

Fresher Content or Smoother Playback? A Brownian-Approximation Framework for Scheduling Real-Time Wireless Video Streams

Ping-Chun Hsieh

National Chiao Tung University
pinghsieh@nctu.edu.tw

Xi Liu

Texas A&M University
xiliu.tamu@gmail.com

I-Hong Hou

Texas A&M University
ihou@tamu.edu

ABSTRACT

This paper presents a Brownian-approximation framework to optimize the quality of experience (QoE) for real-time video streaming in wireless networks. In real-time video streaming, one major challenge is to tackle the natural tension between the two most critical QoE metrics: playback latency and video interruption. To study this trade-off, we first propose an analytical model that precisely captures all aspects of the playback process of a real-time video stream, including playback latency, video interruptions, and packet dropping. Built on this model, we show that the playback process of a real-time video can be approximated by a two-sided reflected Brownian motion. Through such Brownian approximation, we are able to study the fundamental limits of the two QoE metrics and characterize a necessary and sufficient condition for a set of QoE performance requirements to be feasible. We propose a scheduling policy that satisfies any feasible set of QoE performance requirements and then obtain simple rules on the trade-off between playback latency and the video interrupt rates, in both heavy-traffic and under-loaded regimes. Finally, simulation results verify the accuracy of the proposed approximation and show that the proposed policy outperforms other popular baseline policies.

CCS CONCEPTS

•Networks → Network performance analysis;

ACM Reference format:

Ping-Chun Hsieh, Xi Liu, and I-Hong Hou. 2020. Fresher Content or Smoother Playback? A Brownian-Approximation Framework for Scheduling Real-Time Wireless Video Streams. In *Proceedings of International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, Boston, MA, USA, October 11–14, 2020 (*MobiHoc '20*), 10 pages.

DOI: 10.1145/3397166.3409121

1 INTRODUCTION

Real-time wireless video streaming has become ubiquitous due to the widespread use of mobile devices and the rapid development of various live streaming platforms, such as YouTube and Facebook Live. These platforms support not only the broadcast of live videos, but also various interactive activities, such as video conferencing and online webinars. To support the required level of interactivity,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiHoc '20, Boston, MA, USA

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
978-1-4503-8015-7/20/10...\$15.00
DOI: 10.1145/3397166.3409121

the video contents, which are continuously generated by the content providers in real-time, are required to be played smoothly at the video clients with sufficiently low latency (e.g. around 150-300 milliseconds [4]) so as to enable real-time engagement with the audience. Moreover, along with the wide adoption of wireless-enabled cameras, real-time wireless video streaming is now an integral part of many video surveillance applications, such as roadway traffic monitoring and teleoperation of unmanned aerial vehicles. To guarantee the required high responsiveness to the changes in the scene, smooth video playback with low latency is definitely critical.

To support the above applications, it is required to tackle a natural tension between the two critical factors of quality of experience (QoE): *playback latency* and *video interruption*. Playback latency refers to the difference between the generation time of a video frame at the video source and its designated playback time at the client. Playback latency reflects the freshness of the video content and needs to be kept as small as possible. To maintain a constantly low playback latency, each video is configured to meet a certain playback latency requirement, and the video contents that are not delivered to the client by the designated playback time will be dropped. In the meantime, due to the lack of video content to play, the video client instantly experiences video interruption. To achieve smooth playback, the amount of video interruption also needs to be kept as small as possible. However, with a more stringent playback latency, it becomes more difficult to avoid video interruption as there is less room for coping with randomness in network condition during video delivery. This issue becomes even more challenging in a wireless network environment due to the shared wireless resource and the unreliable nature of wireless channels.

While there has been a plethora of studies on the trade-off between prefetching delay and video interruption [10, 13, 20–22, 26, 27], all of them focus only on the playback of *on-demand* videos, which differ significantly from the real-time videos in packet generation, playback latency, and packet dropping. To the best of our knowledge, this paper is the first attempt to analytically study the trade-off between playback latency and video interruption as well as the trade-off of such QoE metrics among different clients for real-time video streams. The main contributions of this paper are:

- We propose an analytical model that precisely captures all aspects of the playback process of a real-time video stream, including the packet generation process, the playback latency, packet dropping, and video interruptions. The proposed model also addresses the unreliable nature of wireless transmissions. Through Brownian approximation, we show that the playback process can be approximated by a two-sided reflected Brownian motion.
- Based on the proposed model and the approximation, we study the fundamental limits of the trade-off between the two most important QoE metrics: the playback latency and video interruptions, among all clients. Moreover, we characterize a necessary

- and sufficient condition for a set of QoE performance requirements to be feasible, given the reliabilities of wireless links.
- Next, we propose a simple policy that jointly determines the amount of playback latency of each client and the scheduling decision of each packet transmission. We show that this policy is able to satisfy any feasible set of QoE performance requirements, and hence we say that it is QoE-optimal.
 - Under the proposed approximation, we study both heavy-traffic and under-loaded regimes and obtain simple rules on the trade-off between playback latency and the video interrupt rates: In the heavy-traffic regime, the video interrupt rates under WLD are inversely proportional to the playback latency; In the under-loaded regime, the video interrupt rates under WLD decrease exponentially fast with the playback latency.
 - Through numerical simulations, we show that the proposed approximation approach can capture the original playback processes accurately, and the proposed scheduling policy indeed outperforms the other popular baseline policies.

The rest of the paper is organized as follows: Section 2 provides an overview of the related research. Section 3 describes the system model and problem formulation. Section 4 discusses the characterization of the playback process. Section 5 presents the Brownian-approximation framework as well as the fundamental network properties. Section 6 presents the proposed scheduling policy and the proof of its QoE-optimality. Section 7 discusses the asymptotic results regarding playback latency. Simulation results are provided in Section 8. Finally, Section 9 concludes the paper.

2 RELATED WORK

Prefetching delay and video interruption. The inclusion of prefetching delay has been one of the major solutions to mitigating video interruption. For a single video stream, Liang and Liang [20] and Parandehgheibi *et al.* [22] study the trade-off between prefetching delay and interruption-free probability, under different video playback models. Luan *et al.* [21] and Xu *et al.* [26] characterize the relation between prefetching delay and playback smoothness by diffusion approximation and the Ballot theorem, respectively. For the case of multiple video streams, Xu *et al.* [27] consider the impact of flow dynamics on the number of video interruption events by solving differential equations. Joseph *et al.* [13] consider a QoE optimization problem, which jointly encapsulates video interruptions, initial prefetching, and video quality adaption, and present an asymptotically optimal scheduling algorithm. Despite the useful insights provided by the above works, they all assume that the videos are on-demand and thereby fail to capture the salient features of real-time video streams.

Brownian approximation. There has been a plethora of existing studies on using Brownian approximation for multi-class queueing networks, such as [6–8, 24]. While the above list is by no means exhaustive, it can be readily seen that the general procedure is to establish the limits of scaled queueing processes in the heavy-traffic regime through the reflection of a Brownian motion obtained from the scaled controlled processes [25]. Below we discuss the prior works that are most relevant to this paper: Several recent works have proposed to utilize Brownian approximation for network scheduling problems. Hou and Hsieh [10, 11] address wireless scheduling for short-term QoE via Brownian approximation. Specifically, under Brownian approximation, they characterize

lower bounds on total video interruptions and propose scheduling policies that achieve these bounds. However, they consider only on-demand videos and thereby fail to handle the inherent features of real-time packet generation and packet dropping in real-time video streaming. For multi-class queues with finite buffers, Atar and Shifrin [2] present heavy-traffic analysis and accordingly resort to solving a Brownian control problem in the diffusion limit. Different from [2], we take a different approach to directly study a two-sided reflected Brownian motion and explicitly characterize the relation between the playback latency and the achievable set of video interruption rates. In this way, we are able to investigate the trade-off of interest and obtain simple design rules in both heavy-traffic and under-loaded regimes.

Real-time wireless scheduling. To address wireless packet scheduling with strict deadlines, Hou *et al.* [9] propose an analytical framework and propose an optimal scheduling policy in terms of delivery ratio requirements. This formulation is later extended to various network settings, such as scheduling with delayed feedback [15], general traffic patterns [5], multi-cast scheduling [16], wireless ad hoc networks [14], and distributed access [19]. All the above works discuss real-time wireless scheduling, with an aim to optimize delivery ratios. By contrast, our goal is to tackle the fundamental trade-off between video interruption and playback latency in real-time video streaming.

3 MODEL AND PROBLEM FORMULATION

In this section, we formally describe the wireless network model, the model for real-time video streaming, and the problem formulation.

3.1 Network Topology and Channel Model

We consider a wireless network with one AP that serves N video clients, each of which is associated with one packet stream of a real-time video generated by a video source. For ease of exposition, we assume that all the videos are streamed in downlink¹, i.e. from the AP to the clients. For temporary storage of the video content to be played, each video is associated with two video buffers: one buffer is on the client side, and the other is maintained by the AP. When the video source generates a video packet, the video packet is first forwarded to the AP and stored at the AP-side buffer. The AP then forwards the video packet to the client to be stored at the client-side buffer. Since the bandwidth between the AP and the video source is usually much larger than the bandwidth at the edge, we also assume that the latency between the AP and the source of video contents is negligible. Time is slotted, and the size of each time slot is chosen to be the total time required for one packet transmission. For each client n , we use $B_n(t)$ and $Q_n(t)$ to denote the number of available video packets in the client-side buffer and that in the AP-side buffer at the end time slot t , respectively. Figure 1 shows an example of the AP-side and client-side video buffers with two clients.

In each time slot, the AP can transmit one packet to exactly one of the video clients. If the AP chooses to transmit a packet to a client whose AP-side buffer is empty, then the AP will simply transmit a dummy packet. By using dummy packets, we can assume that the AP employs a causal work-conserving scheduling policy that always chooses a client to transmit to in each time slot based on

¹While we focus on downlink streams in this paper, the model and the analysis can be easily extended to the uplink case with polling packets.

the past observed history. Let $I_n(t)$ be the indicator of the event that client n is scheduled for a packet transmission at time slot t . Under a work-conserving policy, $\sum_{n=1}^N I_n(t) = 1$, for all $t \geq 0$.

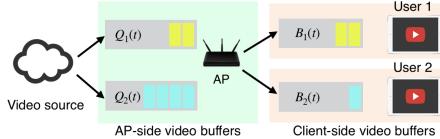


Figure 1: An example of video buffers with two clients.

Regarding wireless transmissions, we consider unreliable wireless transmissions that are subject to interference and collision from other neighboring networks. Since all links in the network experience a similar level of interference, we assume that all links have similar reliability. Specifically, each packet transmitted by the AP will be delivered successfully with probability $p \in (0, 1]$. The AP will be instantly notified about the outcome of the transmission via the acknowledgment from the client and can choose to retransmit the packet in a later time slot if the current transmission fails.

3.2 The Model for Real-Time Video Streaming

Each client is watching a real-time video stream. The stream of client n generates one video packet every $1/\lambda_n$ slots, where $1/\lambda_n$ is a finite positive integer. Hence, the average video bitrate of client n is λ_n packets per time slot. We consider real-time video streams with a fixed *playback latency* of ℓ_n/λ_n slots. Equivalently, ℓ_n is defined as the product of λ_n and the fixed playback latency (in slots). Specifically, for each client n , the video packet generated at the end of time slot t is forwarded immediately to the AP and is designated to be played by the client right after the end of time slot $t + \ell_n/\lambda_n$. The playback latency is intended to reduce potential video rebuffing and hence achieve smoother playback of a real-time video while guaranteeing the freshness of the video contents. Moreover, to maintain a fixed playback latency, a video packet that is not delivered to the client by its designated playback time will be dropped by the AP. When this happens, the client experiences video interruption due to the lack of video packets to play. For the rest of the paper, we call this event an *interruption*. For each client n , we use $D_n(t)$ to denote the total number of video interruptions up to time t , with $D_n(0) = 0$. Since a video interruption event occurs only when a video packet is dropped, $D_n(t)$ also represents the total number of dropped video packets up to time t .

Consider an example of the real-time video playback process with $\lambda_n = 1/2$ (or equivalently one video packet is played every 2 time slots), and $\ell_n = 2$ (or equivalently 4 slots), as illustrated in Figure 2. Since $\ell_n = 2$, we know there are two video packets (dubbed as packet 1 and packet 2 in Figure 2) available for transmission at the AP at $t = 0$. In particular, packet 1 and packet 2 are generated at the end of slots $t = -2$ and $t = 0$, respectively. In this example, the client receives packets in time slots 1, 4, 8, and 9. The client plays packet 1 right after the end of time slot 2 since it successfully receives packet 1. Similarly, the client plays packet 2 right after the end of time slot 4 since it receives packet 2 within the playback latency. By contrast, as the client fails to receive packet 3 within the playback latency, video interruption begins right after the end of time slot 6. Meanwhile, to maintain a fixed playback latency of $\ell_n = 2$, packet 3 is dropped by the AP at the end of time slot 6. At

time 8, the video playback resumes as the client receives packet 4 by time slot 8. Note that the AP is able to deliver the packet 5 during slot 9 since packet 5 is generated at the end of time slot 6 and hence is already available for transmission.

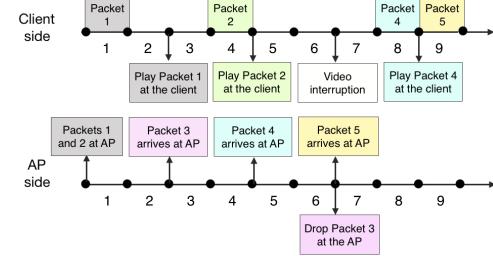


Figure 2: An example of real-time video playback process with $\lambda_n = 1/2$ and $\ell_n = 2$.

3.3 Problem Formulation

In this paper, we are interested in studying the trade-off between the playback latencies and the long-term average video interrupt rates of all clients. Specifically, given a total latency budget ℓ_{tot} , we study the set of video interrupt rates, $\limsup_{t \rightarrow \infty} D_n(t)/t$, that can be achieved under the constraint $\sum_{n=1}^N \ell_n \leq \ell_{\text{tot}}$. The set of achievable video interrupt rates describes the trade-off of video interrupt rates among different clients. Moreover, the relation between the set of achievable video interrupt rates and the value of ℓ_{tot} describes the trade-off between total latency and video interrupt rates. Hence, we formally define the *capacity region for QoE* and introduce the notion of *QoE-optimality* as follows.

Definition 3.1 (Capacity Region for QoE and QoE-Optimality). A $(N+1)$ -tuple $(\ell_{\text{tot}}, \delta_1, \dots, \delta_N)$ is said to be *feasible* if there exists a scheduling policy such that under $\sum_{n=1}^N \ell_n \leq \ell_{\text{tot}}$, we have

$$\limsup_{t \rightarrow \infty} \frac{D_n(t)}{t} \leq \delta_n, \quad (1)$$

for every $n \in \{1, \dots, N\}$. Moreover, the *capacity region for QoE* is defined as the set of all feasible tuples. A scheduling policy is said to be *QoE-optimal* if it can achieve every point in the capacity region for QoE.

The main objective of this paper is to design a QoE-optimal policy that jointly makes scheduling decisions and determines the allocation of the latency budget among the clients.

4 CHARACTERIZATION OF THE BUFFERING AND PLAYBACK PROCESSES

In this section, we formally characterize the playback process of a real-time video with playback latency. As discussed in Section 3.1, each video is associated with two video buffers: one buffer is on the client side, and the other is maintained by the AP. Recall that $B_n(t)$ and $Q_n(t)$ denote the number of available video packets in the client-side buffer and in the AP-side buffer at the end of time slot t , respectively. Given the fixed playback latency ℓ_n , we know that at any point of time, the amount of available and yet unplayed video data, which can be either in the AP-side buffer or in the client-side buffer, is exactly ℓ_n video packets. Therefore,

$$Q_n(t) + B_n(t) = \ell_n, \quad \forall t \geq 0. \quad (2)$$

Then, both $Q_n(t)$ and $B_n(t)$ are non-negative integers with $0 \leq Q_n(t) \leq \ell_n$ and $0 \leq B_n(t) \leq \ell_n$, for all $t \geq 0$. Suppose that the client-side buffer is initially empty, i.e. $B_n(0) = 0$, for all n . By (2), we thereby know $Q_n(0) = \ell_n$, for all n . Note that the video packets stored in the AP-side buffers at time 0 are essentially generated by the content provider during time $[-(\ell_n - 1)/\lambda_n, 0]$.

As described in Section 3.1, if the AP chooses to transmit a packet to client n at time t with $Q_n(t) = 0$ (i.e. the AP-side buffer for client n is empty), the AP will simply transmit a dummy packet to client n . Let $U_n(t)$ be the number of dummy packets delivered by the AP to the client n by time t , with $U_n(0) = 0$. Let $A_n(t)$ be the number of video packets received by client n up to time t , with $A_n(0) = 0$. Upon the designated playback time of each video packet, client n either consumes a video packet from the client-side buffer if $B_n(t) \geq 1$, or experiences video interruption if $B_n(t) = 0$. Let $S_n(t)$ be the number of video packets that have been played by client n by the beginning of time slot t , with $S_n(0) = 0$. Then, we have

$$B_n(t) = A_n(t) - S_n(t). \quad (3)$$

Since a video packet is dropped only when t is an integer multiple of $1/\lambda_n$ and $B_n(t) = 0$, we have

$$D_n(t) = \begin{cases} D_n(t-1) + 1 & , \text{if } B_n(t) = 0 \text{ and } t \in \{k/\lambda_n, k \in \mathbb{N}\} \\ D_n(t-1) & , \text{otherwise.} \end{cases} \quad (4)$$

Therefore, we know that $B_n(t) = 0$ if $D_n(t) - D_n(t-1) = 1$. Define

$$Z_n(t) := (A_n(t) + U_n(t)) - (D_n(t) + S_n(t)). \quad (5)$$

Note that $A_n(t) + U_n(t)$ is the total number of delivered packets, and $S_n(t) + D_n(t)$ is the number of packets that the client n should have played if there is no video interruption. Therefore, $Z_n(t)$ loosely reflects the status of the client-side buffer, with dummy packets included. By the definitions of $B_n(t)$ and $Z_n(t)$ in (3) and (5), we can rewrite $B_n(t)$ as

$$B_n(t) = Z_n(t) - U_n(t) + D_n(t). \quad (6)$$

We summarize the important properties of $B_n(t)$ as follows. For ease of notation, we let $D_n(-1) = 0$. For any $t \geq 0$, we have

$$B_n(t) = (Z_n(t) - U_n(t)) + D_n(t) \geq 0, \quad (7)$$

$$D_n(t+1) - D_n(t) \in \{0, 1\}, \quad D_n(0) = 0, \quad (8)$$

$$B_n(t)(D_n(t) - D_n(t-1)) = 0. \quad (9)$$

Now we turn to the AP-side buffer. Recall that $U_n(t)$ denotes the number of dummy packets received by the client n by time t . As a dummy packet is transmitted to the client n only if the AP-side buffer of the client n is empty, we know $U_n(t)$ can be updated as

$$U_n(t) = \begin{cases} U_n(t-1) + 1 & , \text{if } Q_n(t) = 0, t \notin \{1 + k/\lambda_n, k \in \mathbb{N} \cup \{0\}\}, \\ & \text{a packet is delivered to } n \text{ during slot } t. \\ U_n(t-1) & , \text{otherwise.} \end{cases} \quad (10)$$

Similar to (7)-(9), we summarize the useful properties of $Q_n(t)$ as follows. For ease of notation, we let $U_n(-1) = 0$. For any $t \geq 0$,

$$Q_n(t) = (\ell_n - (Z_n(t) + D_n(t))) + U_n(t) \geq 0, \quad (11)$$

$$U_n(t+1) - U_n(t) \in \{0, 1\}, \quad U_n(0) = 0, \quad (12)$$

$$Q_n(t)(U_n(t) - U_n(t-1)) = 0, \quad (13)$$

where (11) follows directly from (2) and (6). Note that the stochastic processes $D_n(t)$, $U_n(t)$, $B_n(t)$, $Q_n(t)$, $A_n(t)$, $S_n(t)$, and $Z_n(t)$ are right-continuous with left limits for every sample path since all of them change values only at integer t . By (7)-(9) and (11)-(13), we are able to connect $Z_n(t)$ with $D_n(t)$ and $U_n(t)$ in the following theorem. For ease of notation, we use $(\cdot)^+ = \max\{0, \cdot\}$.

THEOREM 1. *For any $Z_n(t)$, there exists a unique tuple of processes $(D_n(t), U_n(t), B_n(t), Q_n(t))$ that satisfies (7)-(9) and (11)-(13), for every sample path. Moreover, $D_n(t)$ and $U_n(t)$ are the unique solutions to the following recursive equations:*

$$D_n(t) = \sup_{0 \leq \tau \leq t} \left(-Z_n(\tau) + U_n(\tau) \right)^+, \quad (14)$$

$$U_n(t) = \sup_{0 \leq \tau \leq t} \left(Z_n(\tau) + D_n(\tau) - \ell_n \right)^+, \quad (15)$$

and $D_n(t)$ and $U_n(t)$ are non-decreasing.

PROOF. We prove the uniqueness result by the *two-sided reflection mapping*. Specifically, we take $Z_n(t)$ as the process of interest and let 0 and ℓ_n be the lower and upper barrier, respectively. If (14)-(15) are satisfied, the uniqueness of $D_n(t)$, $U_n(t)$, $B_n(t)$, and $Q_n(t)$ follows directly from [25, Theorem 14.8.1]. Next, to establish that (14)-(15) indeed hold, we present a useful lemma (provided in Appendix A.1 of [12] due to the space limitation) of discrete-time one-sided reflection mapping, which resembles the classic result of continuous-time one-sided reflection mapping [3, Theorem 6.1]. Based on this lemma, we know that (14) holds if and only if (7)-(9) are satisfied. By using the same argument, we also have that (15) holds if and only if (11)-(13) are satisfied. \square

REMARK 1. The two-sided reflection mapping $\{Z_n(t), D_n(t), U_n(t)\}$ is also called *double Skorokhod mapping* in the literature [17]. Moreover, from (14)-(15), it is easy to check that under any fixed sample path of $Z_n(t)$, a larger ℓ_n will lead to smaller $D_n(t)$ and $U_n(t)$. This fact manifests the fundamental trade-off between playback latency and video interruption.

5 THE BROWNIAN-APPROXIMATION FRAMEWORK

In this section, we formally introduce the Brownian-approximation framework for real-time video playback processes.

5.1 Fundamental Network Properties

To analyze video interruption, we start by introducing $Z(t)$ as

$$Z(t) := \sum_{n=1}^N \frac{Z_n(t)}{p} = \sum_{n=1}^N \frac{(A_n(t) + U_n(t)) - (D_n(t) + S_n(t))}{p} \quad (16)$$

$Z(t)$ is right-continuous with left limits since $Z_n(t)$ is right-continuous with left limits, for all n . Moreover, $Z(0) = 0$ as $Z_n(0) = 0$, for all n . As $Z(t)$ is a weighted sum of $Z_n(t)$, $Z(t)$ loosely reflects the network-wide buffer status on the clients' side, with dummy packets included. Recall that ℓ_{tot} is the total playback latency budget. By Theorem 1, we know that given the process $Z(t)$, there exists a unique pair of non-decreasing processes $(D(t), U(t))$ that satisfies

$$D(t) = \sup_{0 \leq \tau \leq t} \left(-Z(\tau) + U(\tau) \right)^+, \quad (17)$$

$$U(t) = \sup_{0 \leq \tau \leq t} \left(Z(\tau) + D(\tau) - \frac{\ell_{\text{tot}}}{p} \right)^+, \quad (18)$$

where $(\cdot)^+ = \max\{0, \cdot\}$. Note that as $Z(0) = 0$, we also have $D(0) = 0$ and $U(0) = 0$. Next, we describe an important property of $D(t)$ and $D_n(t)$ that holds regardless of the employed policy.

THEOREM 2. *Under any scheduling policy, we have*

$$D(t) \leq \frac{1}{p} \sum_{n=1}^N D_n(t), \quad (19)$$

for all $t \geq 0$ and for every sample path.

PROOF. We prove this by contradiction: Define $t_* := \inf\{t : D(t) > \sum_{n=1}^N \frac{1}{p} D_n(t)\}$ and assume $t_* < \infty$. We apply the recursive equations (14)-(15) and (17)-(18) to find an upper bound for $D(t_*)$ and a lower bound for $\sum_{n=1}^N \frac{1}{p} D_n(t)$ for every $t \geq 0$. By these two bounds and $D(t_*) > \sum_{n=1}^N \frac{1}{p} D_n(t_*)$, we can reach a contradiction. The detailed proof is presented in Appendix A.2 of [12]. \square

5.2 Brownian Approximation For Real-Time Video Streaming

In this section, we are ready to apply Brownian approximation to characterize the behavior of playback interruption.

5.2.1 Approximation Through the Fluid Limit and the Diffusion Limit. We first provide an outline of the approximation approach as follows: consider the *fluid limit* and *diffusion limit* of $Z_n(t)$ as

$$\bar{Z}_n(t) := \lim_{k \rightarrow \infty} \frac{Z_n(kt)}{k}, \quad (20)$$

$$\hat{Z}_n(t) := \lim_{k \rightarrow \infty} \frac{Z_n(kt) - k\bar{Z}_n(t)}{\sqrt{k}}, \quad (21)$$

respectively. Generally speaking, the fluid limit and the diffusion limit are meant to capture the evolution of a stochastic process based on the Strong Law of Large Numbers (SLLN) and the Central Limit Theorem (CLT), respectively [3]. For ease of exposition, we will focus on *ergodic* scheduling policies under which $\{Z_n(t+1) - Z_n(t), t \geq 0\}$ forms a positive recurrent Markov chain. In this case, both limits in (20)-(21) exist [25, Section 4.4], and the fluid limit can be further written as $\bar{Z}_n(t) = t \cdot \bar{Z}_n$. We consider the following approximation for $Z_n(t)$ [3, Section 6.5]:

$$Z_n(t) \xrightarrow{d} \bar{Z}_n(t) + \hat{Z}_n(t) =: Z_n^*(t), \quad (22)$$

where \xrightarrow{d} means that the two stochastic processes are approximately equal in distribution. By (22), we also know $Z_n^*(t)$ is right-continuous with left limits, for every sample path. By Theorem 1, we know that given the process $Z_n^*(t)$, there exists a unique pair of non-decreasing processes $(D_n^*(t), U_n^*(t))$ that satisfies

$$D_n^*(t) = \sup_{0 \leq \tau \leq t} \left(-Z_n^*(\tau) + U_n^*(\tau) \right)^+, \quad (23)$$

$$U_n^*(t) = \sup_{0 \leq \tau \leq t} \left(Z_n^*(\tau) + D_n^*(\tau) - \ell_n \right)^+. \quad (24)$$

Subsequently, based on Theorem 1 and (22)-(24), we consider the following approximation for $D_n(t)$ and $U_n(t)$:

$$D_n(t) \xrightarrow{d} D_n^*(t), \quad U_n(t) \xrightarrow{d} U_n^*(t). \quad (25)$$

Similar to (20)-(21), define the fluid limit and diffusion limit of $Z(t)$

$$\bar{Z}(t) := \lim_{k \rightarrow \infty} \frac{Z(kt)}{k}, \quad (26)$$

$$\hat{Z}(t) := \lim_{k \rightarrow \infty} \frac{Z(kt) - k\bar{Z}(t)}{\sqrt{k}}. \quad (27)$$

Again, under an ergodic scheduling policy, $\{Z(t+1) - Z(t), t \geq 0\}$ forms a positive recurrent Markov chain, and hence we know both limits in (26)-(27) exist [25, Section 4.4]. We will explicitly characterize $\bar{Z}(t)$ and $\hat{Z}(t)$ in Section 5.2.2. Similar to (22), we consider the following Brownian approximation for $Z(t)$ as

$$Z(t) \xrightarrow{d} \bar{Z}(t) + \hat{Z}(t) =: Z^*(t), \quad (28)$$

Next, we further define two processes $D^*(t; \ell_{\text{tot}})$ and $U^*(t; \ell_{\text{tot}})$ as

$$D^*(t; \ell_{\text{tot}}) = \sup_{0 \leq \tau \leq t} \left(-Z^*(\tau) + U^*(\tau; \ell_{\text{tot}}) \right)^+ \quad (29)$$

$$U^*(t; \ell_{\text{tot}}) = \sup_{0 \leq \tau \leq t} \left(Z^*(\tau) + D^*(\tau; \ell_{\text{tot}}) - \frac{\ell_{\text{tot}}}{p} \right)^+ \quad (30)$$

Since $Z^*(t)$ is right-continuous with left limits, by Theorem 1 we know that $D^*(t; \ell_{\text{tot}})$ and $U^*(t; \ell_{\text{tot}})$ can be uniquely characterized by (29)-(30). Note that we use the notations $D^*(t; \ell_{\text{tot}})$ and $U^*(t; \ell_{\text{tot}})$ to make explicit their dependence on the total playback latency budget. Figure 3 summarizes the general recipe of the Brownian approximation framework considered in this paper. Up to this point, we have discussed how to construct the approximation of interest with the help of the fluid and diffusion limits as well as the two-sided reflection mapping. As suggested by Figure 3, we shall proceed to characterize $Z^*(t)$ and $D^*(t; \ell_{\text{tot}})$ (Section 5.2.2) as well as derive $Z_n^*(t)$ and $D_n^*(t)$ under the proposed policy (Section 6).

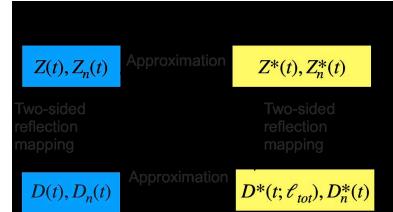


Figure 3: The Brownian approximation framework considered in this paper.

5.2.2 Characterizing $Z^*(t)$. In this section, we explicitly characterize the approximation process $Z^*(t)$. First, we define

$$\tilde{Z}(t) := \sum_{n=1}^N \frac{A_n(t) + U_n(t) - \lambda_n t}{p} \quad (31)$$

By the definitions of $Z(t)$ and $\tilde{Z}(t)$ in (16) and (31), we know $0 \leq Z(t) - \tilde{Z}(t) < N/p$, for all t . Due to the uniformly bounded difference between $Z(t)$ and $\tilde{Z}(t)$, $Z(t)$ and $\tilde{Z}(t)$ have the same fluid limit and diffusion limit, and we use $\tilde{Z}(t)$ as a proxy of $Z(t)$ to help characterize $Z^*(t)$. Under any work-conserving policy, in every time slot, the AP delivers a packet with probability p . Hence, by (31), for any $t \geq 0$,

$$\tilde{Z}(t+1) - \tilde{Z}(t) = \begin{cases} \frac{1}{p} - \sum_{n=1}^N \frac{\lambda_n}{p} & , \text{with probability } p \\ -\sum_{n=1}^N \frac{\lambda_n}{p} & , \text{with probability } 1-p \end{cases} \quad (32)$$

Moreover, $\{\tilde{Z}(t+1) - \tilde{Z}(t)\}$ is i.i.d. across all time slots. Define

$$\varepsilon := 1 - \sum_{n=1}^N \frac{\lambda_n}{p}, \quad (33)$$

which represents the normalized difference between the channel capacity and the traffic load. By (32), we have

$$\mathbb{E}[\tilde{Z}(t+1) - \tilde{Z}(t)] = p \cdot \left(\frac{1}{p} - \sum_{n=1}^N \frac{\lambda_n}{p} \right) + (1-p) \cdot \left(-\sum_{n=1}^N \frac{\lambda_n}{p} \right) = \varepsilon, \quad (34)$$

$$\begin{aligned} \text{Var}[\tilde{Z}(t+1) - \tilde{Z}(t)] &= \left(p \left(\frac{1}{p} - \sum_{n=1}^N \frac{\lambda_n}{p} \right)^2 + (1-p) \left(-\sum_{n=1}^N \frac{\lambda_n}{p} \right)^2 \right) - \varepsilon^2 = \frac{1}{p} - 1 =: \sigma^2. \end{aligned} \quad (35)$$

As $Z(t)$ and $\tilde{Z}(t)$ have the same fluid limit, by the Functional SLLN for i.i.d. random variables [3], we establish the fluid limit of $Z(t)$:

$$\bar{Z}(t) = \lim_{k \rightarrow \infty} \frac{Z(kt)}{k} = \lim_{k \rightarrow \infty} \frac{\tilde{Z}(kt)}{k} = \varepsilon t, \quad (37)$$

almost surely, for any work-conserving scheduling policy. Moreover, as $Z(t)$ and $\tilde{Z}(t)$ have the same diffusion limit, we can establish the diffusion limit of $Z(t)$ as

$$\hat{Z}(t) = \lim_{k \rightarrow \infty} \frac{Z(kt) - k\bar{Z}(t)}{\sqrt{k}} = \lim_{k \rightarrow \infty} \frac{\tilde{Z}(kt) - k\varepsilon t}{\sqrt{k}}, \quad (38)$$

where the last equality follows directly from (37). By the Functional CLT for i.i.d. random variables [3], we know that $\hat{Z}(t)$ is a Brownian motion with zero drift and variance σ^2 , where $\sigma^2 = (1/p) - 1$ as defined in (35)-(36). In other words, for any $t, \Delta t \geq 0$, we know $(\hat{Z}(t + \Delta t) - \hat{Z}(t))$ follows a Gaussian distribution with zero mean and variance $\Delta t \cdot \sigma^2$. Based on (28) and the above discussion on $\bar{Z}(t)$ and $\hat{Z}(t)$, we know that $Z^*(t)$ is a Brownian motion with drift ε and variance σ^2 . By mimicking (7), we can define

$$B^*(t; \ell_{\text{tot}}) := Z^*(t) - U^*(t; \ell_{\text{tot}}) + D^*(t; \ell_{\text{tot}}). \quad (39)$$

By Theorems 1 and (29)-(30), we know $B^*(t; \ell_{\text{tot}}) \in [0, \ell_{\text{tot}}/p]$ and that $B^*(t; \ell_{\text{tot}})$, $U^*(t; \ell_{\text{tot}})$, and $D^*(t; \ell_{\text{tot}})$ satisfy the same set of equations as (7)-(9). As we already know $Z^*(t)$ is a Brownian motion, by [1, Proposition 5.1], we further know that $B^*(t; \ell_{\text{tot}})$ satisfies the ergodic property, i.e. $B^*(t; \ell_{\text{tot}})$ admits a *unique stationary distribution*. As $D^*(t; \ell_{\text{tot}})$ is directly related to the event that $B^*(t; \ell_{\text{tot}})$ hits zero, such ergodic property implies that $D^*(t; \ell_{\text{tot}})$ grows linearly with time at a fixed rate on average. Hence, we can define the long-term average growth rate of $D^*(t; \ell_{\text{tot}})$ as

$$d^*(\ell_{\text{tot}}) := \lim_{t \rightarrow \infty} \frac{D^*(t; \ell_{\text{tot}})}{t}. \quad (40)$$

Note that given $\ell_{\text{tot}} > 0$, both $D^*(t; \ell_{\text{tot}})$ and $d^*(\ell_{\text{tot}})$ are well-defined, regardless of the policy. Moreover, it is easy to check that $d^*(\ell_{\text{tot}})$ is a decreasing function of the total playback latency ℓ_{tot} . This fact also manifests the trade-off between playback latency and video interruptions.

As will be formally shown in Section 7, the asymptotic behavior of $D_n^*(t)$ with respect to the playback latency is largely determined by the value of ε . To prepare for the subsequent analysis, here we highlight the three major regimes regarding the value of ε :

- **Heavy-traffic regime:** This regime represents the case where $\varepsilon = 1 - \sum_{n=1}^N (\lambda_n/p) = 0$. Therefore, $Z^*(t)$ is a driftless Brownian motion with finite variance σ^2 . Note that λ_n/p can be viewed as the equivalent workload of client n as $1/p$ is the expected number of required transmissions for each successful packet delivery. Hence, this regime corresponds to the case where the total channel resource equals the total video bitrate.

- **Under-loaded regime:** In this regime, $\varepsilon = 1 - \sum_{n=1}^N (\lambda_n/p) > 0$, and therefore (28) suggests that $Z^*(t)$ is a Brownian motion with positive drift. This regime corresponds to the case where the total channel resource is strictly larger than the total video bitrate. Therefore, it is intuitively feasible to have $B_n(t)$ close to ℓ_n for most of the time by properly scheduling each client based on its video bitrate. In Section 7, we will see that this effect also manifests itself in the fast-decaying behavior of $D_n^*(t)$ with respect to the playback latency.

- **Over-loaded regime:** This regime corresponds to that $\varepsilon < 0$. If $\varepsilon < 0$, then there must exist one client n that suffers from $B_n(t) = 0$ and hence excessive video interruption for most of the time, regardless of the scheduling policy.

The over-loaded regime is generally not the case of interest in designing policies. Therefore, in this paper we focus mainly on the heavy-traffic and under-loaded regimes, i.e. $\sum_{n=1}^N \frac{\lambda_n}{p} \leq 1$.

5.3 Capacity Region for QoE Under Brownian Approximation

Recall from Definition 3.1 that the capacity region for QoE is defined based on the feasible video interrupt rates $\limsup_{t \rightarrow \infty} D_n(t)/t$ under a playback latency budget. Moreover, recall from (25) that we propose to use $D_n^*(t)$ to approximate the original processes $D_n(t)$. Therefore, subsequently we proceed by considering the approximation $\limsup_{t \rightarrow \infty} D_n(t)/t \approx \limsup_{t \rightarrow \infty} D_n^*(t)/t$ and thereby study the set of feasible tuples based on $\limsup_{t \rightarrow \infty} D_n^*(t)/t$.

To quantify $\limsup_{t \rightarrow \infty} D_n^*(t)/t$, we propose to use $D^*(t; \ell_{\text{tot}})$ and the corresponding $d^*(\ell_{\text{tot}})$ defined in (40) as the reference measure for the following reasons: (i) as the distribution of $Z^*(t)$ does not depend on the employed scheduling policy, by (29)-(30) we know that both $D^*(t; \ell_{\text{tot}})$ and $U^*(t; \ell_{\text{tot}})$ also have invariant distributions under a given ℓ_{tot} across all scheduling policies; (ii) there is an inherent connection between $D_n^*(t)$ and $D^*(t; \ell_{\text{tot}})$ based on the two-sided reflection mappings in (23)-(24) and (29)-(30).

To formally compare the two stochastic processes $D^*(t; \ell_{\text{tot}})$ and $D_n^*(t)$, we first introduce the notion of *stochastic ordering* for stochastic processes as follows.

Definition 5.1 (Stochastic Ordering [23]). Let G_1 and G_2 be two real-valued random variables. We say that $G_1 \leq_{\text{st}} G_2$ if

$$\mathbb{P}[G_1 \geq x] \leq \mathbb{P}[G_2 \geq x], \quad \forall x \in \mathbb{R}. \quad (41)$$

Now we are ready to present an important property which connects $D^*(t; \ell_{\text{tot}})$ with $D_n^*(t)$. Specifically, we show that the inequality in Theorem 2 still holds under the approximation as follows.

THEOREM 3. *Under any $\ell_{\text{tot}} > 0$ and any scheduling policy,*

$$D^*(t; \ell_{\text{tot}}) \leq_{\text{st}} \sum_{n=1}^N \frac{1}{p} D_n^*(t), \quad \forall t \geq 0. \quad (42)$$

PROOF. We prove this by constructing a sequence of processes based on the scaling approach outlined in [25, Chapter 5.4] as well as Theorem 1 and the continuous mapping theorem. The detailed proof is provided in Appendix A.3 of the technical report [12]. \square

REMARK 2. To get some intuition of (42), consider a special case where $\ell_n = \infty$, for all n . This coincides with the on-demand video scenario, i.e. the AP already has the complete video for each client at time 0. In this degenerate case, (24) becomes $U_n^*(t) = 0$ and therefore (23) can be simplified as

$$D_n^*(t) = \sup_{0 \leq \tau \leq t} (-Z_n^*(\tau))^+. \quad (43)$$

Similarly, (30) becomes $U^*(t) = 0$ and (29) can be simplified as

$$D^*(t; \ell_{\text{tot}}) = \sup_{0 \leq \tau \leq t} (-Z^*(\tau))^+. \quad (44)$$

By combining (43)-(44), it is easy to verify that (42) indeed holds after applying the basic properties of supremum. Note that a similar result for this degenerate case (i.e. on-demand videos) has been derived in [10]. Different from [10], the proof of (42) for the general cases (i.e. finite playback latency ℓ_n) requires more involved analysis due to the recursion in (14)-(15) and (29)-(30).

Based on Theorem 3, under the Brownian approximation, we can obtain a necessary condition of a feasible tuple as follows.

COROLLARY 1. *Let $(\ell_{\text{tot}}, \delta_1, \dots, \delta_N)$ be a feasible tuple under the Brownian approximation with $\ell_{\text{tot}} > 0$ and $\delta_n \geq 0$, for all $n = 1, \dots, N$. Then, this tuple must satisfy*

$$\frac{1}{p} \sum_{n=1}^N \delta_n \geq d^*(\delta_{\ell_{\text{tot}}}). \quad (45)$$

PROOF. Recall from the beginning of Section 5.3 that under the Brownian approximation, the tuple is feasible if under the condition that $\sum_{n=1}^N \ell_n \leq \ell_{\text{tot}}$, $\limsup_{t \rightarrow \infty} D_n^*(t)/t \leq \delta_n$, for all n . Given the fact that $D_n^*(t)$ and $D^*(t; \ell_{\text{tot}})$ are non-decreasing processes in t , we divide both sides of (42) by t and take the limit superior to get

$$d^*(\ell_{\text{tot}}) = \lim_{t \rightarrow \infty} \frac{D^*(t; \ell_{\text{tot}})}{t} \leq_{\text{st}} \limsup_{t \rightarrow \infty} \frac{1}{p} \sum_{n=1}^N \frac{D_n^*(t)}{t} \leq \sum_{n=1}^N \frac{\delta_n}{p}. \quad (46)$$

6 A QOE-OPTIMAL SCHEDULING POLICY \square

In this section, we present a QoE-optimal scheduling policy for real-time video streams. Recall that in Section 5.1, we define the capacity region for QoE and provide a necessary condition of feasible tuples in Corollary 1. In this section, we further show that the condition provided in Corollary 1 is also sufficient.

6.1 Scheduling Policy

To begin with, we formally present the *weighted largest deficit policy* (*WLD*) as follows.

Weighted Largest Deficit Policy (WLD):

Let $\{\beta_n\}_{n=1}^N$ be the predetermined positive weight factors.

- (1) During initialization, the AP configures the playback latency of each client n as $\ell_n = \frac{\beta_n}{\sum_{m=1}^N \beta_m} \ell_{\text{tot}}$.
- (2) At each time t , the AP schedules the client with the largest $(\lambda_n t - (A_n(t) + U_n(t))) / \beta_n$, with ties broken arbitrarily.

REMARK 3. $\lambda_n t - (A_n(t) + U_n(t))$ can be viewed as *deficit* for client n as it reflects the difference between the number of packets that should have been played and the actual number of received packets. Moreover, as the video bitrate λ_n is usually predetermined and can be treated as hyperparameters, the WLD policy is able to make scheduling decisions based on $A_n(t)$ and $U_n(t)$, which can be updated based on the acknowledgments from the clients.

6.2 Proof of QoE-Optimality

To show that WLD is QoE-optimal, we first present the following *state-space collapse* property.

THEOREM 4. *For any given weight tuple $(\beta_1, \dots, \beta_N)$ with $\beta_n > 0$, for all n , and for any $\{\lambda_n\}$ and p such that $\sum_n \lambda_n / p \leq 1$, the WLD policy achieves*

$$\frac{1}{\beta_n} Z_n^*(t) = \frac{1}{\beta_m} Z_m^*(t), \quad (47)$$

for all pairs n, m . Moreover, we have

$$Z_n^*(t) = \frac{p\beta_n}{\sum_{m=1}^N \beta_m} Z^*(t), \quad \forall n. \quad (48)$$

PROOF. The proof first constructs N auxiliary processes that track the weighted sums of $\{Z_n(t)\}$. Next, we construct a Lyapunov function and calculates the one-step conditional drift to show that the auxiliary processes are positive recurrent. The detailed proof is presented in Appendix A.4 of the technical report [12]. \square

Recall from Section 5.2.2 that $Z^*(t)$ is a Brownian motion with drift ε and variance σ^2 . By (48) in Theorem 4, we know $Z_n^*(t)$ is also a Brownian motion with positive drift ε_n and variance σ_n^2 under the WLD policy, where

$$\varepsilon_n = \frac{\varepsilon \cdot p\beta_n}{\sum_{m=1}^N \beta_m}, \quad (49)$$

$$\sigma_n^2 = \left(\frac{p\beta_n}{\sum_{m=1}^N \beta_m} \right)^2 \sigma^2. \quad (50)$$

By Theorem 4, we are ready to show that WLD policy achieves every point in the capacity region for QoE.

THEOREM 5. *For any feasible tuple $(\ell_{\text{tot}}, \delta_1, \dots, \delta_N)$, under the WLD policy with $\beta_n / \delta_n = \beta_m / \delta_m$ for every pair n, m and $\ell_n = \ell_{\text{tot}} \beta_n / \sum_{m=1}^N \beta_m$ for every client n , we have*

$$\lim_{t \rightarrow \infty} \frac{D_n^*(t)}{t} = \frac{\delta_n p}{\sum_{m=1}^N \delta_m} d^*(\ell_{\text{tot}}) \leq \delta_n. \quad (51)$$

PROOF. For ease of notation, define $\eta_n := p\beta_n / \sum_{m=1}^N \beta_m$, for all n . By substituting (48) into (23)-(24), we have

$$D_n^*(t) = \sup_{0 \leq \tau \leq t} \left(-\eta_n Z^*(\tau) + U_n^*(\tau) \right)^+, \quad (52)$$

$$U_n^*(t) = \sup_{0 \leq \tau \leq t} \left(\eta_n Z^*(\tau) + D_n^*(\tau) - \ell_n \right)^+. \quad (53)$$

By comparing (52)-(53) with (29)-(30), it is easy to verify that $D_n^*(t) = \eta_n D^*(t; \ell_{\text{tot}})$ and $U_n^*(t) = \eta_n U^*(t; \ell_{\text{tot}})$ is the unique solution to (52)-(53). For each n , we can obtain the limit of $D_n^*(t)/t$:

$$\lim_{t \rightarrow \infty} \frac{D_n^*(t)}{t} = \lim_{t \rightarrow \infty} \frac{1}{t} \left(\frac{p\beta_n}{\sum_{m=1}^N \beta_m} D^*(t; \ell_{\text{tot}}) \right) \quad (54)$$

$$= \frac{\delta_n p}{\sum_{m=1}^N \delta_m} d^*(\ell_{\text{tot}}) \leq \delta_n, \quad (55)$$

where the last inequality in (55) follows from Corollary 1. \square

By Theorem 5, we know the necessary condition given by Corollary 1 is also sufficient. We summarize this result as follows.

THEOREM 6. *For any $(N + 1)$ -tuple $(\ell_{\text{tot}}, \delta_1, \dots, \delta_N)$ with $\ell_{\text{tot}} > 0$ and $\delta_n > 0$, for all n , under the Brownian approximation, the tuple is feasible if and only if $\frac{1}{p} \sum_{n=1}^N \delta_n \geq d^*(\ell_{\text{tot}})$.*

REMARK 4. Note that in Theorem 6, we only consider the case where $\delta_n > 0$, for every client n . Despite this, from an engineering perspective, we can get arbitrarily close to $\delta_n = 0$ by simply assigning an extremely small β_n to client n .

6.3 Choosing β_n for WLD Policy: Examples of Network Utility Maximization for QoE

In this section, we discuss how to properly choose weights $\{\beta_n\}$ for the WLD policy. In practice, the optimal $\{\beta_n\}$ can be determined by solving a network utility maximization (NUM) problem, which encodes the relative importance of the QoE performance of the clients. To demonstrate the connection between NUM and WLD, we briefly discuss the following examples of NUM problem for QoE:

Example 1 (Max-Min Fairness): Suppose the AP follows WLD with a predetermined latency budget ℓ_{tot} and is configured to minimize a network-wide QoE penalty function defined as: $f_1(\{D_n^*(t)\}) := \max_{1 \leq n \leq N} \{\limsup_{t \rightarrow \infty} D_n^*(t)/t\}$. By Theorem 5, this NUM can be converted into an equivalent optimization problem as:

$$\min_{(\ell_{\text{tot}}, \delta_1, \dots, \delta_N) \text{ is feasible}} \max_{n=1, \dots, N} \delta_n. \quad (56)$$

Note that (56) is a standard NUM for max-min fairness with a constraint induced by the capacity region for QoE. Therefore, it is easy to verify that the optimal solution to (56) is $\delta_n = d^*(\ell_{\text{tot}})p/N$, for every n . Moreover, by plugging this solution into Theorem 5, we know that $f_1(\{D_n^*(t)\})$ is minimized when $\limsup_{t \rightarrow \infty} D_n^*(t)/t = \limsup_{t \rightarrow \infty} D_m^*(t)/t$, for all n, m . Therefore, WLD can achieve the optimal QoE penalty by choosing $\beta_n = \beta_m$, for any pair of n, m , as suggested by Theorem 5. Moreover, under the total playback latency budget ℓ_{tot} , $\beta_n = \beta_m$ suggests that we choose $\ell_n = \ell_m$ (or equivalently $\ell_n = \ell_{\text{tot}}/N$).

Example 2 (Weighted Sum of Monomial Penalty): Let $\zeta_n > 0$ be the importance weight of each client n . The AP follows WLD policy with a predetermined latency budget ℓ_{tot} and is configured to minimize a network-wide QoE penalty function $f(\{D_n^*(t)\}) = \sum_{n=1}^N \zeta_n (\limsup_{t \rightarrow \infty} D_n^*(t)/t)^\kappa$, with some constant $\kappa > 1$. By Theorem 5, we can convert this NUM into an equivalent problem:

$$\min_{(\ell_{\text{tot}}, \delta_1, \dots, \delta_N) \text{ is feasible}} \sum_{n=1}^N \zeta_n \cdot \delta_n^\kappa. \quad (57)$$

It is easy to verify that for any $\kappa > 1$, the optimal solution to (57) is $\delta_n = (\zeta_n^{1/\kappa} / \sum_{m=1}^N \zeta_m^{1/\kappa}) d^*(\ell_{\text{tot}})p$, for every n . Again, by Theorem 5, WLD can achieve the optimal network utility by choosing $\beta_n = (\zeta_n^{1/\kappa} / \sum_{m=1}^N \zeta_m^{1/\kappa})$. Regarding the playback latency, WLD simply assigns $\ell_n = \beta_n \cdot \ell_{\text{tot}}$, for each n .

Based on these two examples, we know that the WLD policy can be easily configured to solve a broad class of NUM problems for QoE given the flexibility provided by the WLD policy.

7 ASYMPTOTIC RESULTS WITH RESPECT TO PLAYBACK LATENCY

In this section, we present simple asymptotic rules on the trade-off between playback latency and video interruption under the WLD policy. Recall that in (39)-(40), we discuss the ergodic property of the two-sided reflected Brownian motion. Based on Theorem 5, we know that the video interrupt rates under approximation (i.e. $\lim_{t \rightarrow \infty} D_n^*(t)/t$) exists and depends on the playback latency ℓ_n . To begin with, we consider the heavy-traffic regime, i.e. $\sum_{n=1}^N \lambda_n/p = 1$. The following theorem shows that the video interrupt rate is inversely proportional to the playback latency in heavy-traffic. We use the Little-Oh notation $o(1/\ell_n)$ to denote a function $g(\ell_n)$ that satisfies $\lim_{\ell_n \rightarrow \infty} g(\ell_n)/(1/\ell_n) = 0$.

THEOREM 7. *In the heavy-traffic regime, under the WLD policy, we have*

$$\lim_{t \rightarrow \infty} \frac{D_n^*(t)}{t} = \left(\frac{\sigma_n^2}{2\ell_n} \right) + o\left(\frac{1}{\ell_n} \right). \quad (58)$$

PROOF. This result can be directly obtained by plugging the variance of $Z_n^*(1)$ into [1, Theorem 12.1]. \square

Next, we turn to the under-loaded regime, where $\sum_{n=1}^N \lambda_n/p < 1$. The following theorem shows that the video interrupt rate under approximation decreases exponentially fast with the playback latency in the under-loaded regime.

THEOREM 8. *In the under-loaded regime, under the WLD policy, we have*

$$\lim_{t \rightarrow \infty} \frac{D_n^*(t)}{t} = c \exp\left(\frac{-2\epsilon_n}{\sigma_n^2} \ell_n\right) + o\left(\exp\left(\frac{-2\epsilon_n}{\sigma_n^2} \ell_n\right)\right), \quad (59)$$

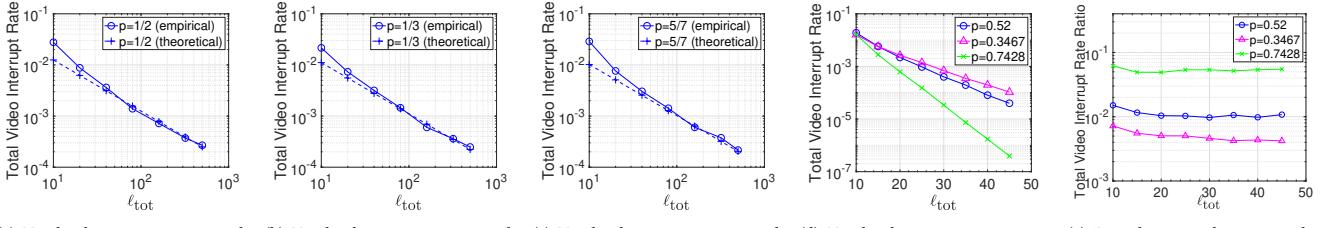
where c is some constant that does not depend on ℓ_n .

PROOF. By [1, Theorem 3.1], this result can be directly obtained by finding the root γ of the Lundberg equation $\mathbb{E}[\exp(\gamma Z_n^*(1))] = 1$. As $Z_n^*(1)$ is a Gaussian random variable with mean ϵ_n and variance σ_n^2 (defined in (49)-(50)), it is easy to verify that $\gamma = -2\epsilon_n/\sigma_n^2$. \square

REMARK 5. Note that a one-dimensional one-sided reflected Brownian motion with negative drift has a stationary distribution, which is exponential [3, Theorem 6.2]. In the under-loaded regime, as shown by Theorem 8, a two-sided reflected Brownian motion also exhibits a similar behavior as the one-sided reflected counterpart.

8 NUMERICAL SIMULATIONS

In this section, we present the simulation results of the proposed policy. Throughout the simulations, we consider a network of one AP and 5 video clients. All the simulation results presented below are the average of 50 simulation trials.



(a) Total video interrupt rate under different ℓ_{tot} : $p = 1/2$. (b) Total video interrupt rate under different ℓ_{tot} : $p = 1/3$. (c) Total video interrupt rate under different ℓ_{tot} : $p = 5/7$. (d) Total video interrupt rates versus ℓ_{tot} in the under-loaded regime: (e) Ratio between the empirical total video interrupt rate and the theoretical estimate: $p = 0.52, 0.3467, 0.7428$.

Figure 4: Evaluation of the approximation accuracy in the heavy-traffic and under-loaded regimes.

8.1 Accuracy of the Approximation

We first evaluate the accuracy of the proposed approximation under the WLD policy. We consider a fully-symmetric network of 5 video clients, where $\ell_n = \ell_{\text{tot}}/5$, for every n . In this case, WLD shall choose $\beta_n = 1/5$, for every client. We consider three heavy-traffic scenarios with $p = 1/2, 1/3, 5/7$ and $\lambda_n = 1/10, 1/15, 1/7$, respectively. To verify the accuracy of the approximation, Figure 4(a)-4(c) show the total video interrupt rates (i.e. $\sum_{n=1}^N D_n(t)/t$) under different playback latency budgets and different channel reliabilities in the heavy-traffic regime. Note that both the x-axis and y-axis are in log scale. We also plot the theoretical estimates of the total video interrupt rates based on Theorem 7 (by (36), we know $\sigma^2 = 1, 2$, and 0.4 for $p = 1/2, 1/3$, and $5/7$, respectively). It can be observed that the empirical rates are very close to the theoretical estimates, and the difference shrinks with the playback latency budget. This is consistent with the asymptotic results in Theorem 7.

Next, we turn to the under-loaded case. We consider three under-loaded scenarios with $p = 0.52, 0.3467, 0.7428$ and $\lambda_n = 1/10, 1/15, 1/7$, respectively. Figure 4(d) shows the total video interrupt rates under different ℓ_{tot} and channel reliabilities (note that the y-axis is in log scale and the x-axis is in linear scale). We can observe that the dependency of empirical rates on ℓ_{tot} is roughly log-linear, as suggested by Theorem 8. To further verify the accuracy of the theoretical estimates provided by Theorem 8, Figure 4(e) plots the ratio between the empirical total interrupt rate and the asymptotic term in (59), i.e. $(\sum_{n=1}^N D_n(t)/t)/(N \exp(\frac{-2\epsilon_n}{\sigma_n^2} \ell_n))$, under different channel reliabilities. We observe that under different ℓ_{tot} , this ratio stays at around 0.01, 0.005, and 0.05 under $p = 0.52, 0.3467$, and 0.7428, respectively. Hence, Figure 4(e) verifies the accuracy of the approximation in the under-loaded regime.

In summary, all the above results suggest that the approximation $D_n(t) \approx D_n^*(t)$ is rather accurate in both heavy-traffic and under-loaded regimes, even with small to moderate latency budgets.

8.2 Comparison With Other Policies

We evaluate the proposed WLD policy against four baseline policies, namely Weighted Random (WRand), Weighted Round Robin (WRR), Earliest Deadline First (EDF), and the Delivery-Based Largest-Debt-First (DBLDF). Under the WRand policy, in each time slot, the AP simply schedules each client n with probability $\lambda_n/\sum_{m=1}^N \lambda_m$. Under the WRR policy, the AP groups multiple time slots into a frame and schedules the clients in a cyclic manner within each frame. Specifically, in each frame, each client n is scheduled for exactly $K\lambda_n/\sum_{m=1}^N \lambda_m$ times, where K is chosen to be the smallest positive

Table 1: QoE penalty and video interruptions under $p = 0.6$, $\ell_{\text{tot}} = 32$ at both $t = 1.5 \times 10^5$ and $t = 3 \times 10^5$ (separated by '|').

Policy	QoE penalty ($\times 10^5$)	Per-client video interruptions	
		Group 1	Group 2
WLD	1.5 5.7	134.0 264.5	158.2 309.3
DBLDF	4.9 19.4	265.3 527.4	264.5 526.7
EDF	14.0 55.7	538.9 1074.1	284.3 565.1
WRR	138.0 550.9	1844.4 3684.3	255.6 513.2
WRand	368.5 1480.2	2994.2 6002.0	573.4 1143.6

Table 2: QoE penalty and video interruptions under $p = 0.65$, $\ell_{\text{tot}} = 32$ at both $t = 1.5 \times 10^5$ and $t = 3 \times 10^5$ (separated by '|').

Policy	QoE penalty ($\times 10^3$)	Per-client video interruptions	
		Group 1	Group 2
WLD	0.01 0.02	1.3 2.3	0.2 0.2
DBLDF	0.19 0.58	5.5 9.4	4.8 8.7
EDF	6.9 27.7	37.7 75.5	20.2 40.4
WRR	2813.7 11344.9	838.1 1683.0	35.8 69.0
WRand	15927.9 63745.6	1982.9 3966.7	258.7 518.8

integer such that $K\lambda_n/\sum_{m=1}^N \lambda_m$ is an integer, for all n . Under the EDF policy, the AP schedules the video packet with the smallest absolute deadline among all the video packets in the AP-side buffers, with ties broken randomly. The EDF policy is widely used in real-time systems given its strong theoretical guarantee for deadline-constrained tasks [18]. Under DBLDF, the AP schedules the client with the largest delivery debt, which is defined as $\lambda_n t - A_n(t)$. Different from WLD, DBLDF tracks only the delivery of video packets and is completely oblivious to the dummy packets. Note that the delivery-debt index was proposed and analyzed in [9] for the frame-synchronized real-time wireless networks. We evaluate the WLD policy as well as the four baseline policies in both heavy-traffic and under-loaded regimes.

To showcase the performance of the proposed policy, we start with the following heavy-traffic scenario: The 5 video clients are divided into two groups: clients 1 and 2 are in Group 1, and clients 3, 4, and 5 belong to Group 2. We consider $\lambda_n = 1/5$ for Group 1 and $\lambda_n = 1/15$ for Group 2. We set $p = 0.6$ and $\ell_{\text{tot}} = 32$. It is easy to verify that $\sum_{n=1}^N \lambda_n/p = 1$. We consider a quadratic QoE

penalty function as $f(\{D_n(t)\}) = \sum_{n=1}^5 \zeta_n (\limsup_{t \rightarrow \infty} D_n(t)/t)^2$ with $\zeta_1 = \zeta_2 = 2$ and $\zeta_3 = \zeta_4 = \zeta_5 = 1$. As described by Example 2 in Section 6.3, for the WLD policy, we choose $\beta_n = 1/8$ and $\ell_n = \ell_{\text{tot}}/8 = 4$ for each client in Group 1 and $\beta_n = 2/8$ and $\ell_n = \ell_{\text{tot}}*(2/8) = 8$ for each client in Group 2. For a fair comparison, we use the same playback latency for all the policies.

Table 1 shows the QoE penalty and the average video interruptions per client in each group at both $t = 1.5 \times 10^5$ and $t = 3 \times 10^5$ (values separated by '|'). Due to space limitation, the figures of the complete evolution of video interruptions are presented in Appendix A.5 of the technical report [12]. We observe that WLD achieves the least amount of video interruptions among all the policies, for both Group 1 and Group 2. Both WRR and WRand have much more video interruptions as they are not responsive to the buffer status. On the other hand, compared to WLD, EDF policy has about 4 times and twice of video interruptions for Group 1 and Group 2, respectively. This is mainly because the design of EDF does not take the existence and heterogeneity of the playback latency into account and is also completely oblivious to the target QoE penalty function. Under WLD, as expected from the choice of β_n , each client in Group 1 has only about 80% of the video interruptions experienced by a client in Group 2 (the slight mismatch in this ratio comes from the effect of a small ℓ_n , similar to the effect described in Figure 4(a)-4(c)). Moreover, compared to WLD, DBLDF has about 2 times of the video interruptions for both groups. This shows that it is indeed sub-optimal to keep track of only the delivery of video packets and ignore the dummy packets. The above results verify that WLD can achieve the optimal network utility by choosing the proper parameters β_n .

Next, we repeat the same experiments but in the under-loaded regime. We set $p = 0.65$ and keep the other parameters identical to those for Table 1. Table 2 shows the performance in terms of video interruption and QoE penalty in the under-loaded regime. The figures of the complete evolution of video interruptions are presented in Appendix A.5 of the technical report [12]. Similar to the heavy-traffic setting, the baseline policies have much more video interruptions than WLD. Note that in this case, WLD has almost zero video interruptions for both groups as the video interrupt rate decreases much faster with the playback latency in the under-loaded regime, as suggested by Theorem 8.

9 CONCLUSION

This paper studies the critical trade-off between playback latency and video interruption, which are the two most critical QoE metrics for real-time video streaming. With the proposed analytical model and the Brownian approximation scheme, we study the fundamental limits of the latency-interruption trade-off and thereby design a QoE-optimal scheduling policy. Through both rigorous analysis and extensive simulations, we show that the proposed approximation framework can capture the original playback processes very accurately and offer simple design rules on the interplay between playback latency and video interruption.

ACKNOWLEDGMENTS

This material is based upon work supported in part by the Ministry of Science and Technology of Taiwan under Contract No. MOST 108-2636-E-009-014, in part by NSF and Intel under contract number CNS-1719384, in part by the U.S. Army Research Laboratory and the

U.S. Army Research Office under contract/Grant Number W911NF-18-1-0331, and in part by Office of Naval Research under Contract N00014-18-1-2048.

REFERENCES

- [1] Lars Nørvang Andersen, Søren Asmussen, Peter W Glynn, and Mats Pihlgård. 2015. Lévy processes with two-sided reflection. In *Lévy Matters V*. Springer, 67–182.
- [2] Rami Atar and Mark Shifrin. 2015. An asymptotic optimality result for the multiclass queue with finite buffers in heavy traffic. *Stochastic Systems* 4, 2 (2015), 556–603.
- [3] Hong Chen and David D Yao. 2001. *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*. Vol. 46. Springer.
- [4] Cisco. 2017. Video Quality of Service (QoS) Tutorial. <https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-video/212134-Video-Quality-of-Service-QoS-Tutorial.pdf>.
- [5] Lei Deng, Chih-Chun Wang, Minghua Chen, and Shizhen Zhao. 2017. Timely wireless flows with general traffic patterns: Capacity region and scheduling algorithms. *IEEE/ACM Transactions on Networking* 25, 6 (2017), 3473–3486.
- [6] J Michael Harrison. 1988. Brownian models of queueing networks with heterogeneous customer populations. (1988), 147–186.
- [7] J Michael Harrison. 2000. Brownian models of open processing networks: Canonical representation of workload. *Annals of Applied Probability* (2000), 75–103.
- [8] J Michael Harrison and Jan A Van Mieghem. 1997. Dynamic control of Brownian networks: state space collapse and equivalent workload formulations. *The Annals of Applied Probability* (1997), 747–771.
- [9] IH Hou, V Borkar, and PR Kumar. 2009. A Theory of QoS for Wireless. *Proc. of IEEE INFOCOM* (2009), 486–494.
- [10] I-Hong Hou and Ping-Chun Hsieh. 2017. The capacity of QoE for wireless networks with unreliable transmissions. *Queueing Systems* 87 (2017), 131–159.
- [11] Ping-Chun Hsieh and I-Hong Hou. 2018. Heavy-traffic analysis of QoE optimality for on-demand video streams over fading channels. *IEEE/ACM Transactions on Networking* 26, 4 (2018), 1768–1781.
- [12] Ping-Chun Hsieh, Xi Liu, and I-Hong Hou. 2019. Fresher Content or Smoother Playback? A Brownian-Approximation Framework for Scheduling Real-Time Wireless Video Streams. <https://arxiv.org/abs/1911.00902>.
- [13] Vinay Joseph and Gustavo de Veciana. 2014. NOVA: QoE-driven optimization of DASH-based video delivery in networks. In *Proc. of IEEE INFOCOM*. 82–90.
- [14] Xiaohan Kang, Weina Wang, Juan José Jaramillo, and Lei Ying. 2014. On the performance of largest-deficit-first for scheduling real-time traffic in wireless networks. *IEEE/ACM Transactions on Networking* 24, 1 (2014), 72–84.
- [15] Kye Seob Kim, Chih-Ping Li, Igor Kadota, and Eytan Modiano. 2015. Optimal scheduling of real-time traffic in wireless networks with delayed feedback. In *Proc. of Allerton*. 1143–1149.
- [16] Kye Seob Kim, Chih-ping Li, and Eytan Modiano. 2014. Scheduling multicast traffic with deadlines in wireless networks. In *Proc. of IEEE INFOCOM*. 2193–2201.
- [17] Lukasz Kruck, John Lehoczky, Kavita Ramanan, Steven Shreve, et al. 2008. Double Skorokhod map and reneging real-time queues. In *Markov Processes and Related Topics: A Festschrift for Thomas G. Kurtz*. 169–193.
- [18] Phillip A Laplante. 2004. Real-time systems design and analysis. Wiley.
- [19] Bin Li and Attila Eryilmaz. 2013. Optimal distributed scheduling under time-varying conditions: A fast-CSMA algorithm with applications. *IEEE Transactions on Wireless Communications* 12, 7 (2013), 3278–3288.
- [20] Guanfeng Liang and Ben Liang. 2008. Effect of delay and buffering on jitter-free streaming over random VBR channels. *IEEE Transactions on Multimedia* 10, 6 (2008), 1128–1141.
- [21] Tom H Luan, Lin X Cai, and Xuemin Shen. 2010. Impact of network dynamics on user's video quality: Analytical framework and QoS provision. *IEEE Transactions on Multimedia* 12, 1 (2010), 64–78.
- [22] Ali ParandehGheibi, Muriel Médard, Asuman Ozdaglar, and Srinivas Shakkottai. 2011. Avoiding interruptions—A QoE reliability function for streaming media applications. *IEEE Journal on Selected Areas in Communications* 29, 5 (2011), 1064–1074.
- [23] Moshe Shaked and J George Shanthikumar. 2007. *Stochastic orders*. Springer.
- [24] Alexander L Stolyar et al. 2004. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability* 14, 1 (2004), 1–53.
- [25] Ward Whitt. 2002. *Stochastic-process limits: an introduction to stochastic-process limits and their application to queues*. Springer Science & Business Media.
- [26] Yuedong Xu, Eitan Altman, Rachid El-Azouzi, Majed Haddad, Salaheddine Elayoubi, and Tania Jimenez. 2014. Analysis of buffer starvation with application to objective QoE optimization of streaming services. *IEEE Transactions on Multimedia* 16, 3 (2014), 813–827.
- [27] Yuedong Xu, Salah Eddine Elayoubi, Eitan Altman, and Rachid El-Azouzi. 2013. Impact of flow-level dynamics on QoE of video streaming in wireless networks. In *Proc. of IEEE INFOCOM*. 2715–2723.