognormal distribution of video duration

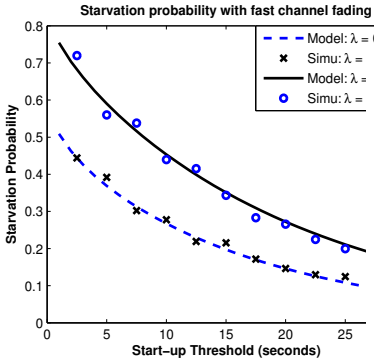

Figure 19: Starvation probability VS start-up threshold with Rayleigh fading (dashed line and crosses are for $\lambda = 0.07$; solid line and circles are for $\lambda = 0.09$)

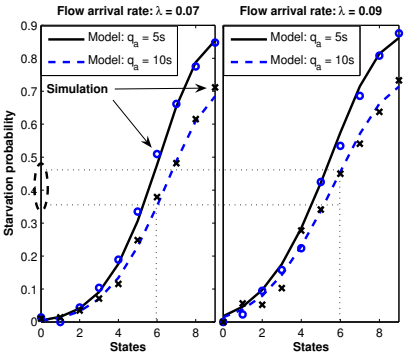

Figure 20: Starvation probability at different states with Rayleigh fading

5.2 Variable bit-rate streaming

We evaluate the QoE metrics of VBR streaming with a different set of parameters. The bandwidth is set to 2.0Mbps, and the flow arrival rate is set to 0.08. Each video streaming has the mean playabck rate of 360Kbps and a frame rate 25fps. The size of video files are exponentially distributed with the mean 2.16×10^7 bits (equivalent to 60s with the playback rate 360Kbps). Then, the traffic load of the system is given by $\rho = 0.864$. The per-flow throughput in states $5 \sim 9$ are insufficient to support the mean playback rate.

We first investigate how the playback variance influences the prefetching and the playback processes. Fig.12 shows the starvation probabilities when the start-up threshold and the variance change. When $a = 0.01$, the starvation probabilities computed from the VBR model are the same as those computed from the CBR model. While they differ greatly with $a = 1$. For the case $a = 1$, the jittering of playback rate influences the starvation probability more with $q_a = 2$ than with $q_a = 8$. Fig.13 compares the probabilities that the prefetching process ends at the state 2 and 7 respectively. From this set of experiment, we can see that even $a = 0.5$ does not obviously

influence the prefetching.

Fig.14 compares the numerical results of VBR streaming with the model for CBR streaming. In our simulation, the mean frame size is 14400 bits. According to [21], the video frame size roughly follows Erlang distribution. If the Erlang distribution is the sum of k i.i.d exponentially distributed r.v.s., the mean of these r.v.s. is $14400/k$. We consider two cases in this set of experiments, $k = 1$ (i.e. exponential r.v.) and $k = 3$. The resulting variances are $\bar{\sigma}^2 = 0.04$ (i.e. $a = 0.02$) for $k = 1$ and $\bar{\sigma}^2 = 0.013$ (i.e. $a = 0.0066$) for $k = 3$. The simulation time is 3×10^6 playback slots. From Fig.14, we are surprised to see that the Erlang distributions of video frames do not obviously influence the starvation probabilities. The analytical framework for CBR streaming is good enough to model the starvation behavior for VBR streaming.

5.3 Non-exponential Video Length Distribution

The type of video length distribution varies with regard to the type of content. A measurement study in [37] 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 distributions. Inspired by this observation, we want to evaluate the accuracy of proposed models experimentally when the benchmark exponential video length distribution does not hold. Here, the lognormal and Pareto distributions are considered. To enable fair comparison, we let all the distributions have the identical mean video length.

We consider three different lognormal distributions whose mean and variable of the corresponding normal distribution are (1.0, 3.5943), (1.2, 3.3743) and (1.5, 2.9693) respectively. Two Pareto distributions are analyzed with the set of parameters (30, 2) and (20, 1.5). The first parameter of Pareto distribution represents the minimum length for all the videos. Note that all the different set of parameters yield the same mean video length (i.e. 60s). Fig.15 and 17 illustrate probability density functions (PDFs) of the above distributions in contrast to the exponential distribution. The PDFs of lognormal and Pareto distributions do not resemble that of the exponential distribution.

Our purpose is to examine whether the starvation probability with lognormal and Pareto video length distributions can be well predicted by the model with exponential distribution. In Fig.16 and 18, we plot the starvation probability via simulations as the start-up threshold increases from 0 to 20s. For lognormal distribution, when the set of parameters are (1.0, 3.5943) and (1.2, 3.3743), the starvation probabilities in our simulations are very close to the model for the exponential distribution. If we retrospect to Fig.17, their PDFs are also close to that of exponential distribution. For the lognormal distribution with the set of parameters (1.5, 2.9693), the peak value of its PDF curve is 0.042 while that of the exponential distribution is 0.0167. This implies that these two distributions are quite distinct. As a result, the model for exponential distribution loses its accuracy in this case. In term of Pareto video length distribution, the first parameter determines the minimum video length. Let us take the set of parameters (30, 2) as an example. According to Fig.17, the PDF curve of the Pareto distribution is above that of the exponential distribution if the video length is less than 72s, and is below that vice versa. This implies that the Pareto distribution has less short and less long videos than the exponential distribution. Therefore, in comparison to exponential distribution, the starvation probability of this Pareto distribution is larger with a short start-up threshold for there are less short videos. As the start-up threshold becomes relatively larger, the starvation probability of the Pareto distribution is smaller for there are less long videos. Our simulations in Fig.18 validate the above observations. To summarize, our model well captures the trend of starvation probability as the start-up threshold increases. Even the distribution of video length is no longer exponential, the proposed model can still accurately predict the starvation probability for a variety of distributions.

6 Extension to Fast Fading

This section models the starvation behavior of CBR streaming when users experience fast channel fading. We compute the first two moments of bit arrival process and show how these parameters can be feed into our analytical framework.

Network description. Due to the change of radio condition (e.g. user mobility, or a car passing by the user), the signal strength is no longer a constant at different scheduling slots. To explore the multiuser diversity gain, the base station adopts the normalized SNR scheduling algorithm for allocating time slots to coexisting flows.

We begin with the scenario with a fixed population of i users (or flows) served by a single base station. In each slot, the users measure their channel qualities and feedback them to the BS. Based on the channel quality indications, the BS transmits to only one of the users every slot. Denote by $\gamma_{j,n}$ the instantaneous signal to noise ratio (SNR) of user j, $(1 \leq j \leq i)$, at slot n. As stated in most of previous work, we assume that all the users experience Rayleigh fast-fading. Denote by $\bar{\gamma}_j$ the average SNR of user j. Then, the received SNR of user j is an exponentially distributed random variable with the following probability density function $g_j(\gamma) = \frac{1}{\bar{\gamma}_j} \exp(-\frac{\gamma}{\bar{\gamma}_j})$. The NSNR scheduler selects the user that has the highest relative SNR for transmission, $j_n^* = \max_j\{\gamma_{j,n}/\bar{\gamma}_j,\ j = 1, 2, \cdots, i\}$, where j^* is the scheduled user at slot n. In this section, we consider the case of homogeneous average SNRs (i.e. $\bar{\gamma}_j = \bar{\gamma}$ for all j). Therefore, the NSNR scheduler is equivalent to the maximum sum rate (MSR) scheduler that gives the largest per-user throughput.Since the SNRs of different users are independently distributed, the scheduled SNR, denoted by γ^*, has the following probability density function [26] $g^*(\gamma) = \frac{i}{\bar{\gamma}} \exp(-\frac{\gamma}{\bar{\gamma}})\big(1 - \exp(-\frac{\gamma}{\bar{\gamma}})\big)^{i-1}$. Denote by $f(\gamma)$ the data rate of a user with the SNR γ. Here, $f(\cdot)$ can be a linear function in the low-SNR regime and a logarithmic function in the high SNR regime if the modulation scheme is continuous. For discrete modulations, $f(\cdot)$ is a step function of γ. Without loss of generality, we let $f(\gamma) = \log_2(1 + \gamma)$.

Analysis of throughput process. The fast fading along with NSNR scheduling brings variation of bit arrivals to the receiver. The analytical framework for VBR streaming can be naturally extended to this scenario. The only modification lies in that the jittering of playback rate is substituted by that of bit arrivals. Therefore, we need the knowledge of the mean throughput and its variance measured in the duration of video contents. To achieve this goal, we must obtain the mean throughput and its variance measured in bits first.

Denote by r_i^* the transmission rate of the user with the best SNR at a slot in each Hz when there are i active flows in the cell. Denote by r_i the transmission rate to *one particular flow* at a slot per Hz. Given the assumption that all the flows have the same average SNR, each flow has the equal probability of being scheduled. Hence, we can see

$$r_i := \begin{cases} r_i^* & \text{w.p. } \frac{1}{i}; \\ 0 & \text{w.p. } \frac{i-1}{i}. \end{cases} \tag{63}$$

For the r.v. r_i^*, its mean and variance are computed by

$$E[r_i^*] = \int_0^\infty f(\gamma) \cdot g^*(\gamma) d\gamma, \tag{64}$$

$$\mathrm{Var}[r_i^*] = \int_0^\infty f(\gamma)^2 \cdot g^*(\gamma) d\gamma - (E[r_i^*])^2. \tag{65}$$

The eqs (63)-(65) yield

$$E[r_i] = \frac{1}{i} E[r_i^*], \tag{66}$$

$$\begin{aligned} \mathrm{Var}[r_i] &= E[r_i^2] - (E[r_i])^2 = \frac{1}{i} E[(r_i^*)^2] - \frac{1}{i^2}(E[r_i^*])^2 \\ &= \frac{1}{i}\mathrm{Var}[r_i^*] + (E[r_i^*])^2(\frac{1}{i} - \frac{1}{i^2}). \end{aligned} \tag{67}$$

Denote by s the duration of scheduling slot (usually 2ms), and by B the width of wireless spectrum in Hz. Then, the mean and the variance of per-flow throughput measured in the *duration of video contents* are $\frac{B\cdot s\cdot E[r_i]}{Bitrate}$ and $(\frac{B\cdot s}{Bitrate})^2\cdot \mathrm{Var}[r_i]$ respectively in one slot.

Let R_i be the r.v. of *per-flow throughput in one second* that is measured by the duration of video contents. In one second, the total throughput of a flow at one Hz is the sum of throughput in $\frac{1}{s}$ slots. Therefore, the r.v. R_i is the sum of $\frac{1}{s}$ i.i.d r.v.s corresponding to the per-slot throughput. We can express the mean and the variance of R_i as follows:

$$E[R_i] = \frac{1}{s}\cdot\frac{B\cdot s\cdot E[r_i]}{Bitrate} = \frac{B\cdot E[r_i^*]}{i\cdot Bitrate}, \tag{68}$$

$$\begin{aligned}\mathrm{Var}[R_i] &= \frac{1}{s}\cdot(\frac{B\cdot s}{Bitrate})^2\cdot\mathrm{Var}[r_i]\\ &= \big(\frac{1}{i}\mathrm{Var}[r_i^*]+(E[r_i^*])^2(\frac{1}{i}-\frac{1}{i^2})\big)\cdot\frac{B^2\cdot s}{Bitrate^2}.\end{aligned} \tag{69}$$

Why does throughput variation of fast fading have very limited impact on starvations? In general, the frequency width B is 1∼5 MHz, the bit-rate is usually greater than 200 Kbps, and s equals to 0.002s. Then, $\mathrm{Var}[R_i]$ is usually at the order of 10^{-2}. If starvation happens at state i, $E[R_i]$ is usually less than 1, which means that $\frac{B}{Bitrate}$ needs to be small. However, the small $\frac{B}{Bitrate}$ results in the small variance $\mathrm{Var}[R_i]$. This is to say, if the variance of bit arrival process is large, there might not exist starvations. On the contrary, if the starvations appear, the variance is usually small so that its impact on the starvation is negligible. For this reason, we directly use the framework without diffusion approximation to model the streaming QoE in a fast fading channel.

Markov model of flow dynamics. To analyze the interaction between NSNR scheduling and the flow dynamics, a fluid-level capacity model is required. When the average SNR of all active users are the same, the per-flow throughput in each slot is i.i.d and only depends on the quantity of flows (see eq.(64)). Given the exponentially distributed video size, we can model the flow dynamics as a Markov process.

The Markov processes in Fig.2-4 contain transitions rates such as μ_i, ν_i and φ_i. However, it is not direct to feed the parameters of this section into the above Markov processes. In Fig.2, state i refers to the number of flows in the system. The departure rate is computed by $\mu_i = i\theta E[R_i]$ for $i \in S\cup\{K\}$, recalling that $E[R_i]$ is average per-user throughput in video duration per second. It is easy to obtain the stationary distribution of having i flows by

$$z_i^a = \frac{\lambda^i}{\prod_{l=1}^i\mu_l}\left[1+\sum_{j=1}^K\frac{\lambda^j}{\prod_{l=1}^j\mu_l}\right]^{-1}, \quad \forall i=0,\cdots,K,$$

(with the convention that $\prod$ over an empty set is 1). When a tagged user joins in the system and is also admitted, it observes i other flows with the following stationary distribution $\{\pi\}$:

$$\pi_i = \frac{z_i^a}{1-z_K^a} = \frac{\frac{\lambda^i}{\prod_{l=1}^i\mu_l}}{1+\sum_{j=1}^{K-1}\frac{\lambda^j}{\prod_{l=1}^j\mu_l}}, \quad \forall i\in S.$$

The Markov processes shown in Fig.3-4 are conditioned on the existence of the tagged flow. At state i, the per-user throughput is $E[R_{i+1}]$ because there are i flows plus the tagged one. Hence, the transition rate ν_i is computed by $\nu_i := i\theta\cdot E[R_{i+1}]$ for all $i\in S$. The transition rate φ_i is expressed as $\varphi_i := \theta\cdot E[R_{i+1}]$. Define $\tilde{\mu}_i$ as the total departure rate at state i that has

$$\tilde{\mu}_i := \varphi_i+\nu_i = (i+1)\theta E[R_{i+1}] = \mu_{i+1}, \tag{70}$$

in the presence of the tagged flow. The constants b_i and c_i are obtained by

$$b_i = E[R_{i+1}] \quad \text{and} \quad c_i = b_i - 1, \ \ \forall i\in S. \tag{71}$$

Substituting the above parameters to the framework in section 3, we can derive the approximated QoE metrics in a fast fading channel with flow dynamics.

Numerical Examples. Consider a wireless channel with frequency width of 1MHz. The average SNRs of users is 5dB. The base station allows at most 10 flows simultaneously, and schedules the transmission to one of them in every slot of duration 0.002s. The video duration is exponentially distributed with the mean of 90 seconds and the video bit rate is chosen to be 480Kbps. Then, the mean throughput are {3.5749, 2.3702, 1.7844, 1.4369, 1.2061, 1.0412, 0.9174, 0.8207, 0.7432, 0.6794} times the playback rate at states from 0 to 9. In other words, the mean throughput at states 6∼9 are insufficient to support the continuous playback. The variances at all states are {0.0083, 0.0144, 0.0144, 0.0134, 0.0124, 0.0114, 0.0105, 0.0098, 0.0091, 0.0086}, which are small enough. We consider two flow arrival rates, $\lambda = 0.07$ and $\lambda = 0.09$. For $\lambda = 0.07$, the traffic load ρ is greater than 1 at states 0∼5 and less than 1 at states 6∼9. For the latter case, there have $\rho > 1$ at all the states. Each set of simulation lasts 2×10^7 time slots.

Though we have demonstrated the reason that fast fading has very limited impact on the starvations, it is necessary to validate our claim through numerical examples. In Fig.19 we compare the starvation probabilities measured from a Rayleigh fading channel, and those computed from the model without considering throughput variation. The simulation matches the model quite well, which means that the flow-level dynamics have a dominant impact on the playback interruption, while the impact of throughput variation due to Rayleigh fading is negligible. In Fig.20 we examine the starvation probabilities when the playback process begins at different states. We test two start-up thresholds, $q_a = \{5, 10\}$, and two flow arrival rates, $\lambda = \{0.07, 0.09\}$. One can observe that the starvation probabilities do not differ much in high states (e.g. 8 and 9). However, the starvation probabilities in the states with mean throughput around 1 are distinguishable, in which state 6 is an example. With $\lambda = 0.09$, a tagged flow sees the congested network (more other flows) with a higher probability, and also encounters a higher probability of starvation afterwards.

7 Discussions and Potential Limitations

In recent years, the QoE metrics have been well modeled for a *single* flow that encounters variations in packet arrival rate [12, 8, 13, 14]. These variations arise from random packet loss in a wireless channel, congestion in a bottleneck, or abrupt change of bandwidth, in which they can be captured by an "independent" external stochastic process. Different from the above works, this paper targets at a more practical and challenging scenario with multiple streaming flows competing for finite capacity. Streaming flows arrive to the bottleneck dynamically where each flow is not on transmission perpetually, but has a finite duration. Hence, a flow's QoE is coupled with that of concurrent flows. Existing approaches are not capable to compute the QoE metrics with flow-level dynamics. Our major contribution is to build a novel framework to characterize the QoE metrics analytically in this scenario. We further demonstrate that variations caused by fast channel fading and video playback have marginal impact on the QoE metrics in comparison to flow-level dynamics. Despite of our contributions, this work still has several potential limitations.

Solutions are not completely explicit. The QoE metrics are expressed in the form of matrices explicitly. However, we still need to compute the eigenvalues of two matrices, which can only be done numerically. We justify this limitation from two angles. Firstly, computing eigenvalues of small matrices has negligible complexity. Equipped with our models, network planners can obtain streaming QoE immediately in any configuration without running simulations or emulations for a long period. Secondly, obtaining completely explicit QoE metrics is even impossible for a single stream. The important work [12] presents a nearly explicit solution with integral of error function. Authors in [14] come up with a close-form lower bound of starvation probability for a single stream. The solution in [8] is explicit only for M/M/1 and M/D/1 queueing models. In our opinion, completely explicit QoE metrics do not exist in the scenario that the competing video streams arrive and leave dynamically.

Assumptions on Channel Fading. The assumption of i.i.d SNR has been widely adopted to analyze the capacity of wireless channel and the performance of channel-aware schedulers [32, 33,

34, 35]. Although in practice the SNRs of users do not follow identical Rayleigh distributions, we insist on this assumption in this work. The reason lies in our main purpose which is to develop a novel analytical framework to characterize streaming QoE affected by flow-level dynamics at the shared bottleneck. This framework can be naturally extended to more realistic scenarios such as heterogeneous channel gains and more general distributions of video duration. Let us take the scenario of heterogeneous channel gains as an example. Users having the same SNR (or mean SNR) fall in one class. The service rates of users in different classes are different, which means that their departure rates are also different. The number of streams sharing the bottleneck can be modeled as a multi-dimensional Markov process. A state contains the number of concurrent flows in each class so that the total number of states is the product of the number of classes and the maximum number of allowed flows. We can then construct the PDEs and the ODEs on top of them, and compute the QoE metrics accordingly. The multi-dimensional Markov process involves a more complicated transition matrix between pair-wise states. However, it does not increase the computational complexity significantly. In today's cellular networks, each cell is designed to be small (e.g. Femtocell) to accommodate several users simultaneously. Meanwhile, only several physical transmission rates are offered to users so that the number of classes is small. Therefore, the eigenvalues of the transition matrices can be calculated easily and hence the QoE metrics.

Assumptions on Video Length. When video duration is not exponentially distributed, our models can still reveal how the starvation probability decreases with the increase of start-up threshold. In this work, we intentionally consider lognormal and Pareto video length distributions that prevail in Youtube-like Internet streaming systems. Our models based on exponential distribution well predict starvation probability of lognormal and Pareto distributions in some representative simulations. This validates the robustness of our modeling framework. Certainly, the accuracy of prediction relies on to what extent a non-exponential distribution resembles the exponential counterpart with identical mean. When video length distribution is far away from the exponential one, an alternative approach is to approximated by a phase-type distribution, that is, the combination of several exponential distributions with different expectations. This is equivalent to the scenario of non-identically distributed SNRs. The flows having the same mean duration fall in the same class. We can construct the multi-dimensional Markov process observed by a tagged flow and compute the QoE metrics in the same way. Due to the limit space and the organization of this work, we leave them for our future study.

8 Conclusion

In this work, we developed an analytical framework to compute the QoE metrics of media streaming service in wireless data networks. Our framework takes into account the dynamics of playout buffer at three time scales, the scheduling duration, the video playback variation, as well as the flow arrivals or departures. We show that the proposed models can accurately predict the distribution of prefetching delay and the probability generating function of buffer starvations. The analytical results demonstrate that the flow dynamics have dominant influence on QoE metrics compared to the variation of throughput caused by fast channel fading and that of video playback rate caused by VBR streaming.

References

[1] Cisco Virtual Networking Index. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html

[2] S. Borst and N. Hegde, "Integration of Streaming and Elastic Traffic in Wireless Networks", Infocom 2007.

[3] K. Stuhlmuller, N. Farber, M. Link and B. Girod, "Analysis of video transmission over lossy channels", *IEEE J. Sel. Areas Commun.*, Vol.18, No.6, pp:1012-1032, 2000.

[4] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience", *IEEE J. Sel. Areas Commun.*, 2000.

[5] Z.H. He, et al, "Joint Source Channel Rate-Distortion Analysis for Adaptive Mode Selection and Rate Control in Wireless Video Coding", *IEEE J. Sel. Areas Commun.*, Vol.12, No.6, pp:511-523, 2002.

[6] L. Rong, S.E. Elayoubi and O.B. Haddada, "Performance Evaluation of Cellular Networks Offering TV Services", *IEEE Trans. on Vehicular Tech.*, 2010.

[7] M.K. Karray, "Analytical evaluation of QoS in the downlink of OFDMA wireless cellular networks serving streaming and elastic traffic" *IEEE Trans. on Wireless Commun.*, 2010.

[8] Y.D. Xu, E. Altman, et. al, "Probabilistic Analysis of Buffer Starvation in Markovian Queues", *Proc. of IEEE Infocom 2012.*

[9] Y.D. Xu, E. Altman, et. al, "QoE Analysis of Media Streaming in Wireless Data Networks", *Proc. of IFIP Networking 2012.*

[10] Y.D. Xu, S.E. Elayoubi, E. Altman and R. El-azouzi, "Impact of Flow Dynamics on QoE of Streaming Services in Wireless Networks", *Proc. of IEEE Infocom 2013*, pp:2715-2723, Italy, 2013.

[11] Y.D. Xu, E. Altman, R. El-Azouzi, M. Haddad, S.E. Elayoubi and T. Jimenez. "Analysis of Buffer Starvation with Application to Objective QoE Optimization of Streaming Services", *IEEE Trans. Multimedia*, Vol.16(3), pp:813-827, 2014.

[12] 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.

[13] 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.

[14] A. ParandehGheibi et al, "Avoiding Interruptions a QoE Reliability Function for Streaming Media Applications", *IEEE J. Sel. Areas Commun.*, Vol.29, No.5, pp:1064-1074, 2011.

[15] H. Sanneck, G. Carle, and R. Koodli, " A framework model for packet loss metrics based on loss run length," *Proc. of ACM SIGMM 2000.*

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

[17] T. Bonald and A. Proutiere, "A Queueing Analysis of Data Networks", *Queueing Networks*, Springer, 2011.

[18] J.G. Choi and S. Bahk, "Cell-throughput analysis of the proportional fair scheduler in the single-cell environment," *IEEE Trans. Veh. Technol.*, Vol, 56, pp:766-778, 2007.

[19] G. Song and Y. Li, "Asymptotic throughput analysis for channel-aware scheduling," *IEEE Trans. Commun.*, Vol.54, No.10, pp.1827-1834, 2006.

[20] H.J. Kushner and P.A. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions", *IEEE Trans. Wireless Commun.*, Vol.3, No.4, pp.1250-1259, 2004.

[21] D. Masi, M. Fischer, D. Garbin, "Video Frame Size Distribution Analysis," *The Telecom. Rev.*, **19**, 2008.

[22] F. Dobrian, A. Awan, I. Stoica, et.al, "Understanding the Impact of Video Quality on User Engagement", *ACM SIGCOMM'2011.*

[23] S. Elayoubi and B. Fourestie, "Performance evaluation of admission control and adaptive modulation in OFDMA WiMax systems", *IEEE/ACM Trans. Networking*, **16**(5), 2008.

[24] Y. Lu and C.L. Tsai, "The expected discounted penalty at ruin for a Markov-Modulated risk process perturbed by diffusion," *North Amer. Actuarial J.* **11**(2):136-152, 2008.

[25] F. Dufresne, H.U. Gerber, "Risk theory for the compound Poisson process that is perturbed by diffusion", *Insurance: Mathematics and Economics*, Vol.10, pp:51-59, 1991.

[26] Y.J. Chang, F.T. Chien, and C.C. Kuo, "Cross-layer QoS Analysis of Opportunistic OFDM-TDMA and OFDMA Networks", *IEEE J. Sel. Areas Commun.*, Vol.25, 2007.

[27] S. Borst, "User-Level Performance of Channel-Aware Scheduling Algorithms in Wireless Data Networks", *Proc. of IEEE Infocom 2003*.

[28] B. Wang, W. Wei, Z. Guo and D. Towsley, "Multimedia Streaming via TCP: An Analytic Performance Study", *ACM TOMCCAP*, Vol.5, No.3, pp:1-23, 2004.

[29] G.A. Geist, "Reduction of a general matrix to tridiagonal form", *SIAM. J. Matrix Anal. & Appl.*, Vol.12, No.2, pp:362-373, 1991.

[30] G.H. Golub, and V.F. Van Loan, *Matrix Computations*, *John Hopkins University Press*, 3rd, pp:439-445, 1996.

[31] X. Cheng, C. Dale, and J.C. Liu "Statistics and social networks of Youtube videos", *Proc. of IEEE IWQoS.*, pp:229-238, 2008.

[32] D. Gesbert, M. Alouini, "How Much Feedback is Multi-User Diversity Really Worth?", *Proc. of IEEE ICC'2004.*, pp:234-238, 2004.

[33] P. Dighe, R. Mallik and S. Jamuar, "Analysis of TransmitReceive Diversity in Rayleigh Fading", *IEEE Trans. Communications*, Vol.51, No.4, pp:694-703, 2003.

[34] G. Song and Y. Li, "Asymptotic Throughput Analysis for Channel-Aware Scheduling", *IEEE Trans. Communications*, Vol.54, No.10, pp:1827-1834, 2006.

[35] A. Lapidoth and S. Shamai, "Fading Channels: How Perfect Need Perfect Side Information Be?", *IEEE Trans. Information Theory*, Vol.48, No.5, pp:1118-1134, 2002.

[36] Y. Chen, et al. "A Measurement-based Study of MultiPath TCP Performance over Wireless Networks", Proc. of ACM IMC 2013, Barcelona.

[37] X. Cheng, C. Dale, and J.C. Liu, "Statistics and social network of YouTube videos," Proc. of IEEE IWQoS, Netherlands, 2008, pp. 229.

Appendix (Supplementary Part)

Summary of Notations

Solving PDEs

Suppose that $U_i(q,t)$ is a function of variable r where q and t are expressed as $q(r)$ and $t(r)$. We take first-order derivative of $U_i(q,t)$ over r and obtain

$$\frac{dU_i}{dr} = \frac{\partial U_i}{\partial q}\frac{dq}{dr} + \frac{\partial U_i}{\partial t}\frac{dt}{dr}. \tag{72}$$

We first solve the following homogeneous PDEs originated from Eq.(12)

$$\frac{\partial U_i}{\partial t} + b_i\frac{\partial U_i}{\partial q} = 0, \quad \forall i \in S, \tag{73}$$

Comparing Eqs.(72) with Eqs.(73), we have

$$\frac{dq}{dr} = b_i \quad \text{and} \quad \frac{dt}{dr} = 1. \tag{74}$$

The above simple differential equations give rise to

$$t = t_0 + r \quad \text{and} \quad q = q_0^{(i)} + b_i r.$$

In general, t_0 is set to 0 such that there have

$$t = r \quad \text{and} \quad q = q_0^{(i)} + b_i t.$$

Then, U_i is a function of the variable $q_0^{(i)}$. Define $G_i(\cdot)$ to be a continuous and differentiable function in the range $[-\infty, +\infty]$. U_i is solved by

$$U_i(q,t) = G_i(q_0^{(i)}) = G_i(q - b_i t) \tag{75}$$

when the PDEs are homogeneous. We next proceed to consider the inhomogeneous parts at Eqs.(12) in the matrix form

$$d\mathbf{U}/dr = -\mathbf{M}_S. \tag{76}$$

Then, there has

$$\begin{aligned} \mathbf{U}(q,t) &= \exp(-\mathbf{M}_S r)\cdot U_0 = \exp(-\mathbf{M}_S t)\cdot U_0 \\ &= \exp(-\mathbf{M}_S t)\cdot\{G_i(q-b_i t)\} \end{aligned} \tag{77}$$

where $\{G_i(q-b_i t)\}$ denotes a column vector of $G_i(q-b_i t)$ for all $i = 0, \cdots, K-1$. In Eq.(73), there does not involve the transition of states. Based on the physical meaning, the starvation probability before time t is 0 if $q - b_i t > 0$, and is 1 vice versa. This yields the following expression

$$\mathbf{G}_i(q,t) = \begin{cases} 0 & \text{if } q - b_i t > 0; \\ 1 & \text{if } q - b_i t \le 0; \end{cases} \quad \forall i \in \mathbf{S}. \tag{78}$$

Submitting Eq.(78) to Eq.(77), we solve the inhomegeneous PDEs by

$$\begin{aligned} \mathbf{U}(q,t) &= \exp\left(-\mathbf{M}_S t\right)\cdot\mathbf{G}(q,t) \\ &= D_S \exp\left(-\Lambda_S t\right) D_S^{-1}\cdot\mathbf{G}(q,t). \end{aligned} \tag{79}$$

Proof of Lemma 1

Proof: Without loss of generality, we consider a tridiagonal matrix T in the form

$$T = \begin{pmatrix} x_1 & y_1 & 0 & \cdots & 0 & 0 \\ z_2 & x_2 & y_2 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \cdots & z_N & x_N \end{pmatrix}$$

where x_i, y_i, z_i are all real constants. Our claim is a natural conclusion of the following lemma.

Lemma 2 *[29] Assume that the coefficients y_i, $i = 1, \cdots, N-1$ are nonzero, and the products $y_i z_i$ are positive. Then, the matrix T is similar to a symmetric tridiagonal matrix. Therefore, its eigenvalues are all real.*

Here, $\mathbf{M}_V$ satisfies the conditions in the above lemma. Thus, $\mathbf{M}_V$ is similar to a symmetric matrix, and is diagonalizable. According to Gershgorin circle theorem [30], every eigenvalue of $\mathbf{M}_V$ lies within at least one of the Gershgorin discs. Because the diagonal element is positive, and is larger than the sum of absolution values of non-diagonal elements in each line, every eigenvalue cannot be negative. This concludes the proof. ∎

Notation	Definitions
λ	flow arrival rate
C	capacity of static wireless chennel
F	file size (random variable)
$1/\theta$	mean video duration
$1/\theta_F$	mean video file size
K	maximum number of coexisting flows
$Bitrate$	video bit-rate in bps
b_i, c_i	$b_i := \frac{C}{Bitrate(i+1)}, c_i := b_i - 1$
S	the set of other flows except the tagged flow
q	playout buffer length measured in seconds
q_a	start-up threshold
q_b	rebuffering threshold
$Q(t)$	playout queue length at time t
$Q_a(t)$	playout queue length of start-up phase at time t
$Q_b(t)$	playout queue length of playback phase at time t
T_a	start-up delay
T_b	time till starvation (empty playout buffer)
T_e	completion time of downloading
μ, μ_i	departure rate of a flow at state i
ρ, ρ_i	λ/μ, λ/μ_i,
ν_i	departure rate of a flow at state i observed by the tagged flow
φ_i	rate of completion of downloading
$W_i(q_a)$	starvation probability with initial state i and start-up threshold q_a
$U_i(q,t)$	starvation probability before time t, given initial conditions (i,q)
$V_{ij}(q;q_a)$	the probability that the prefetching process starts at state i and ends at state j, given initial buffer length q and threshold q_a
$X_i(q;q_a)$	expected start-up delay, given initial state i and initial buffer length q_a and start-up threshold q_a
$Y_{ij}(q)$	probability that a playback process starts at state i and starves at state j, given initial buffer length q
$Z_i(q)$	starvation probability of VBR streaming given initial state i and initial buffer length q
L	number of starvations experienced by a stream
$\mathcal{B}, \mathcal{B}_h$	standard Wiener process
σ^2	variance of frame size in VBR streaming
r_i	transmission rate to one particular flow per slot/Hz
r_i^*	transmission rate of user with best SNR per slot/Hz
R_i	per-flow throughput in one second measured by video duration
B	width of wireless spectrum in Hz
π_i	steady state probability of number of flows at state i
s	duration of a scheduling slot (2ms)
$\tilde{\mu}_i$	departure rate of a flow at state i in the presence of fast fading
Λ_*	eigenvalues of matrices $\mathbf{M}_*$
D_*	invertible matrix obtained from $\Lambda_* = D_*^{-1}\mathbf{M}_* D_*$

Table 1: Glossary of main notations