

Media Streaming via TFRC: An Analytical Study of the Impact of TFRC on User-Perceived Media Quality

Lisong Xu, Josh Helzer

Department of Computer Science and Engineering
University of Nebraska, Lincoln, NE 68588-0115

Email: {xu, jhelzer}@cse.unl.edu

Abstract—TCP-Friendly Rate Control (TFRC) is being adopted in Internet standards for congestion control of various streaming media applications. In this paper, we consider the transmission of pre-recorded media from a server to a client by using TFRC, and analytically study the impact of TFRC on user-perceived media quality, which is roughly measured by calculating the rebuffering probability. A rebuffering probability is defined to be the probability that the total duration of all rebuffering events experienced by a user is longer than a certain threshold. Two approaches are presented to help an application determine an appropriate initial buffering delay and media playback rate in order to achieve a certain rebuffering probability under a given network condition. First, we derive a closed-form expression to approximate the average TFRC sending rate, which could be used as the maximum allowed playback rate of a media stream. Second, we develop a queueing model for a TFRC client buffer with the traffic described by a Markov-Renewal-Modulated Deterministic Process (MRMDP), and present an iterative method to calculate the rebuffering probability.

I. INTRODUCTION

In the past few years we have witnessed an explosive growth in the usage of streaming media applications such as Real Player, Windows Media Player, and QuickTime Player, which enable a user to start playing an audio or video stream without downloading it entirely from a remote media source. According to a recent industry study [1], 14.2 billion video streams were served in 2004, an 80% increase compared to 2003, and the number is forecast to reach 21 billion in 2005.

The rapid growth in streaming media usage has heightened the need for a good transport protocol for streaming media, which is loss tolerant but delay sensitive [12]. TCP, the current dominant transport protocol, is not well-suited for streaming media, because its reliable transmission mechanisms and abrupt rate drops in response to packet loss may introduce undesirable packet delay.

Currently, TCP-Friendly Rate Control (TFRC) [5] is being adopted in Internet standards [6], [7], [8] for congestion control of various streaming media applications. TFRC maintains approximately the same average sending rate as TCP running under comparable network conditions, while providing a relatively smooth sending rate, which helps packets to meet the real-time constraints required by streaming media. It could be implemented by a media streaming application, or be incorporated into the popular UDP-based real-time transport protocol RTP [24].

Even though the performance of TFRC has been extensively studied [5], [4], [29], [21], [25], [38], [34], [11], [33], little is

known about its quantitative impact on user-perceived media quality. In this paper, we develop an analytical model for a TFRC client buffer, and attempt to answer the following question: *Under what conditions can TFRC provide satisfactory user-perceived media quality?*

Various metrics [10], [16], [32], [31], [15], [27] have been proposed to measure user-perceived media quality. However, there is still no single metric efficient and comprehensive enough to assess user playback experience, since it is jointly affected by various network, application, and human factors. In this paper, we roughly evaluate user-perceived media quality by measuring the occurrence of rebuffering events. A *rebuffering event* occurs when the media data in a TFRC client buffer has been exhausted, and the player is forced to stop playback until enough data has been received to resume. Intuitively, the greater the number and duration of rebuffering events, the worse the user-perceived media quality. Since a rebuffering event is relatively easy to measure (compared with extensive physiological experiments required by other metrics [10]), and since it also captures some important aspects of playback experience, it has been widely used as a metric for user-perceived media quality [23], [30], [32].

The *main contribution of our work* is the proposed analytical model for a TFRC client buffer, which enables us to calculate the rebuffering probability based on the parameters of the network, streaming media, and TFRC client buffer. A *rebuffering probability* is defined to be the probability that the total duration of all rebuffering events experienced by a user is longer than a certain threshold. Specifically, we

- derive a closed-form expression to approximate the average sending rate of a TFRC stream, based on the work presented in [29] and [21]. The obtained average TFRC sending rate can be used as the maximum allowed playback rate of a media stream.
- present a queueing model for prerecorded media streaming via TFRC, where the TFRC traffic is described by a Markov-Renewal-Modulated Deterministic Process (MRMDP), and the TFRC client buffer is modeled as an MRMDP/D/1 queueing node.
- develop an iterative algorithm to measure the rebuffering probability of an MRMDP/D/1 queueing node. This algorithm can be used by a streaming media application to find an appropriate initial buffering delay and media playback rate to provide better playback experience.

TABLE I
KEY NOTATION

Notation	Description
p	loss event rate (1/Packets)
θ_n	the n^{th} loss interval: the number of packets sent between the n^{th} and $(n+1)^{\text{th}}$ loss events (Packets)
$\hat{\theta}_n$	the n^{th} weighted average loss interval (Packets)
ϕ_i	traffic rate associated with MRMDP state i (Packets/Second)
ϕ_{tfrc}	time-average TFRC sending rate (Packets/Second)
ϕ_{media}	media playback rate (Packets/Second)

The following two examples demonstrate some potential applications of our work. Example 1: Determining an initial buffering delay. Let x denote the number of packets that a player buffers before a stream plays back. While a large x helps to improve user playback experience by diminishing rebuffering events, it increases the undesired initial buffering delay that a user has to wait prior to playback. Given the current network delay and loss event rate, our work can be used to find the *optimal* x that minimizes the delay while maintaining a certain rebuffering probability. The network delay and loss event rate can be estimated either by using the cached information of previous streams, or by applying some measurement techniques [9], [2], [26], which are beyond the scope of this paper.

Example 2: Selecting from multiple-rate streams. A recent survey [14] of the streaming audio and video stored on the Internet reports that a significant fraction of media is encoded with multiple-rate encoding techniques, such as Microsoft Intelligent Streaming, and RealNetworks Sure Stream, which encode the same content as multiple streams with different rates for networks with different bandwidths. For a given network condition, our work can be used to find the *optimal* stream that has the maximum rate while achieving a certain rebuffering probability.

The rest of the paper is organized as follows. Section II gives the problem setting. Section III presents a TFRC traffic model. Section IV proposes a queueing model for a TFRC client buffer, and describes an iterative algorithm to calculate the rebuffering probability. Section V shows some numerical results. Related work is reviewed in Section VI, and we conclude the paper by Section VII.

II. PROBLEM SETTING

This section describes the problem setting of modeling multimedia streaming via TFRC. The key notation used throughout the paper is summarized in Table I.

A. TCP-Friendly Rate Control

The sending rate of TFRC [6] is adjusted according to Eq. (1), which is a simplified version of the TCP throughput function [17] when assuming $p < 0.54$ and no delayed-ack.

$$R(p) = \frac{1}{t_{rtt}\sqrt{\frac{2p}{3}} + t_{rto}3\sqrt{\frac{3p}{8}}p(1+32p^2)} \quad (1)$$

As described in [6], a *loss event* is defined to be one or more lost or ECN-marked packets within one round-trip time (RTT). Loss event rate p is the number of loss events as a fraction of the total number of transmitted packets. t_{rtt} is the exponentially weighted moving average of RTTs. At every RTT, t_{rtt} is updated by $t_{rtt} = 0.9t_{rtt} + 0.1t$, where t is the latest RTT sample. Since t_{rtt} is updated every RTT, and changes slowly due to the large constant 0.9, we assume that t_{rtt} is a constant for a TFRC flow. Retransmission timeout period t_{rto} is usually set to $4t_{rtt}$.

Following the notation in [29], we denote θ_n to be the n^{th} *loss interval*, which is the number of packets sent between the n^{th} and $(n+1)^{\text{th}}$ loss events, and denote $\hat{\theta}_n$ to be the *weighted average loss interval* calculated by using the 8 most recent loss intervals as follows.

$$\hat{\theta}_n = \frac{\theta_{n-1}}{6} + \frac{\theta_{n-2}}{6} + \frac{\theta_{n-3}}{6} + \frac{\theta_{n-4}}{6} + \frac{\theta_{n-5}}{7.5} + \frac{\theta_{n-6}}{10} + \frac{\theta_{n-7}}{15} + \frac{\theta_{n-8}}{30} \quad (2)$$

In this paper, we consider only the *basic control* [29] of TFRC, where TFRC sets its sending rate to the rate calculated by the formula (i.e. $R(\frac{1}{\hat{\theta}_n})$) at the time when the n^{th} loss event is detected by the TFRC server. Vojnović and Le Boudec [29] show that the average sending rate ϕ_{tfrc} of a TFRC flow with basic control can be calculated as follows, if successive loss intervals are independent of each other.

$$\phi_{tfrc} = \frac{E[\theta_n]}{E\left[\frac{\theta_n}{R(\frac{1}{\hat{\theta}_n})}\right]} = \frac{1}{E\left[\frac{1}{R(\frac{1}{\hat{\theta}_n})}\right]} \quad (3)$$

Vojnović and Le Boudec [29] also prove that if TFRC and TCP experience the same loss event rate $p = \frac{1}{E[\hat{\theta}_n]}$, then the average sending rate ϕ_{tfrc} of TFRC is less than the average sending rate ϕ_{tcp} of TCP. That is,

$$\phi_{tfrc} < \phi_{tcp} = R\left(\frac{1}{E[\hat{\theta}_n]}\right) \quad (4)$$

Rhee and Xu [21] further present an approximation to the average TFRC sending rate as follows. Note that due to the convexity of $\frac{1}{\theta_n}$, we have $\frac{1}{E[\hat{\theta}_n]} \neq E\left[\frac{1}{\hat{\theta}_n}\right]$.

$$\phi_{tfrc} \approx R\left(E\left[\frac{1}{\hat{\theta}_n}\right]\right) \quad (5)$$

B. Distribution of TFRC Loss Intervals

It is well known [39] that the packet losses of a flow are not independently and identically distributed (IID), due to the strong correlation between consecutive packet losses. However, since all packet losses within the same RTT is counted as a single loss event, it is reasonable to consider that loss events are IID. Furthermore, a recent study [39] on a large measurement dataset of Internet traffic shows that loss events can be well modeled as a Poisson process. Since their dataset does not include TFRC traffic, we have conducted some experiments by running TFRC [35] over PlanetLab [20] to study the characteristic of TFRC loss intervals.

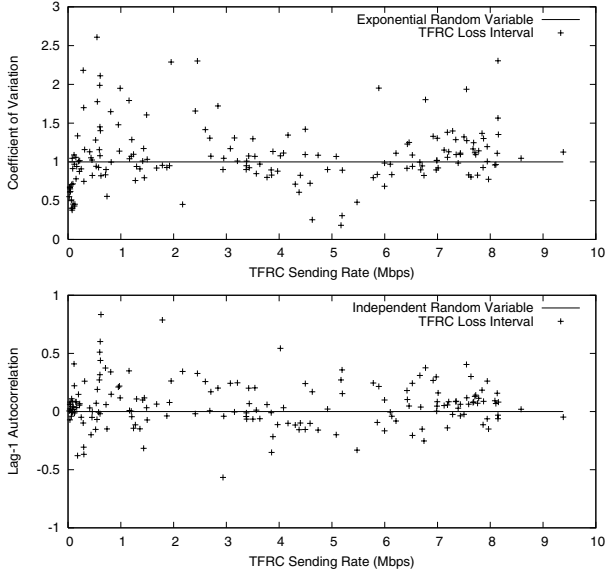


Fig. 1. CoV and Lag-1 autocorrelation of TFRC loss intervals measured by running TFRC over various PlanetLab nodes

Three batches of experiments were performed at different times. The first batch consisted of 99 15-minute transfers between endpoints chosen at random from a pool of 302 PlanetLab nodes. This set of endpoints was used to repeat the tests for the second and third batches. Due to the transient nature of PlanetLab nodes, only 65 of the original 99 transfers were successfully run in all three batches.

For each successful experiment, we measure the coefficient of the variation (CoV) of TFRC loss intervals, where CoV is the standard deviation normalized by the mean value. Figure 1.a shows that most of the CoV of TFRC loss intervals are very close to that of an exponential random variable (i.e. one). For each successful experiment, we also measure the lag-1 autocorrelation [19] of TFRC loss intervals, which can be used as an indication of the dependence of consecutive loss intervals. The larger the absolute value, the stronger the dependence. Figure 1.b shows most of the lag-1 autocorrelations of TFRC loss intervals are very close to that of independent random variables (i.e. zero). We notice that for both figures, the experiments with very large CoV or lag-1 autocorrelation usually correspond to international connections between two countries, such as between US and China.

Overall, our experiment results confirm the finding by Zhang et al. [39] that loss intervals could be considered as independent and identical exponential random variables. Therefore, in this paper we assume that the cumulative distribution function and probability density function of θ_n are given by the following equations, respectively.

$$F_{\theta_n}(k) = P(\theta_n \leq k) = 1 - e^{-pk} \quad (6)$$

$$f_{\theta_n}(k) = \frac{dF_{\theta_n}(k)}{dk} = pe^{-pk} \quad (7)$$

which also implies we assume that loss interval θ_n could be any positive real number.

C. Media Streaming

Since we are interested primarily in the effectiveness of TFRC as a congestion control protocol for streaming media by providing a relatively smooth sending rate, we focus on the impact of the variation of TFRC sending rates on a rebuffering probability, and assume that all packets experience the same network delay. We do not consider the impact of lost packets and network delay jitter, assuming that the lost packets can be recovered or concealed by applying error concealment techniques [18], [22], and the effect of network delay jitter can be mitigated by using playout adaptation techniques [13]. Developing more advanced models that can remove these assumptions is of future interest.

We consider the transmission of *pre-recorded* media from a server to a client by using TFRC, where the server is assumed to transmit the media as fast as the calculated TFRC sending rate. The media duration is assumed to be infinite. Our experiments presented in Section V show that a media stream with a sufficiently long duration (say 400 seconds) has almost the same rebuffering probability as a stream with an infinite duration, and therefore can be analyzed as if its duration is infinite.

We consider a TFRC client buffer as a fluid buffer with an infinite capacity, where fluid arrives at a TFRC governed rate, and drains off at a fixed rate of ϕ_{media} . We consider only a fixed playback rate ϕ_{media} , since a recent study [14] shows most streaming media on the Internet is encoded with a constant bit rate (CBR). The fluid model implies we assume that the number of packets in a TFRC client buffer could be any non-negative real number. As we will see in the next few sections, this fluid model greatly simplifies the analysis while maintaining good accuracy.

III. MODELING TFRC TRAFFIC

In this section, we model TFRC traffic with a Markov-Renewal-Modulated Deterministic Process (MRMDP), which captures some important properties of TFRC. The proposed MRMDP is then used to develop a queueing model for a TFRC client buffer, described in the next two sections.

A. Markov-Renewal-Modulated Deterministic Processes

We propose a *Markov-Renewal-Modulated Deterministic Process* (MRMDP) as an extension of a traditional Markov-Modulated Deterministic Process (MMDP), which is introduced to represent correlated deterministic traffic, such as the superposition of ATM CBR traffic [37]. The traffic rate of an MRMDP with m states (denoted by m -MRMDP thereafter) is modulated by a Markov renewal process $\{(S_n, T_n), n = 0, 1, \dots\}$ ¹, where random variable $S_n \in [0, m-1]$ is the MRMDP state immediately after the n^{th} state transition point, and random variable $T_n \in [0, \infty)$ is the time instant when the n^{th} state transition occurs. A 2-MRMDP (i.e. an MRMDP with 2 states) is illustrated in Figure 2. State i ($i \in [0, m-1]$)

¹A stochastic process $\{(S_n, T_n), n = 0, 1, \dots\}$ is a Markov renewal process, if $P(S_{n+1} = j, T_{n+1} - T_n \leq t | S_n, S_{n-1}, \dots, T_n, T_{n-1}, \dots) = P(S_{n+1} = j, T_{n+1} - T_n \leq t | S_n)$

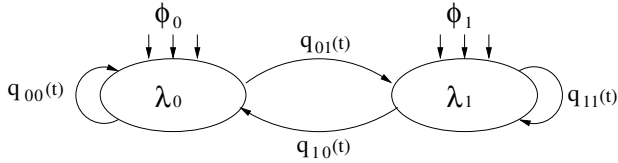


Fig. 2. A 2-MRMDP (i.e. an MRMDP with two states)

of an MRMDP is completely described by the following three types of parameters.

- $\frac{1}{\lambda_i}$: the expected duration of MRMDP state i . If $S_n = i$, then τ_n is an exponential random variable with an expected value of $\frac{1}{\lambda_i}$, where $\tau_n = T_{n+1} - T_n$.
- ϕ_i : the fixed traffic rate associated with MRMDP state i . If $S_n = i$, then the traffic rate is fixed at ϕ_i during interval $[T_n, T_{n+1})$.
- $q_{ij}(t)$: transition probability from MRMDP state i to j with the time spent at state i longer than t . The next state S_{n+1} of an MRMDP depends on both the current state S_n and the current state duration τ_n . For a time-homogeneous MRMDP, the state transition probability is independent of n . In this paper, we define $q_{ij}(t)$ of a time-homogeneous MRMDP by

$$q_{ij}(t) = P(t < \tau_n, S_{n+1} = j | S_n = i) \quad (8)$$

We also define $q_{ij} = q_{ij}(0)$. Since τ_n is always greater than zero, we have

$$q_{ij} = P(S_{n+1} = j | S_n = i) \quad (9)$$

Our proposed MRMDP differs from the traditional MMDP in their state transition probabilities. The former captures the correlation between S_{n+1} and τ_n , whereas the latter assumes that S_{n+1} and τ_n are independent. This difference makes an MRMDP more appropriate for TFRC, since the next sending rate of TFRC depends on both the current sending rate and the current loss interval size.

B. An MRMDP for TFRC Traffic

In this subsection, we consider the following two questions in modeling TFRC traffic with an m -MRMDP. 1) What is the duration of an MRMDP state for TFRC traffic? 2) What are the m MRMDP states for TFRC traffic?

What is the duration of an MRMDP state for TFRC traffic? As described in Section II, we consider only the basic control mode, where TFRC sends packets at a fixed rate until the next loss event. Since TFRC changes its sending rates only at a loss event, we can consider a loss event as an MRMDP state transition point, and then the size of a loss interval corresponds to the duration of an MRMDP state, as illustrated in Figure 3. If $S_n = i$, then an MRMDP sends packets at a fixed rate of ϕ_i . The duration of the n^{th} loss interval is given by

$$\tau_n = \frac{\theta_n}{\phi_i} \quad (10)$$

where θ_n is the total number of packets sent during the n^{th} loss interval. Since θ_n is exponentially distributed with an expected

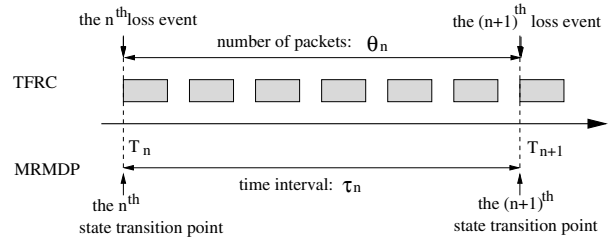


Fig. 3. An MRMDP for TFRC traffic

value of $\frac{1}{p}$, and ϕ_i is a constant, τ_n is an exponential random variable with an expected value of $\frac{1}{\phi_i p}$. It follows

$$\lambda_i = \phi_i p \quad (11)$$

What are the m MRMDP states for TFRC traffic? An m -MRMDP is associated with m sending rates, one for each state. However, $\hat{\theta}$, the weighted average loss interval, has a possibly infinite number of values, thus so does the sending rate. Therefore, we have to partition the set of all possible $\hat{\theta}$ values into m subsets, and then assign a single fixed sending rate with each subset. Define $\hat{\Theta} = \{\hat{\theta} | \hat{\theta} > 0\}$. Below, we discuss our approach to partition $\hat{\Theta}$ into m subsets $\hat{\Theta}_0, \dots, \hat{\Theta}_{m-1}$. If $\hat{\theta}_n$, the n^{th} weighted average loss interval, belongs to $\hat{\Theta}_i$, then we consider that $S_n = i$.

Inspired by the work of [37] and [3], we first partition $\hat{\Theta}$ into two subsets: $\hat{\Theta}_{\text{under}}$ and $\hat{\Theta}_{\text{over}}$. $\hat{\Theta}_{\text{under}}$ corresponds to the case where the sending rate is lower than or equal to the media playback rate, and $\hat{\Theta}_{\text{over}}$ corresponds to the case where the sending rate is higher than the media playback rate. Let ϕ_{media} denote the media playback rate, and define $\hat{\theta}_{\text{media}} = \frac{1}{R^{-1}(\phi_{\text{media}})}$, where $R^{-1}(\cdot)$ is the inverse function of $R(\cdot)$. We have

$$\hat{\Theta}_{\text{under}} = \{\hat{\theta} | \hat{\theta} \leq \hat{\theta}_{\text{media}}\} \quad \hat{\Theta}_{\text{over}} = \{\hat{\theta} | \hat{\theta} > \hat{\theta}_{\text{media}}\}$$

$\hat{\Theta}_{\text{under}}$ is further divided into $\frac{m}{2}$ non-overlapping subsets: $\hat{\Theta}_0, \hat{\Theta}_1, \dots, \hat{\Theta}_{\frac{m}{2}-1}$, where $\hat{\Theta}_i = \{\hat{\theta} | b_i < \hat{\theta} \leq b_{i+1}\}$ and

$$b_i = i \cdot \frac{\hat{\theta}_{\text{media}}}{m/2} \quad i \leq \frac{m}{2} \quad (12)$$

$\hat{\Theta}_{\text{over}}$ is also divided into $\frac{m}{2}$ non-overlapping subsets: $\hat{\Theta}_{\frac{m}{2}+1}, \dots, \hat{\Theta}_{m-1}$, where $\hat{\Theta}_i = \{\hat{\theta} | b_i < \hat{\theta} \leq b_{i+1}\}$ and

$$b_i = \hat{\theta}_{\text{media}} + (i - \frac{m}{2}) \frac{\hat{\theta}_{\text{max}} - \hat{\theta}_{\text{media}}}{m/2} \quad \frac{m}{2} \leq i < m \quad (13)$$

$$b_m = \infty \quad (14)$$

$\hat{\theta}_{\text{max}}$ is a number larger than most $\hat{\theta}$ values (e.g. $\frac{2}{p}$, since the average is $\frac{1}{p}$). The reason for using $\hat{\theta}_{\text{max}}$ is that $\hat{\Theta}_{\text{over}}$ does not have a maximum number. Note that our partition method requires that m is an even number no less than 2.

Intuitively, the larger the value of m , the better the accuracy of the model. However, as m increases, the time and space complexity of the analysis increases very quickly.

In the next few subsections, we describe how to assign a fixed sending rate ϕ_i with each $\hat{\Theta}_i$, and how to calculate the state transition probability $q_{ij}(t)$.

C. Distribution of the Weighted Average Loss Interval

In order to determine ϕ_i and $q_{ij}(t)$, we first obtain the distribution of $\hat{\theta}_n$, the weighted average loss interval. The cumulative distribution function $F_{\hat{\theta}_n}(k)$ and probability density function $f_{\hat{\theta}_n}(k)$ of $\hat{\theta}_n$ can be obtained by using the Laplace transform method. Since θ_{n-w} ($w = 1, \dots, 8$) is exponentially distributed with an expected value of $\frac{1}{p}$, the Laplace transform of θ_{n-w} is given by $L_{\theta_{n-w}}(s) = \frac{p}{p+s}$. Eq. (2) shows that $\hat{\theta}_n$ is the sum of multiple independent random variables, so the Laplace transform of $\hat{\theta}_n$ is given by

$$L_{\hat{\theta}_n}(s) = \left(\frac{p}{p+\frac{s}{6}}\right)^4 \left(\frac{p}{p+\frac{s}{7.5}}\right) \left(\frac{p}{p+\frac{s}{10}}\right) \left(\frac{p}{p+\frac{s}{15}}\right) \left(\frac{p}{p+\frac{s}{30}}\right) \quad (15)$$

By inverting $L_{\hat{\theta}_n}(s)$, we can get $f_{\hat{\theta}_n}(k)$, which, however, is very complicated. In order to simplify the analysis, we prefer to derive a simple and fast approximation of $f_{\hat{\theta}_n}(k)$ at the expense of less accuracy. Inspired by the fact that the sum of multiple independent and identical exponential random variables is an Erlang random variable, which has a relatively simple probability density function, we attempt to approximate $f_{\hat{\theta}_n}(k)$ with the probability density function of an Erlang random variable. For an Erlang random variable E with H stages and each stages with an expected value of $\frac{1}{\mu}$, the cumulative distribution function and probability density function of E can be obtained [28] by

$$F_E(H, \mu, k) = P(E \leq k) = 1 - \sum_{h=0}^{H-1} \frac{(\mu k)^h}{h!} e^{-\mu k} \quad k \geq 0 \quad (16)$$

$$f_E(H, \mu, k) = \frac{dF_E(H, \mu, k)}{dk} = \frac{\mu^H}{(H-1)!} k^{H-1} e^{-\mu k} \quad (17)$$

We define $\hat{\theta}_n$ as a new weighted average loss interval. Specifically,

$$\hat{\theta}_n = \frac{\theta_{n-1}}{7} + \frac{\theta_{n-2}}{7} + \frac{\theta_{n-3}}{7} + \frac{\theta_{n-4}}{7} + \frac{\theta_{n-5}}{7} + \frac{\theta_{n-6}}{7} + \frac{\theta_{n-7}}{7} \quad (18)$$

Lemma 1: The probability density function $f_{\hat{\theta}_n}(k)$ is given by

$$f_{\hat{\theta}_n}(k) = \frac{(7p)^7}{6!} k^6 e^{-7pk} \quad (19)$$

Proof: $\hat{\theta}_n$ is an Erlang random variable with 7 stages, each of them with an expected value of $\frac{1}{7p}$. Therefore, $f_{\hat{\theta}_n}(k) = f_E(7, 7p, k)$. ■

We hope that $f_{\hat{\theta}_n}(k)$ is a good approximation to $f_{\hat{\theta}_n}(k)$. Moment matching method is a popular approach to find an approximate distribution to another unknown or complicated distribution. Here, we use this method to determine whether $\hat{\theta}_n$ and $\hat{\theta}_n$ have similar distributions.

Lemma 2: Random variables $\hat{\theta}_n$ and $\hat{\theta}_n$ have the same first moment (i.e. the expected value), and the difference ratios of their second and third moments are less than 0.2% and 0.6%, respectively.

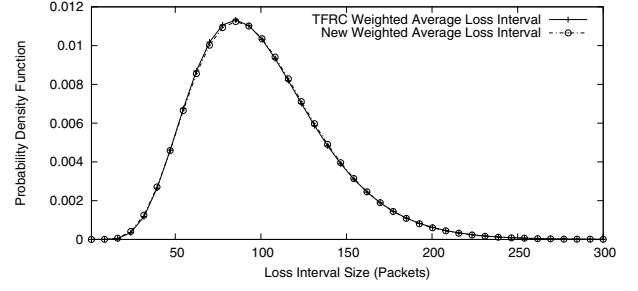


Fig. 4. The accurate probability density function of TFRC weighted average loss interval $\hat{\theta}_n$ obtained by inverting Eq. (15), and that of our new weighted average loss interval $\hat{\theta}_n$ given by Eq. (19). Loss event rate p is set to 0.01

Proof: The moments of a random variable can be obtained by differentiating its Laplace transform. The k^{th} moment of $\hat{\theta}$ is given by $E[\hat{\theta}^k] = (-1)^k \frac{d^k L_{\hat{\theta}}(s)}{ds^k} \Big|_{s=0}$, where $L_{\hat{\theta}}(s)$ is described by Eq. (15). Therefore, the first three moments of $\hat{\theta}_n$ are equal to $\frac{1}{p}$, $\frac{1.1444}{p^2}$, and $\frac{1.4778}{p^3}$, respectively.

The Laplace transform of $\hat{\theta}_n$ is $L_{\hat{\theta}_n}(s) = \left(\frac{p}{p+s/7}\right)^7$. Following the same procedure, we get the first three moments of $\hat{\theta}_n$, which are equal to $\frac{1}{p}$, $\frac{1.1429}{p^2}$, and $\frac{1.4694}{p^3}$, respectively.

We can see that $\hat{\theta}_n$ and $\hat{\theta}_n$ have the same first moment $\frac{1}{p}$. The difference ratios of their second and third moments are given by

$$\frac{|E[\hat{\theta}_n^2] - E[\hat{\theta}_n^2]|}{E[\hat{\theta}_n^2]} \approx 0.14\% \quad \frac{|E[\hat{\theta}_n^3] - E[\hat{\theta}_n^3]|}{E[\hat{\theta}_n^3]} \approx 0.57\%$$

respectively. ■

Figure 4 shows the accurate $f_{\hat{\theta}_n}(k)$ obtained by inverting Eq. (15), and $f_{\hat{\theta}_n}(k)$. We can see that both the figure and lemma show that $f_{\hat{\theta}_n}(k)$ is a good approximation to $f_{\hat{\theta}_n}(k)$. Finally, we have

$$f_{\hat{\theta}_n}(k) \approx f_{\hat{\theta}_n}(k) = \frac{(7p)^7}{6!} k^6 e^{-7pk} \quad k \geq 0 \quad (20)$$

$$F_{\hat{\theta}_n}(k) \approx F_{\hat{\theta}_n}(k) = 1 - \sum_{h=0}^6 \frac{(7pk)^h}{h!} e^{-7pk} \quad (21)$$

We wish to add that, we have tested various definitions for $\hat{\theta}_n$, including Erlang random variables with various stage numbers. Among them, Eq. (18) gives the best result. Note that even though $f_{\hat{\theta}_n}(k) \approx f_{\hat{\theta}_n}(k)$, the value of $\hat{\theta}_n$ could be quite different from that of $\hat{\theta}_n$ given the same set of $\theta_{n-1}, \dots, \theta_{n-8}$. That is, variable $\hat{\theta}_n$ itself is not a good approximation to variable $\hat{\theta}_n$.

D. TFRC Sending Rate Associated with an MRMDP State

In this subsection, we determine the fixed sending rate ϕ_i associated with MRMDP state i for TFRC traffic. Recall that if $\hat{\theta}_n \in \hat{\Theta}_i$, then $S_n = i$. Therefore, we set ϕ_i to the conditional time-average TFRC sending rate given $\hat{\theta}_n \in \hat{\Theta}_i$.

Lemma 3: The conditional time-average sending rate of TFRC given $S_n = i$ is given by

$$\phi_i = \frac{1}{E\left[\frac{1}{R\left(\frac{1}{\hat{\theta}_n}\right)} \mid S_n = i\right]} \quad (22)$$

This lemma can be derived by using Eq. (3) proved by Vojnović and Le Boudec [29]. However, due to the complexity introduced by the TFRC sending rate function $R\left(\frac{1}{\hat{\theta}_n}\right)$, it is hard to accurately calculate ϕ_i . Therefore, we resort to approximate Eq. (5) discovered by Rhee and Xu [21], and estimate ϕ_i as follows.

$$\phi_i \approx R\left(E\left[\frac{1}{\hat{\theta}_n} \mid S_n = i\right]\right) \quad (23)$$

Rhee and Xu [21], however, do not show how to obtain $E\left[\frac{1}{\hat{\theta}_n}\right]$. Below, we present two closed-form solutions to calculate $E\left[\frac{1}{\hat{\theta}_n} \mid S_n = i\right]$ and $E\left[\frac{1}{\hat{\theta}_n}\right]$, respectively.

Lemma 4: The conditional expected value of $\frac{1}{\hat{\theta}_n}$ given $S_n = i$ can be obtained by

$$E\left[\frac{1}{\hat{\theta}_n} \mid S_n = i\right] = \frac{7p}{6} \cdot \frac{F_E(6, 7p, b_{i+1}) - F_E(6, 7p, b_i)}{F_{\hat{\theta}_n}(b_{i+1}) - F_{\hat{\theta}_n}(b_i)} \quad (24)$$

Proof:

$$\begin{aligned} E\left[\frac{1}{\hat{\theta}_n} \mid S_n = i\right] &= E\left[\frac{1}{\hat{\theta}_n} \mid b_i < \hat{\theta}_n \leq b_{i+1}\right] \\ &= \frac{\int_{b_i}^{b_{i+1}} f_{\hat{\theta}_n}(k) \frac{1}{k} dk}{F_{\hat{\theta}_n}(b_{i+1}) - F_{\hat{\theta}_n}(b_i)} \\ &= \frac{\frac{7p}{6} \int_{b_i}^{b_{i+1}} \frac{(7p)^6}{5!} k^5 e^{-7pk} dk}{F_{\hat{\theta}_n}(b_{i+1}) - F_{\hat{\theta}_n}(b_i)} \end{aligned}$$

Notice that $\frac{(7p)^6}{5!} k^5 e^{-7pk}$ is the probability density function of an Erlang random variable with 6 stages, each stage with an expected value of $\frac{1}{7p}$. ■

Using Lemma 4, we can easily prove the following lemma by replacing b_i and b_{i+1} with 0 and ∞ , respectively.

Lemma 5: The expected value of $\frac{1}{\hat{\theta}_n}$ can be obtained by

$$E\left[\frac{1}{\hat{\theta}_n}\right] = \frac{7p}{6} \quad (25)$$

Substituting the above lemma into Eq. (5), we obtain a closed-form approximate solution for the average TFRC sending rate.

$$\phi_{tfrc} \approx R\left(E\left[\frac{1}{\hat{\theta}_n}\right]\right) = R\left(\frac{7p}{6}\right) \quad (26)$$

E. State Transition Probability of an MRMDP for TFRC

In this subsection, we derive the state transition probability $q_{ij}(t)$, which describes the transition probability from MRMDP state i to j after the MRMDP stays at state i for

more than t seconds. Starting with the definition of $q_{ij}(t)$ (i.e. Eq. (8)), we have

$$\begin{aligned} q_{ij}(t) &= P(t < \tau_n, S_{n+1} = j \mid S_n = i) \\ &= P(\phi_i t < \theta_n, b_j < \hat{\theta}_{n+1} \leq b_{j+1} \mid S_n = i) \\ &= P(\phi_i t < \theta_n, b_j < \hat{\theta}_{n+1} \mid S_n = i) \\ &\quad - P(\phi_i t < \theta_n, b_{j+1} < \hat{\theta}_{n+1} \mid S_n = i) \\ &= \frac{P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_j < \hat{\theta}_{n+1})}{P(b_i < \hat{\theta}_n \leq b_{i+1})} \\ &\quad - \frac{P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_{j+1} < \hat{\theta}_{n+1})}{P(b_i < \hat{\theta}_n \leq b_{i+1})} \end{aligned} \quad (27)$$

The common denominator $P(b_i < \hat{\theta}_n \leq b_{i+1})$ is equal to $F_{\hat{\theta}_n}(b_{i+1}) - F_{\hat{\theta}_n}(b_i)$. However, it is very difficult to accurately calculate the numerators $P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_j < \hat{\theta}_{n+1})$ and $P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_{j+1} < \hat{\theta}_{n+1})$. Below, we present an approximation approach to calculate the first numerator, from which the second one can be obtained by substituting b_j with b_{j+1} .

Probability $P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_j < \hat{\theta}_{n+1})$ depends on the distributions of three random variables: θ_n , $\hat{\theta}_n$, and $\hat{\theta}_{n+1}$. In order to simplify the calculation of this probability, we attempt to find a random variable $\check{\theta}_{n+1}$, which is a function of θ_n and $\hat{\theta}_n$, and is approximately equal to $\hat{\theta}_{n+1}$.

We define $\check{\theta}_n$ as another new weighted average loss interval. Specifically,

$$\check{\theta}_n = \frac{\theta_{n-1}}{6} + \frac{\hat{\theta}_{n-1}}{1.16} - \frac{1}{34.8p} \quad (28)$$

The second and third coefficients (i.e. 1.16 and 34.8) are selected so that $\check{\theta}_n$ and $\hat{\theta}_n$ have the same expected value. Note the difference between $\check{\theta}_n$ and $\hat{\theta}_n$. The former defined by Eq. (18) is designed so that $f_{\check{\theta}_n}(k)$ is easy to obtain, and $f_{\check{\theta}_n}(k) \approx f_{\hat{\theta}_n}(k)$. In contrast, the latter defined by Eq. (28) is designed so that $\check{\theta}_n$ is a function of θ_{n-1} and $\hat{\theta}_{n-1}$, and $\check{\theta}_n \approx \hat{\theta}_n$.

Lemma 6: The first two moments of random variable $\hat{\theta}_n - \check{\theta}_n$ are relatively small compared to those of $\hat{\theta}_n$. Specifically, we have

$$\frac{E[\hat{\theta}_n - \check{\theta}_n]}{E[\hat{\theta}_n]} = 0, \quad \frac{E[(\hat{\theta}_n - \check{\theta}_n)^2]}{E[\hat{\theta}_n^2]} < 0.4\%$$

respectively.

The above lemma can be proved by using the Laplace transform method. Since the proof of Lemma 6 is very similar to that of Lemma 2, we omit it here. Lemma 6 shows that the difference between $\hat{\theta}_n$ and $\check{\theta}_n$ is relatively very small. That is, $\hat{\theta}_n \approx \check{\theta}_n$. Following the same procedure, we can show that $\hat{\theta}_{n+1} \approx \check{\theta}_{n+1}$, and then we have

$$\begin{aligned} &P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_j < \hat{\theta}_{n+1}) \\ &\approx P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_j < \check{\theta}_{n+1}) \end{aligned} \quad (29)$$

We conclude this subsection with the following lemma.

Lemma 7: $P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_j < \check{\theta}_{n+1}) =$

$$\begin{cases} e^{-p\phi_i t} (F_{\hat{\theta}_n}(b_{i+1}) - F_{\hat{\theta}_n}(b_i)) & B \leq b_i \\ C (F_E(7, 1.83p, B) - F_E(7, 1.83p, b_i)) \\ + e^{-p\phi_i t} (F_{\hat{\theta}_n}(b_{i+1}) - F_{\hat{\theta}_n}(B)) & b_i < B \leq b_{i+1} \\ C (F_E(7, 1.83p, b_{i+1}) - F_E(7, 1.83p, b_i)) & b_{i+1} < B \end{cases} \quad (30)$$

where $B = 1.16b_j + 0.033p - 0.193\phi_i t$
and $C = 3.83^7 e^{-6pb_j - 0.172}$

Proof:

$$\begin{aligned} & P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_j < \check{\theta}_{n+1}) \\ &= P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, b_j < \frac{\theta_n}{6} + \frac{\hat{\theta}_n}{1.16} - \frac{1}{34.8p}) \\ &= P(\phi_i t < \theta_n, b_i < \hat{\theta}_n \leq b_{i+1}, 6(b_j + \frac{1}{34.8p} - \frac{\hat{\theta}_n}{1.16}) < \theta_n) \\ &= \int_{b_i}^{b_{i+1}} f_{\hat{\theta}_n}(k) P(\phi_i t < \theta_n, 6(b_j + \frac{1}{34.8p} - \frac{k}{1.16}) < \theta_n) dk \end{aligned}$$

Let $\psi_1 = \phi_i t$, and $\psi_2 = 6(b_j + \frac{1}{34.8p} - \frac{k}{1.16})$. To calculate the above probability, we consider which of ψ_1 and ψ_2 is larger. Let B denote the value of k , such that $\psi_1 = \psi_2$. It follows $B = 1.16b_j + 0.033p - 0.193\phi_i t$. After some straightforward manipulation, we have

- if $B < b_i$, then $\psi_1 > \psi_2, \forall k \in (b_i, b_{i+1}]$
- if $b_i \leq B < b_{i+1}$, then
 - $\psi_1 \leq \psi_2, \forall k \in (b_i, B]$, and
 - $\psi_1 > \psi_2, \forall k \in (B, b_{i+1}]$
- if $b_{i+1} \leq B$, then $\psi_1 \leq \psi_2, \forall k \in (b_i, b_{i+1}]$

First, we consider the case where $\psi_1 \leq \psi_2$, and suppose that $k \in (k_1, k_2]$.

$$\begin{aligned} & \int_{k_1}^{k_2} f_{\hat{\theta}_n}(k) P(\psi_1 < \theta_n, \psi_2 < \theta_n) dk \\ &= \int_{k_1}^{k_2} f_{\hat{\theta}_n}(k) P(\psi_2 < \theta_n) dk \\ &= \int_{k_1}^{k_2} \frac{(7p)^7}{6!} k^6 e^{-7pk} e^{-p\psi_2} dk \\ &= \left(\frac{7}{1.83}\right)^7 e^{-6pb_j - 0.172} \int_{k_1}^{k_2} \frac{(1.83p)^7}{6!} k^6 e^{-1.83pk} dk \\ &= 3.83^7 e^{-6pb_j - 0.172} (F_E(7, 1.83p, k_2) - F_E(7, 1.83p, k_1)) \end{aligned}$$

Second, we consider the case where $\psi_1 > \psi_2$, and suppose that $k \in (k_1, k_2]$.

$$\begin{aligned} & \int_{k_1}^{k_2} f_{\hat{\theta}_n}(k) P(\psi_1 < \theta_n, \psi_2 < \theta_n) dk \\ &= \int_{k_1}^{k_2} f_{\hat{\theta}_n}(k) P(\psi_1 < \theta_n) dk \\ &= e^{-p\phi_i t} (F_{\hat{\theta}_n}(k_2) - F_{\hat{\theta}_n}(k_1)) \end{aligned}$$

Finally we can get $q_{ij}(t)$ by using Eqs. (27), (29) and (30).

IV. AN m -MRMDP/D/1 QUEUEING NODE FOR TFRC CLIENT BUFFER

In this section, we present a queueing model for a TFRC client buffer with traffic described by an MRMDP proposed in Section III. The rebuffering probability of the queueing model, which is used as a metric of user-perceived media quality, is calculated by using an iterative method.

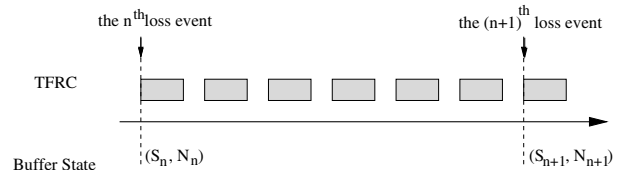


Fig. 5. Imbedded Markov process for a TFRC client buffer

A. Queueing Model for a TFRC Client Buffer

We model a TFRC client buffer as an m -MRMDP/D/1 queueing node, with an m -MRMDP traffic arrival model, a deterministic service rate of ϕ_{media} , and a buffer with infinite capacity. In order to simplify the analysis, we consider a TFRC client buffer as a fluid buffer, where fluid arrives at an m -MRMDP governed rate, and drains off at a fixed rate of ϕ_{media} . Consequently, the number of packets in a TFRC client buffer could be any non-negative real number.

Let $S(t) \in [0, m-1]$ denote the state of the m -MRMDP at time t , and $N(t) \in [0, \infty)$ denote the number of packets in the buffer. Unfortunately, the continuous-time continuous-space stochastic process $(S(t), N(t))$ is not a Markov process, because the next state of an MRMDP depends not only on the current state, but also on the elapsed time at the current state.

We now apply the method of the imbedded Markov process to our MRMDP/D/1 queueing node. To simplify the description, we assume that the RTT of a TFRC flow is zero. With this assumption, a data packet arrives at the TFRC client immediately, and a loss event is detected by the TFRC server immediately. Note that we make this assumption only to simplify the description. The analysis is still valid for TFRC flows with non-zero RTTs.

We define the imbedded Markov process (S_n, N_n) to be the buffer state immediately after the n^{th} loss event as illustrated in Figure 5. Once (S_n, N_n) is given, the next state (S_{n+1}, N_{n+1}) depends only on the current state (S_n, N_n) , not on any of the previous states. Given $S_n = i$, the probability of $S_{n+1} = j$ is described by $q_{ij}(t)$, which is defined by Eq. (8): $q_{ij}(t) = P(t < \tau_n, S_{n+1} = j | S_n = i)$. $\tau_n = \frac{\theta_n}{\phi_i}$ is the time interval between the n^{th} and $(n+1)^{th}$ loss events. Between these two loss events, the total number of packets arrived at the buffer is θ_n , and a maximum of $\tau_n \phi_{media}$ packets drain from the buffer or until the buffer becomes empty, therefore, N_{n+1} can be obtained as follows given that $S_n = i$ and $N_n = x$.

$$\begin{aligned} N_{n+1} &= \max\left(0, x + \theta_n - \tau_n \phi_{media}\right) \\ &= \max\left(0, x + \theta_n \left(1 - \frac{\phi_{media}}{\phi_i}\right)\right) \end{aligned} \quad (31)$$

Note that Eq. (31) does not consider the impact of lost packets, which is reasonable if a TFRC client applies error concealment techniques [18], [22] to recover or conceal lost packets. Otherwise, Eq. (31) has to be adjusted accordingly to exclude the number of lost packets.

B. Rebuffering Probability $\gamma(t|x)$ of m -MRMDP/D/I

A *rebuffering event* occurs when the media data in a TFRC client buffer has been exhausted, and the player is forced to stop playback until enough data has been received to resume. Intuitively, the greater the number and duration of rebuffering events, the worse the user-perceived media quality. We define a *rebuffering probability* $\gamma(t|x)$ as the probability that the total duration of all rebuffering events experienced by a user with an initial queue size (i.e. N_0) of x is more than t seconds. For instance, $\gamma(10|1500) = 1\%$ means the probability that a user with 1500 packets initially in the buffer experiences a total of more than 10 seconds of stalled playback is 1%. Note that a rebuffering probability depends also on media playback rate ϕ_{media} , loss event rate p , and packet RTT t_{rtt} . We omit them from the condition of a rebuffering probability, in order to have a concise notation.

We are interested in a special case of $\gamma(t|x)$ with $t = 0$. $\gamma(0|x)$ gives the probability that a user with $N_0 = x$ experiences at least one rebuffering event. Note that $\gamma(t|x)$ can be obtained by using $\gamma(0|x)$.

$$\gamma(t | x) = \gamma(0 | x + \phi_{media}t) \quad (32)$$

This equation can be intuitively explained as follows. If a buffer with initially x packets experiences a total of t rebuffering time, then the total time that the buffer is empty is t . Considering that a t seconds of empty buffer can be filled with $\phi_{media}t$ packets, we can see that a buffer with initially $x + \phi_{media}t$ packets will not experience any rebuffering event.

Rebuffering probability $\gamma(0|x)$ can be calculated by considering all possible initial MRMDP states (i.e. S_0) as follows

$$\begin{aligned} \gamma(0|x) &= P(\exists n, N_n \in [0, 1] | N_0 \in [x, x+1]) \quad (33) \\ &= \sum_{i=0}^{m-1} P(S_0 = i) \gamma(0 | i, x) \\ &= \sum_{i=0}^{m-1} (F_{\hat{\theta}_n}(b_{i+1}) - F_{\hat{\theta}_n}(b_i)) \gamma(0 | i, x) \end{aligned}$$

where $\gamma(0|i, x)$ is the probability that a user with $S_0 = i$ and $N_0 = x$ experiences at least one rebuffering event. Specifically,

$$\gamma(0|i, x) = P(\exists n, N_n \in [0, 1] | S_0 = i, N_0 \in [x, x+1]) \quad (34)$$

In the next few subsections, we present an iterative method to calculate $\gamma(0|i, x)$, and then $\gamma(t|x)$ and $\gamma(0|x)$ can be obtained by using Eqs. (32) and (33), respectively. Note that, the average TFRC sending rate ϕ_{tfrc} should be higher than or equal to ϕ_{media} . Otherwise, if $\phi_{tfrc} < \phi_{media}$, then (S_n, N_n) is an ergodic Markov process with a steady state, and consequently the rebuffering probability $\gamma(0|x)$ is equal to 100% for any finite x .

C. Rebuffering Probability $\gamma(0|i, x)$ of m -MRMDP/D/I

Let $q(i, x, j, y)$ denote the state transition probability of a TFRC client buffer from state (i, x) to state (j, y) . Specifically,

$$\begin{aligned} q(i, x, j, y) & \quad (35) \\ = P(S_{n+1} = j, N_{n+1} \in [y, y+1] | S_n = i, N_n \in [x, x+1]) \end{aligned}$$

For a TFRC stream with an infinite duration, we can get a recursive equation of $\gamma(0|i, x)$ by considering all possible next states of a TFRC client buffer from state (i, x) . The buffer may go to state (j, y) with a probability of $q(i, x, j, y)$, and we know that the rebuffering probability of a buffer with an initial state of (j, y) is given by $\gamma(0|j, y)$. It follows

$$\gamma(0 | i, x) = \begin{cases} 1 & x = 0 \\ \sum_{j=0}^{m-1} \sum_{y=0}^{\infty} q(i, x, j, y) \gamma(0 | j, y) & x > 0 \end{cases} \quad (36)$$

Eq. (36) can be rewritten in the following matrix form.

$$\Gamma = Q\Gamma \quad (37)$$

Vector Γ is the rebuffering probability vector defined by

$$\Gamma = (\begin{matrix} \gamma(0|0, 0), & \gamma(0|0, 1), & \dots, \\ \gamma(0|1, 0), & \gamma(0|1, 1), & \dots, \\ \dots & \dots & \dots \\ \gamma(0|m-1, 0), & \gamma(0|m-1, 1), & \dots \end{matrix})^T \quad (38)$$

Matrix Q is defined by

$$Q = \begin{pmatrix} A_{0,0} & A_{0,1} & \dots & A_{0,m-1} \\ A_{1,0} & A_{1,1} & \dots & A_{1,m-1} \\ \dots & \dots & \dots & \dots \\ A_{m-1,0} & A_{m-1,1} & \dots & A_{m-1,m-1} \end{pmatrix} \quad (39)$$

where matrix $A_{i,j}$ is given by

$$A_{i,j} = \begin{pmatrix} 1_{i=j} & 0 & \dots \\ q(i, 1, j, 0) & q(i, 1, j, 1) & q(i, 1, j, 2) & \dots \\ q(i, 2, j, 0) & q(i, 2, j, 1) & q(i, 2, j, 2) & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \quad (40)$$

The left-top element of $A_{i,j}$ is equal to 1 if i is equal to j ; otherwise, it is zero.

In the next two subsections, we present an approximation method to calculate matrix Q , and an iterative method to solve Eq. (37) for Γ , respectively. We want to mention that once Q is given, we can also calculate the following two items. *First*, the distribution of the first passage time of the system to an empty buffer, which is the time for the first rebuffering event to occur. *Second*, the distribution of the duration of a rebuffering event. Developing efficient algorithms to calculate these distributions is of further interest.

D. State Transition Probability $q(i, x, j, y)$ of m -MRMDP/D/I

In this subsection, we calculate the state transition probability $q(i, x, j, y)$ of an m -MRMDP/D/I queueing node. Define V_n by

$$V_n = \theta_n \left(1 - \frac{\phi_{media}}{\phi_i}\right) = \tau_n(\phi_i - \phi_{media}) \quad (41)$$

Eq. (31) can be rewritten as $N_{n+1} = \max(0, N_n + V_n)$. Solving for V_n , we get

$$\begin{cases} V_n = N_{n+1} - N_n & \text{if } N_{n+1} > 0 \\ |V_n| \geq N_n & \text{if } N_{n+1} = 0 \end{cases} \quad (42)$$

We consider $q(i, x, j, y)$ for two cases: $i \geq \frac{m}{2}$ and $i < \frac{m}{2}$.

1) If $i \geq \frac{m}{2}$, then $\phi_i > \phi_{media}$, and $V_n > 0$. Therefore, N_{n+1} must be greater than N_n . If $y < x$, we have $q(i, x, j, y)=0$; otherwise, it can be obtained as follows.

$$\begin{aligned} & q(i, x, j, y) \quad i \in \left[\frac{m}{2}, m-1\right), y \in [x, \infty) \\ & = P(S_{n+1} = j, y \leq N_{n+1} < y+1 \mid S_n = i, x \leq N_n < x+1) \\ & \approx P(S_{n+1} = j, y-x < N_{n+1} - N_n \leq y-x+1 \mid S_n = i) \\ & = P(S_{n+1} = j, y-x < V_n \leq y-x+1 \mid S_n = i) \\ & = P(S_{n+1} = j, y-x < \tau_n(\phi_i - \phi_{media}) \leq y-x+1 \mid S_n = i) \\ & = q_{ij} \left(\frac{y-x}{\phi_i - \phi_{media}}\right) - q_{ij} \left(\frac{y-x+1}{\phi_i - \phi_{media}}\right) \end{aligned} \quad (43)$$

2) If $i < \frac{m}{2}$, then $\phi_i \leq \phi_{media}$, and $V_n \leq 0$. Therefore, N_{n+1} must be smaller than or equal to N_n . If $y > x$, then $q(i, x, j, y)=0$; otherwise, it can be obtained as follows. If $0 < y \leq x$, we have

$$\begin{aligned} & q(i, x, j, y) \quad i \in \left[0, \frac{m}{2}\right), y \in (0, x] \\ & \approx P(S_{n+1} = j, x-y < |V_n| \leq x-y+1 \mid S_n = i) \\ & = q_{ij} \left(\frac{x-y}{\phi_{media} - \phi_i}\right) - q_{ij} \left(\frac{x-y+1}{\phi_{media} - \phi_i}\right) \end{aligned} \quad (44)$$

If $y = 0$, we have

$$\begin{aligned} & q(i, x, j, y) \quad i \in \left[0, \frac{m}{2}\right), y = 0 \\ & \approx P(S_{n+1} = j, x < |V_n| \mid S_n = i) \\ & = q_{ij} \left(\frac{x}{\phi_{media} - \phi_i}\right) \end{aligned} \quad (45)$$

We do not show all intermediate steps for the second case, as they are very similar to those for the first case.

E. Iterative Method to Calculate the Rebuffering Probability

In this subsection, we describe an iterative method to solve Eq. (37) (i.e. $\Gamma = Q\Gamma$) for Γ . Since $\gamma(0|i, 0) = 1 \neq 0$ for $\forall i$, it has only one unique solution. Specifically, we

- Step 1: initialize vector Γ , set $\gamma(0|i, 0) = 1 \forall i$, and set all other elements to 0
- Step 2: calculate $\hat{\Gamma} = Q\Gamma$
- Step 3: check the following convergence condition. If it is satisfied, then stop; otherwise set $\Gamma = \hat{\Gamma}$, and go back to Step 2.

$$\frac{\sum_i \sum_x (\gamma(0|i, x) - \hat{\gamma}(0|i, x))^2}{\sum_i \sum_x \gamma(0|i, x)^2} < \epsilon$$

where ϵ determines the accuracy and running time of the algorithm. The smaller the value of ϵ , the better the accuracy, but the longer the running time. In our experiments, we set ϵ to 10^{-6} .

There are two implementation issues of this iterative method. *First*, since we assume the buffer capacity is infinite, the orders of vector Γ and matrix Q should be infinite, too. In the implementation, we therefore use a truncated Γ and Q . The order of the truncated Γ is set to mM , where m is the number of MRMDP states, and M could be considered as the maximum queue size. The order of the truncated Q is then $mM \times mM$. In our experiments, we set M to 6000. With a packet size of 1500 bytes, $M = 6000$ means we can calculate the rebuffering probability with up to $6000 * 1500 \approx 9$ Mbytes data initially in the client buffer. *Second*, we must reduce the space required by the truncated Q which is in the order of $O(m^2M^2)$. Instead of maintaining matrix Q , we keep only $q_{ij}(t)$, and then the required space can be reduced to the order of $O(m^2M)$.

V. NUMERICAL STUDIES AND DISCUSSIONS

In this section, we validate our proposed model for a TFRC client buffer through comparison with simulation results, and we also study the impact of various model parameters on user-perceived media quality.

A. Simulation Methodology

This subsection describes how we simulate TFRC and TFRC client buffer to measure a rebuffering probability, which is used as a rough metric of user-perceived media quality. Limited by simulation time, we consider only a rebuffering probability as low as 10^{-6} . A rebuffering probability of 10^{-6} means statistically one of 10^6 users experiences stalled playback due to rebuffering events. In this case, we must simulate at least 10^6 users in order to catch a rebuffering event. In our experiments, we therefore run the simulation for 10^6 times (the larger the better) for every given set of media, network, and buffer parameters. Due to the requirement for such a large number of simulation runs, we decide to develop a loss-event-level TFRC simulator, which runs significantly faster than packet-level TFRC simulators such as NS-2.

We do not simulate the slow start phase of TFRC, and assume that TFRC is already in the congestion avoidance phase with 8 randomly-generated most recent loss intervals. Loss events are generated randomly based on an exponential distribution. At each loss event, TFRC recalculates its sending rate by using Eq. (1). Initially, there are x packets in a TFRC client buffer, where x is called an *initial queue size*. Packets arrive at the buffer at the rate calculated by TFRC, and leave from the buffer at a fixed media playback rate of ϕ_{media} .

Since our model considers only a media stream with an infinite duration, we must run a simulation for a sufficiently long time. Figure 6 shows the rebuffering probability $\gamma(0|1500)$ as the media duration varies from 15 to 1000 seconds. The media playback rate ϕ_{media} is set to 100 packets per second, and the initial queue size is set to 1500 packets. That is, the first

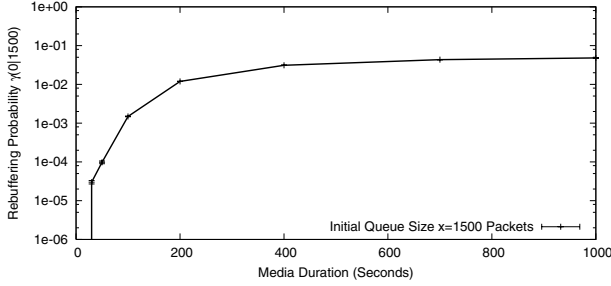


Fig. 6. Impact of media duration on the rebuffering probability $\gamma(0|1500)$. Media playback rate ϕ_{media} is set to 100 packets per second.

1500/100=15 seconds of media data is already in the buffer. Therefore the rebuffering probability of the media with a duration of 15 seconds is zero. We can see that the rebuffering probability increases as the media duration increases. Intuitively, the longer the media duration, the larger the possibility of rebuffering events. The rebuffering probability increases very quickly when the media duration is less than 200 seconds. After 200 seconds, it increases very slowly, and beyond 400 seconds, it becomes almost a constant. In our experiments, we therefore run a simulation for 2000 seconds, or until a rebuffering event occurs (i.e. the number of packets in a client buffer is less than 1). Figure 6 shows that a media stream with a sufficiently long duration (in this case, longer than 400 seconds) has almost the same rebuffering probability as a stream with an infinite duration, and therefore can be analyzed as if its duration is infinite.

To validate our models, we have run simulations with various parameters. Limited by space, we show only the simulation results when loss event rate p is set to 0.01, the RTT of TFRC streams is set to 0.1 seconds, and the timeout period is set to 0.4 seconds. The 95% confidence intervals are shown for all simulation results obtained by measuring a total of 10 rebuffering probability samples. However, most of them are very narrow and are barely visible in the figures.

B. Impact of Initial Queue Sizes (or Initial Buffering Delays)

In order to provide a user with better playback experience, a media player usually buffers the first x packets of a media stream prior to playback. Intuitively, the larger the value of x , the better the playback experience, however, the longer the initial buffering delay. In this subsection, we study the impact of initial queue size x on a rebuffering probability $\gamma(0|x)$. Since for any t , $\gamma(t|x)$ can be easily obtained by using Eq. (32), we show only the results of $\gamma(0|x)$.

Figure 7 shows the rebuffering probability $\gamma(0|x)$ obtained with the following two methods.

- simulating a TFRC client buffer by using our simulator
- analyzing a 10-MRMDP/D/1 by using the iterative method, i.e., Eq. (33)

In the figure, their results are referred to as the results of simulation and iterative method, respectively. The top two plots show $\gamma(0|x)$ of a TFRC client buffer with $\phi_{media}=100$ packets per second, as x increases from 1 to 4000 packets. The

bottom two plots show the results with $\phi_{media}=80$. We can see that the results of the iterative method are very close to those of the simulation. Note that, we do not show simulation results with $\phi_{media} = 80$ for $x > 2000$, because our simulation cannot measure any rebuffering probability lower than 10^{-6} as explained in the previous subsection.

With $\phi_{media} = 100$, it looks like the simulation results between $[0, 4000]$ lie on a straight line. The same holds for the simulation results between $[0, 500]$ with $\phi_{media} = 80$. Since the Y-axis uses a log scale, a straight line in the figure corresponds to an exponential distribution. From Figure 7 and other simulation results not shown here, we find that the rebuffering probability $\gamma(0|x)$ of a TFRC client buffer decreases exponentially, as the initial queue size x increases linearly up to a point depending on the media playback rate ϕ_{media} . The smaller the media playback rate, the faster the decreasing speed.

Furthermore, because $\gamma(t|x) = \gamma(0|x + \phi_{media}t)$, we can see that for a given x , $\gamma(t|x)$ decreases exponentially, as t increases linearly up to a point depending on ϕ_{media} . That is, the probability for a user to experience a total of t seconds of stalled playback decreases exponentially, as t increases linearly up to a certain point.

C. Impact of Media Playback Rates

A recent survey [14] reports that a significant fraction of media stored on the Internet is encoded with multiple-rate encoding techniques. When a user connects to a media server, the stream most appropriate for the current network condition is sent to the user. In this subsection, we study the impact of media playback rate ϕ_{media} on a rebuffering probability.

Figure 8 shows the rebuffering probability of a TFRC client buffer with $x=1500$ packets, as ϕ_{media} increases from 70 to 110 packets per second. We observe that the results of the iterative method are very accurate. Figure 8 also plots three vertical lines from the right to left:

- $R(E[\frac{1}{\theta_n}])$: the estimated average sending rate of a TFRC flow proposed in [21]. For this case, since $p = 0.01$, we get $R(E[\frac{1}{\theta_n}]) = R(\frac{7p}{6}) = 102.6$ by using Eq. (26).
- ϕ_{tfrc} : the average sending rate of a TFRC flow. Since we do not have a method to accurately calculate ϕ_{tfrc} , we measure it by using our simulator. We get $\phi_{tfrc} = 103.9$.
- ϕ_{tcp} : the average sending rate of a TCP flow under the same network conditions, which can be obtained by using Eq. (1). We get $\phi_{tcp} = R(p) = 112.3$.

We observe that $R(E[\frac{1}{\theta_n}]) \approx \phi_{tfrc} < \phi_{tcp}$, which is consistent with the results shown in [29] and [21] (i.e. Eqs. (4) and (5)). We can also see that the rebuffering probability increases as ϕ_{media} , and it approaches 100% as ϕ_{media} reaches ϕ_{tfrc} . In order to have a rebuffering probability less than 100%, a streaming media application must make sure that ϕ_{media} is lower than ϕ_{tfrc} . Therefore, $R(\frac{7p}{6})$ instead of $R(p)$ should be used as the maximum allowed playback rate of a media stream.

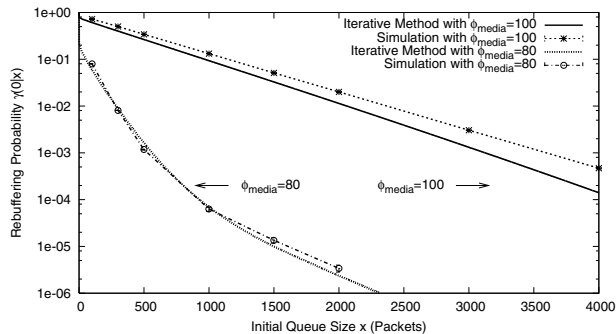


Fig. 7. Impact of initial queue size x on the rebuffering probability $\gamma(0|x)$ of a TFRC client buffer with media playback rate $\phi_{media}=100$ and 80 (packets/second), respectively.

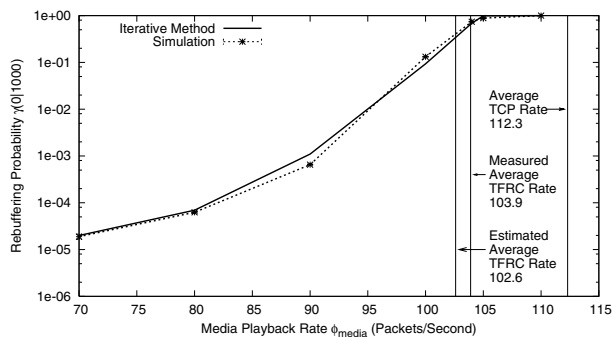


Fig. 8. Impact of media playback rate ϕ_{media} on the rebuffering probability $\gamma(0|1000)$ of a TFRC client buffer with an initial queue size of 1000 packets

VI. RELATED WORK

Even though the performance of TFRC has been extensively studied, little is known about its quantitative impact on user-perceived media quality. Most of the current literature focuses on the effectiveness of TFRC as a congestion control protocol [5], [4], [29], [21], or indirectly evaluate its impact on media quality by measuring the sending rate smoothness [5], [38], [34]. Kim et al. [11] and Wang et al. [33] have performed some simulation and experiments to study the impact of TFRC on media quality. Wu et al. [36] present an analytical model to study the impact of Group of Pictures (GOP) and Forward Error Correction (FEC) on the playable frame rate of a TFRC flow. Shen et al. [25] analytically studied the packet delay of a TFRC flow in wireless networks.

VII. CONCLUSIONS

In this paper, we consider the transmission of pre-recorded media via TFRC, and analytically study the impact of TFRC on user-perceived media quality, which is roughly measured by calculating the rebuffering probability. A TFRC client buffer is modeled as an MRMDP/D/1 queueing node. An iterative algorithm is presented to calculate the rebuffering probability of a queueing node. We also provide a closed-form solution to approximate the average TFRC sending rate, which can be used as the maximum allowed playback rate of a media stream. Our work can be used to help a streaming media application

to determine an appropriate initial buffering delay and media playback rate based on the current network condition.

In this paper, we evaluate our models only by comparison with simulation results. Validating our models by extensive Internet experiments is of future interest. We are also interested in analyzing the impact of TFRC on *live* media streaming, and in developing efficient algorithms to calculate the distribution of the time for the first rebuffering event to occur, and the distribution of the duration of a rebuffering event.

VIII. ACKNOWLEDGMENTS

The work reported in this paper is supported in part by Maude Hammond Fling Faculty Research Fellowship and Layman Fund Award. We would like to thank the anonymous reviewers for their constructive suggestions and comments.

REFERENCES

- [1] AccuStream IMedia Research. Streaming media 2004-2007: Market development and user data analysis. <http://www.accustreamresearch.com/>, 2005.
- [2] M. Allman, W. Eddy, and S. Ostermann. Estimating loss rates with TCP. *ACM Performance Evaluation Review*, 31(3), December 2003.
- [3] A. Baiocchi, N. B. Melazzi, M. Listani, A. Roveri, and R. Winkler. Loss performance analysis of an ATM multiplexer loaded with high-speed on-off sources. *IEEE Journal on Selected Areas in Communications*, 9(3):388–393, April 1991.
- [4] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker. Dynamic behavior of slowly-responsive congestion control algorithms. In *Proceedings of ACM SIGCOMM*, San Diego, CA, August 2001.
- [5] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM*, pages 43–56, Stockholm, Sweden, August 2000.
- [6] S. Floyd, M. Handley, J. Padhye, and J. Widmer. TCP friendly rate control (TFRC): Protocol specification. RFC 3448, January 2003.
- [7] S. Floyd and E. Kohler. TCP friendly rate control (TFRC) for voice: VoIP variant and faster restart. *Internet draft draft-ietf-dccp-tfrc-voip-02.txt*, July 2005.
- [8] S. Floyd, E. Kohler, and J. Padhye. Profile for DCCP congestion control ID 3: TFRC congestion control. *Internet draft draft-ietf-dccp-ccid3-11.txt*, March 2005.
- [9] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: estimating latency between arbitrary internet end hosts. In *Proceedings of the SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.
- [10] ITU-T Recommendation P.910. *Subjective Video Quality Assessment Methods for Multimedia Applications*. ITU-T, Geneva, 1996.
- [11] T. Kim and M. H. Ammar. Optimal quality adaptation for MPEG-4 fine-grained scalable video. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, April 2003.
- [12] J. Kurose and K. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley, 2005.
- [13] N. Laoutaris and I. Stavrakakis. Instream synchronization for continuous media streams: A survey of playout schedulers. *IEEE Network Magazine*, 16(3), May 2002.
- [14] M. Li, M. Claypool, R. Kinicki, and J. Nichols. Characteristics of streaming media stored on the web. *ACM Transactions on Internet Technology (to appear)*, 2005.
- [15] D. Loguinov and H. Radha. Measurement study of low-bitrate Internet video streaming. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, November 2001.
- [16] X. Lu, R. Morando, and M. Zarki. Understanding video quality and its use in feedback control. In *Proceedings of 12th International Packet Video Workshop*, Pittsburgh, PA, April 2002.
- [17] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe. Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of the ACM SIGCOMM*, pages 303–314, 1998.
- [18] C. Perkins, O. Hodson, and V. Hardman. A survey of packet loss recovery techniques for streaming audio. *IEEE Network*, 12(5), September 1998.

- [19] H. Perros. *Computer Simulation Techniques - The Definitive Introduction*. <http://www.csc.ncsu.edu/faculty/perros/simulation.pdf>, 2003.
- [20] PlanetLab. <http://www.planet-lab.org/>.
- [21] I. Rhee and L. Xu. Limitations in equation-based congestion control. In *Proceedings of ACM SIGCOMM*, pages 49–60, Philadelphia, PA, August 2005.
- [22] P. Salama, N. B. Shroff, and E. J. Delp. Error concealment in encoded video. *IEEE Journal on Selected Areas in Communications*, 18(6):1129–1140, June 2000.
- [23] J. Salehi, Z. Zhang, J. Kurose, and D. Towsley. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. *ACM SIGMETRICS Performance Evaluation Review*, 24(1):222–231, May 1996.
- [24] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. RFC 3550, July 2003.
- [25] H. Shen, L. Cai, and X. Shen. Performance analysis of TFRC over wireless links with truncated link level ARQ. *IEEE Transaction on Wireless Communications (to appear)*, 2005.
- [26] J. Sommers, P. Barford, N. Duffield, and A. Ron. Improving accuracy in end-to-end packet loss measurement. In *Proceedings of ACM SIGCOMM*, Philadelphia, PA, August 2005.
- [27] S. Tao, J. Apostolopoulos, and R. Guérin. Real-time monitoring of video quality in IP networks. In *Proceedings of NOSSDAV*, Stevenson, WA, June 2005.
- [28] K. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley and Sons, New York, 2001.
- [29] M. Vojnović and J. Le Boudec. On the long run behavior of equation-based rate control. In *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, August 2002.
- [30] B. Wang, J. Kurose, P. Shenoy, and D. Towsley. Multimedia streaming via TCP: An analytic performance study. In *Proceedings of ACM Multimedia*, New York City, NY, October 2004.
- [31] Y. Wang, M. Claypool, and Z. Zuo. An empirical study of RealVideo performance across the internet. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, November 2001.
- [32] Z. Wang, S. Banerjee, and S. Jamin. Studying streaming video quality: From an application point of view. In *Proceedings of ACM Multimedia (poster)*, Berkeley, CA, November 2003.
- [33] Z. Wang, S. Banerjee, and S. Jamin. Media-aware rate control. *Technical Report, Department of EECS, University of Michigan*, November 2004.
- [34] Z. Wang, S. Banerjee, and S. Jamin. Media-friendliness of a slowly-responsive congestion control protocol. In *Proceedings of ACM NOSSDAV*, Kinsale, Ireland, June 2004.
- [35] J. Widmer. TFRC Implementation. <http://www.icir.org/tfrc/code/>.
- [36] H. Wu, M. Claypool, and R. Kinicki. A model for MPEG with forward error correction and TCP-friendly bandwidth. In *Proceedings of NOSSDAV*, Monterey, CA, June 2003.
- [37] T. Yang and D. H. K. Tsang. A novel approach to estimating the cell loss probability in an ATM multiplexer loaded with homogeneous on-off sources. *IEEE Transactions on Communication*, 43:117–126, January 1995.
- [38] Y. Yang, M. Kim, and S. Lam. Transient behaviors of TCP-friendly congestion control protocols. *Computer Networks*, 41(2):193–210, February 2003.
- [39] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker. On the constancy of Internet path properties. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, November 2001.