A Model for TCP-based Video Streaming

Bing Wang, Jim Kurose, Prashant Shenoy, Don Towsley

Deptarment of Computer Science
University of Massachusetts, Amherst, MA 01003

***Abstract*—**

TCP is widely used by commercial video streaming systems. When a packet has not arrived by its playback time, a typical practice in these commercial system is that the client simply stops and waits for this packet, and then resumes playback. This stop-and-wait playout strategy is easy to implement. However, stopping playout due to late packet arrivals renders the viewing experience unsatisfactory. A continuous playout strategy, i.e., continuing playout regardless of late packet arrivals, also leads to unsatisfactory viewing experience, since late packet arrivals cause glitches in the playback. The performance of both the stop-and-wait and the continuous playout strategies therefore depends on the frequency of late packet arrivals during the playback of the video. In this paper, we develop discrete-time Markov models to evaluate the performance of live and stored video streaming using TCP. We validate the models using *ns* simulations and experiments conducted over the Internet. Based on the models, we provide guidelines as to when using TCP for streaming leads to satisfactory performance.

I. Introduction

The rapid growth of network bandwidth, especially the installment of high-bandwidth services such as cable modem and digital subscriber loop (DSL) at resident houses, makes video streaming more promising than ever. A common wisdom with regard to video streaming is to use UDP rather than TCP as the transport protocol. The main reason for not using TCP is that the backoff and retransmission in TCP can lead to undesirable end-to-end delays that violate the timeliness requirement for streaming. We refer to a packet arriving later than its playback time as a *late packet*. UDP does not have the issue of late packets since it does not have a backoff or retransmission mechanism. However, two other issues, namely satisfying TCP friendliness and loss recovery, have to be solved when using UDP for streaming. TCP friendliness is important since TCP is largely responsible for the stability of the network and it is desirable for network traffic to react to congestion in a TCP-friendly manner in order to avoid network collapse [1]. Loss recovery is needed since UDP does not provide reliability and losses in a stream, especially of header information, can render a segment of video unviewable.

Although both TCP and UDP have disadvantages when used for video streaming, there have been a greater amount of research on using UDP for streaming. This is because UDP provides more flexibility to higher level protocols than TCP. When using UDP, rate control and loss recovery strategies can be exploited by higher layers to achieve TCP friendliness and recover loss. Reducing the number of late packets while using TCP, however, seems beyond the control of higher layer protocols, since they are caused by the congestion control and avoidance mechanisms embedded in TCP. Therefore, it appears that the problems arising when using TCP for streaming are more difficult to overcome than those when using UDP.

Despite the difficulties, using TCP for streaming has obvious advantages. First, TCP is by definition TCP friendly. Second, reliable transmission provided by TCP removes the need for loss recovery at higher levels. These advantages motivate several efforts on using TCP for streaming [2], [3], [4], [5]. A common characteristic of these efforts is that they combine *client-side buffering* and *rate adaptation* together to deal with the variability in the available TCP throughput. Client-side buffering prefetches data into the client buffer by introducing a startup delay in order to absorb short-term fluctuations in the TCP throughput. Rate adaptation adjusts the bitrate (or quality) of the video in order to deal with long-term fluctuations. In [2], [3], rate adaptation requires prioritization to be associated with various frames in the video in order to control the frame rate. In [4], [5], rate adaptation is based on the periodic feedback from the client on available bandwidth and only applies to layered video. All of the above schemes, however, have limitations: prioritization of video frames requires extra work at the application level and may not be suitable for live streaming; the schemes based on layered videos restrict the formats of the videos that can be streamed using TCP.

Although there are very few research efforts on using TCP for video streaming, TCP is widely used by commercial video streaming systems. For instance, RealPlayer uses TCP as the default transport protocol [6]. These com-

mercial systems, however, usually do not exploit sophisticated rate adaptation at the application level. Instead, they use a simple stop-and-wait playout strategy to deal with late packets: when late packets are encountered, the client stops for the late packets and then resumes playback. Not requiring rate adaptation at the application level simplifies the design and implementation of the commercial systems. However, stopping playout due to late packets renders the viewing experience unsatisfactory. A continuous playout strategy, i.e., continuing playout regardless of late packets, also leads to unsatisfactory viewing experience, since late packets cause glitches in the playback. The performance of both the stop-and-wait and the continuous playout strategies depends on the frequency of late packets during the playback of the video.

In this paper, we study the performance when using TCP *directly* for streaming (i.e., without rate adaptation at the application level). We aim to answer two related questions: (i) What is the performance of using TCP directly for streaming? (ii) Under what conditions does the use of TCP provide satisfactory viewing experience? Answering the above questions is important in determining whether using TCP directly for streaming is sufficiently good and when some more sophisticated mechanism (either using UDP or TCP with rate adaptation) is required to achieve good performance. We develop a discrete-time Markov model for streaming using TCP to answer the above questions. The model is based on the continuous playout strategy and provides insights into the stop-and-wait playout strategy (see Section V-E). Our main contributions are:

- We develop models for both *live* and *stored* video streaming using TCP to study the effect of various parameters (i.e., loss rate, round trip time and timeout value in TCP as well as the video playback rate) on the likelihood of late packets for a startup delay on the order of seconds. The models are validated using *ns* [7] simulation and Internet experiments.
- Using the model, we explore the parameter space to provide guidelines as to when using TCP directly for streaming leads to a satisfactory performance.

The rest of the paper is organized as follows. Section II presents the models for live and stored video streaming using TCP. Validation of the models using *ns* simulations and Internet experiments is described in Sections III and IV respectively. A performance study based on the models is presented in Section V. Finally, Section VI concludes the paper and presents future work.

II. Models for streaming using TCP

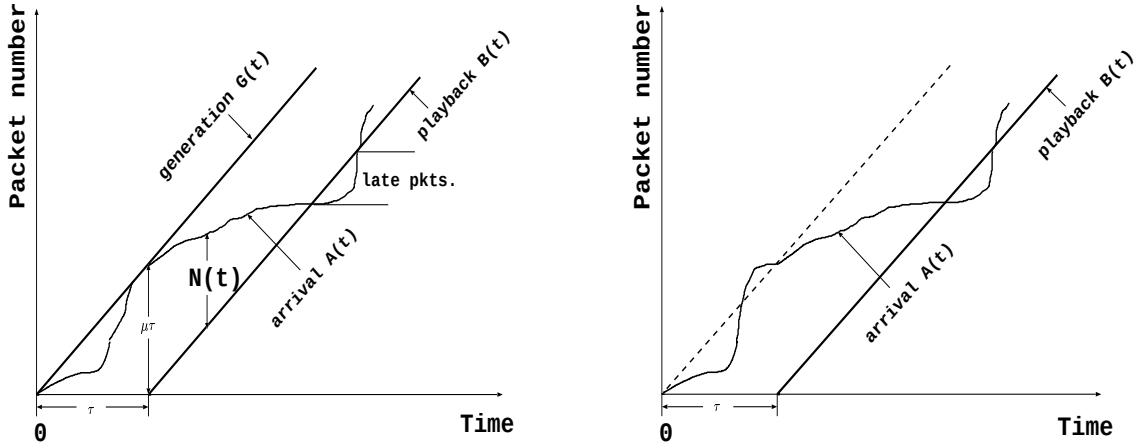
In this section, we describe the problem setting and then present discrete-time Markov models for live and stored video streaming using TCP. The key notation introduced in this section is summarized in Table I for easy reference.

A. Problem setting

Consider a client requesting a video from the server. Corresponding to the request, the server streams the video to the client using TCP. Throughout the paper, we assume that the average TCP throughput is no less than the bandwidth of the video. This guarantees that, on average, the throughput provided by TCP satisfies the requirement for streaming the video. However, fluctuations in the instantaneous TCP throughput can still lead to significant late packet arrivals. All the packets arriving earlier than their playback times are stored at the client's local buffer. We assume this local buffer is sufficiently large so that no packet loss is caused by buffer overflow at the client side. This assumption is reasonable since most machines are equipped with a large amount of storage nowadays.

For simplicity, we consider a CBR (constant bit rate) video. The playback rate of the video is μ packets per second. For simplicity, all packets are assumed to be of the same size. For analytical tractability, we assume continuous playback at the client. That is, a client does not stop and wait for a late packet but plays back at a constant rate of μ packets per second. A late packet therefore leads to a glitch during the playback and causes performance degradation. We are interested in the *fraction of late packets*, i.e., the probability that a packet is late. In Section V-E, we discuss how the insights obtained from our models can be applied to the stop-and-wait behavior in the commercial streaming systems.

We study two forms of streaming that correspond respectively to live and stored video streaming in practice. In live streaming, the server generates video content in real time and is only able to transmit the content that has already been generated. The transmission is therefore constrained by the generation rate of the video at the application level. Hence we refer to this form of streaming as *constrained streaming*. For a stored video, we assume the server transmits the video as fast as allowed by the available TCP bandwidth in order to fully utilize the available TCP bandwidth. We refer to this form of streaming as *unconstrained streaming* since the application does not impose any constraint on the transmission. We next illustrate the characteristics of constrained and unconstrained streaming. For ease of exposition, each packet is associ-



(a) Constrained streaming. (b) Unconstrained streaming.

Fig. 1. Video streaming using TCP: Constrained and unconstrained streaming.

ated with a packet sequence number and the first packet has sequence number of 1.

Constrained streaming is illustrated in Fig. 1(a). Without loss of generality, we assume the first packet is generated at time 0. Later packets are generated at a constant rate equal to the playback rate of the video. In the figure, $G(t)$ represents the number of packets generated at the server by time t. Then $G(t) = \mu t$. At the client side, let $A(t)$ denote the number of packets reaching the client by time t. Since the TCP transmission is constrained by the generation rate at the server, we have $A(t) \leq G(t)$. Denote $B(t)$ to be the number of packets played by the client by time t. The playback of the video commences at time τ. That is, the startup delay is τ seconds. Then $B(t) = \mu(t - \tau)$, $t \geq \tau$. Observe that $G(t) - B(t) = \mu\tau$. A packet arriving earlier than its playback time is referred to as an *early packet*. At time t, let the number of early packets be $N(t)$. Then $N(t) = A(t) - B(t)$. A negative value of $N(t)$ indicates that the packet arrival is behind the playback by $-N(t)$ packets. Since $A(t) \leq G(t)$ and $G(t) - B(t) = \mu\tau$, we have $N(t) \leq G(t) - B(t) = \mu\tau$. That is, there are at most $\mu\tau$ early packets in constrained streaming at any time t, as shown in Fig. 1(a). This observation is to be used in the model for constrained streaming later in this section.

Unconstrained streaming is illustrated in Fig. 1(b). As shown in the figure, the packet transmission is only limited by the available TCP throughput and no constraint is imposed from the application level. Therefore, the number of early packets at time t, $N(t)$, can be larger than $\mu\tau$.

As described above, a negative value of $N(t)$ indicates that late packets occur at time t. We need to model $N(t)$ in order to obtain the fraction of late packets. Since $N(t) = A(t) - B(t)$ and $B(t)$ is a simple linear function of t, the main difficulty left is modeling $A(t)$. We use the the models in [8], [9] to describe $A(t)$, as discussed below.

Notation	Definition
μ	Playback rate of the video (packets per second)
τ	Startup delay (seconds)
X_i	State of the TCP source in the ith round
S_i	Number of packets transmitted successfully by TCP in the ith round
R	Round trip time (seconds)
L	Length of the video (measured in rounds)
f	Fraction of late packets
N_i	Number of early packets in the ith round
N_i^l	Number of late packets in the ith round
Y_i^c	State of the model for constrained streaming in the ith round
Y_i^u	State of the model for unconstrained streaming in the ith round
P_i	Probability of having at least one late packet in the ith round

TABLE I
KEY NOTATION.

B. Model for TCP throughput

TCP is a window-based protocol with several mechanisms used to regulate its sending rate in response to network congestion. Timeout and congestion avoidance are two mechanisms that have significant impact on the throughput. For completeness, we give a brief description of these two mechanisms. More detailed description can be found in [10]. For every packet sent by the source, TCP starts a retransmission timer and waits for an acknowledgment from the receiver. The retransmission timer expires (timeouts) when the ACK for the corresponding packet is lost and there are no triple duplicate ACKs. When timeout occurs, the packet is retransmitted and the window size is reduced to one. Furthermore, the retransmission timer value for this retransmitted packet is set to be twice

the previous timer value. This exponential backoff behavior continues until the retransmitted packet is successfully acknowledged. In congestion avoidance, the window size increases by one packet when all packets in the current window are acknowledged. In most versions of TCP, such as TCP Reno and TCP Sack, the window size is reduced by half when triple duplicate ACKs are received. If timeout occurs before receiving triple duplicate ACKs, the window size is reduced to one.

In [8], [9], the behavior of TCP is described by a discrete-time Markov model, where each time unit is the length of a "round". A round starts with the back-to-back transmission of W packets, where W is the current size of TCP congestion window. Once all packets in the congestion window are sent, no more packets are sent until ACKs for some or all of these W packets are received. The reception of the ACKs marks the end of the current round and the beginning of the next round. The length of a round is assumed to be a round trip time (RTT). Packet losses in different rounds are assumed to be independent and packet losses in the same round are correlated: if a packet is lost, all remaining packets until the end of the round are lost. Furthermore, the effect of lost ACKs is regarded as ignorable.

Let $\{X_i\}_{i=1}^{\infty}$ be a discrete-time Markov model for the TCP source, where X_i is the state of the model in the ith round. Following the notation in [8], [9], X_i is a tuple: $X_i = (W_i, C_i, L_i, E_i, R_i)$, where W_i is the window size in the ith round; C_i models the delayed ACK behavior of TCP ($C_i = 0$ and $C_i = 1$ indicate the first and the second of the two rounds respectively); L_i is the number of packets lost in the $(i-1)$th round; E_i denotes whether the connection is in a timeout state and the value of the backoff exponent in the ith round; R_i indicates if a packet being sent in the timeout phase is a retransmission ($R_i = 1$) or a new packet ($R_i = 0$). Denote the number of packets transmitted successfully by TCP in the ith round as S_i. Then S_i is determined by X_i and X_{i+1}. Let $E[S_i]$ be the expectation of S_i. We have $E[S_i] = E[S_i \mid X_i = (w, c, l, e, r), X_{i+1} = (w', c', l', e', r')]$. The detailed description of how to compute $E[S_i]$ can be found in [8], [9], [11]. The total number of packets transmitted successfully by TCP up to the kth round is $\sum_{i=1}^{k} S_i$.

C. Models for constrained and unconstrained streaming

We now develop discrete-time Markov models for constrained and unconstrained streaming. Each time unit corresponds to the length of a round, which is assumed to be a RTT of length as R time units. We consider a video whose length is L rounds. The playback rate of the video is μR packets per round.

Let f denote the fraction of late packets during the playback of the video. Our goal is to derive models for determining f as a function of various system parameters (including the loss rate, RTT, retransmission timer in the TCP flow and the video playback rate). Let N_i denote the number of early packets in the ith round, which is a discrete-time version of $N(t)$ introduced earlier (see Section II-A) and $N_i = N(iR)$. For simplicity of notation, we assume the number of packets played back in a round, μR, to be an integer. Let N_i^l be the number of late packets in the ith round. Then $N_i^l \in \{0, 1, \ldots, \mu R\}$. Let the expected number of late packets in the ith round be $E[N_i^l]$. Then

$$E[N_i^l] = \sum_{k=1}^{\mu R} k P(N_i^l = k)$$

where $P(N_i^l = k)$ is the probability of having k late packets in the ith round. The fraction of late packets is

$$f = \frac{\sum_{i=1}^{L} E[N_i^l]}{\mu R L} \tag{1}$$

where the numerator and denominator correspond respectively to the expected number of late packets throughout the playback of the video and the total number of packets in the video.

In order to obtain $P(N_i^l = k)$, we introduce N_i^b to be

$$N_i^b = \begin{cases} 0, & N_i \geq 0 \\ -N_i, & N_i < 0 \end{cases} \tag{2}$$

N_i^b can be thought of as the number of packets that the packet arrival falls behind the playback of the video in the ith round. Expression (2) follows directly from the definition of N_i and N_i^b. We obtain $P(N_i^l = k)$ as

$$P(N_i^l = k) = \begin{cases} P(N_i^b = k), & k < \mu R \\ P(N_i^b \geq \mu R), & k = \mu R \end{cases} \tag{3}$$

Note that while the number of late packets N_i^l in the ith round is at most μR, N_i^b can be larger than μR. When $N_i^b \geq \mu R$, we have $N_i^l = \mu R$. Therefore, $P(N_i^l = \mu R) = P(N_i^b \geq \mu R)$.

Summarizing the above, the fraction of late packets can be obtained from N_i, $i = 1, 2, \ldots, L$. We next describe the models for constrained and unconstrained streaming, focusing on how to derive N_i from the models.

1) Constrained streaming: Let $\{Y_i^c\}_{i=1}^{L}$ be a discrete-time Markov model for constrained streaming, where Y_i^c is the state of the model in the ith round. Y_i^c is a tuple represented as (X_i, N_i), where X_i and N_i are the state of the TCP source and the number of early packets in the ith

round respectively. Let $N_{max} = \mu\tau$. Then, as observed earlier (see Section II-A), $N_i \leq N_{max}$ for $i = 1, 2, \ldots, L$. The evolution of N_i follows

$$N_{i+1} = \min(N_{max}, N_i + S_i - \mu R)$$

where S_i is the number of packets transmitted successfully by TCP in the ith round. Observing the fact that $N_i \leq N_{max}$ for $i = 1, 2, \ldots, L$, the TCP source does not send out any packet in the $(i+1)$th round if $N_i = N_{max}$. A detailed description of the state transition probabilities for the Markov chain $\{Y_i^c\}_{i=1}^L$ and the time taken for each state transition can be found in [11].

The fraction of late packets is computed from (1). We consider videos of lengths significantly larger than the RTT. In this case, the fraction of late packets can be approximated by taking the length of the video, L, to infinity. That is, the fraction of late packets can be approximated by the steady state probability

$$\lim_{L\to\infty} \frac{\sum_{i=1}^{L} E[N_i^l]}{\mu R L} = \lim_{i\to\infty} \frac{E[N_i^l]}{\mu R}$$

We solve for the state occupancies using the steady state analysis in the TANGRAM-II modeling tool [12].

2) Unconstrained streaming: Let $\{Y_i^u\}_{i=1}^L$ be a discrete-time Markov model for unconstrained streaming, where Y_i^u is the state of the model in the ith round. Here Y_i^u only contains the state of the TCP source in the ith round, that is, $Y_i^u = X_i$. To reduce the computation overhead, the number of early packets in the ith round, N_i, is excluded from the state space. Instead, it is represented by an *impulse reward*. An impulse reward associated with a state transition is a generic means to define measure of interest (see [13] for references on reward models). We associate an impulse reward of $\rho_{yy'}$ to a transition from state $Y_i^u = y$ to state $Y_{i+1}^u = y'$, defined to be the difference between the number of packets received and played back during this transition. Denote the accumulation of this impulse reward up to the ith round as N_i'. When the transmission and playback both start at time 0, N_i' is the total number of early packets in the ith round . Since the playback starts at time τ instead of 0, the number of early packets in the ith round, N_i, has the following relationship with N_i'

$$N_i = N_i' + \mu\tau$$

which provides a way to obtain N_i from the impulse reward. The detailed description of the impulse reward can be found in [11].

The fraction of late packets is computed from (1) through a transient analysis over the length of the video using the TANGRAM-II modeling tool [12]. Note that the fraction of late packets depends heavily on the position of the round. This is because , under the assumption that the average TCP throughput is higher than the video bandwidth, the number of early packets approaches infinity and, hence, the fraction of late packets approaches 0 as the length of the video goes to infinity.

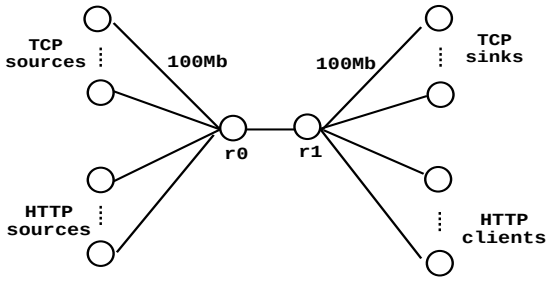


Fig. 2. Validation setting in *ns*: Packet losses are caused by buffer overflow on the link from router r_0 to r_1.

Denote P_i as the probability of having at least one late packet in the ith round. Then

$$P_i = P(N_i < 0) = P(N_i' < -\mu\tau) \tag{4}$$

Let β be the probability that at least one late packet occurs during the playback of the video. That is,

$$\beta = 1 - P(N_1 \geq 0, N_2 \geq 0, \ldots, N_L \geq 0)$$

This is a difficult quantity to compute exactly. As shown in [11], an upper bound on β is

$$\beta \leq 1 - \Pi_{i=1}^{L}(1 - P_i) \tag{5}$$

III. Model validation using *ns* simulations

In this section, we validate the models for constrained and unconstrained streaming using *ns* simulations [7]. The topology is shown in Fig. 2. Multiple TCP and HTTP sources are connected to router r_0 and their corresponding sinks connected to router r_1. Each HTTP source contains 16 connections. The HTTP traffic is generated using empirical data provided by *ns*. The bandwidth and queue length of a link from a source/sink to its corresponding router are 100 Mbps and 1000 packets respectively. The propagation delay of the link from a source/sink to its corresponding router is uniformly distributed in $[10, 20]$ ms. One of the TCP flows is used to stream video, referred to as the video stream. For this video stream, denote the round trip propagation delay as D; the average loss rate as p; the RTT as R and the value of the first retransmission timer as R_{TO}. For simplicity, R_{TO} is rounded to be a multiple of R. We further define $T_O = R_{TO}/R$. Since R_{TO} is based on the average and the variance of round trip times, T_O reflects the variation of the RTTs.

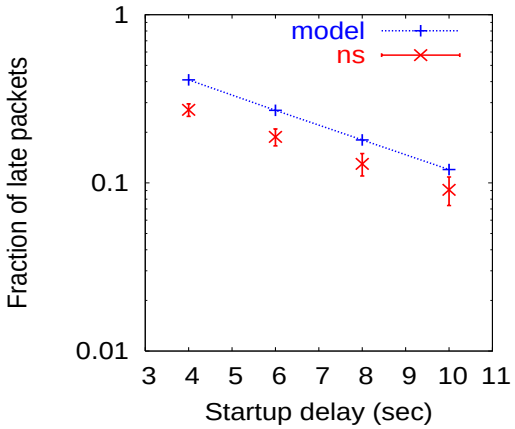


Fig. 3. Constrained streaming: The fraction of late packets versus the startup delay for a 7000-second video.

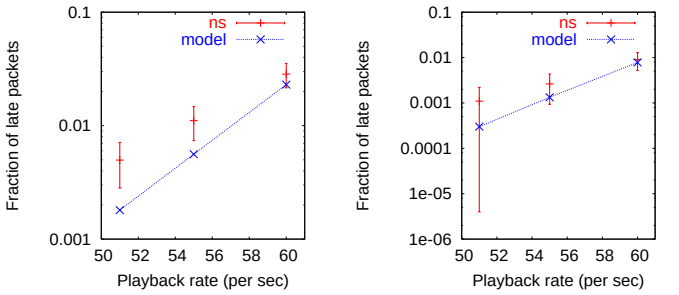


Fig. 4. Unconstrained streaming: The fraction of late packets versus the video playback rate for the startup delay of 4 seconds (a) and 6 seconds (b).

The link from router r_0 and r_1 forms a bottleneck link where packet losses occur due to buffer overflow. We create different settings by varying the bandwidth, buffer size and the propagation delay of the bottleneck link as well as the number of flows (TCP and HTTP) traversing the bottleneck link. For each setting, we run multiple simulations. For the purpose of late-packet model validation, when the loss (packet drop) rate in the model is p, ideally, the loss rate from the simulation should be p or very close to p. However, due to the randomness in the background traffic, the loss rate for the video stream in different runs may vary significantly in a fixed setting. We therefore select the runs with loss rate in the range of $(1 \pm \epsilon)p$, where $\epsilon < 15\%$, for model validation. (Recall that our goal here is to validate our model for predicting late-arriving packets for a given value of p). In each setting, we compare the result predicted by the model to that obtained from the simulations. The 95% confidence intervals for the simulation are obtained from the selected runs.

A. Model validation for constrained streaming

We validate the model for constrained streaming in four settings as listed in Table II. In these settings, the number of TCP sources varies from 6 to 10. The number of HTTP sources is 15, 30 or 40. The buffer size of router r_0 ranges from 50 to 100 packets. The bandwidth of the link from r_0 to r_1 is 3.7 or 5 Mbps. The propagation delay from r_0 to r_1 is 5 or 40 ms. A TCP flow is associated with a CBR source for video streaming. The playback rate of the video is 25 or 50 packets per second and each packet is 1500 bytes. Therefore, the bandwidth of the video is 300 or 600 Kbps. The various parameters for the video stream are listed in Table II: The round trip propagation delay is 50 or 120 ms; the loss rate ranges from 0.004 to 0.019; R ranges from 160 to 210 ms and T_O ranges from 2 to 4. In each setting, the fraction of late packets predicted by the model is compared to that from the simulation. We next describe the validation for one setting in detail; the results for other settings being similar.

In this setting, 10 TCP sources and 40 HTTP sources are connected to router r_0. The video stream has a playback rate of 25 packets per second. The round trip propagation delay of this video stream, D, is 120 ms. We generate 60 simulation runs. Each run lasts for 7000 seconds. We assume the video length to be 7000 seconds, corresponding to the entire length of a simulation run. The average loss rate of all the runs is 0.019. We use $p = 0.019$ in the model and select runs with loss rates in the range of 0.017 to 0.021 for the reason given earlier. Among the selected 39 runs, the values of R and T_O are close with the average of 210 ms and 2 respectively. These values are used in the model to obtain the fraction of late packets. Fig. 3 depicts the fraction of late packets versus the startup delay predicted by the model and obtained from the simulation. We observe a good match between the model and the simulation.

B. Model validation for unconstrained streaming

We validate the model for unconstrained streaming in four settings as listed in Table III. A TCP flow is used for unconstrained video streaming. The various parameters of this video stream (including p, R, T_O and the average throughput) are estimated and listed in Table III. For each setting, we vary the playback rate of the video and compare the results from the model to those from the simulation. We next describe one setting in detail; the results for other settings are similar.

In this setting, 5 TCP sources and 30 HTTP sources are connected to router r_0. We generate 1000 simulation runs. Each run lasts for 200 seconds. We assume the length of the video to be 80 seconds, corresponding to approximately the initial 80 seconds of a simulation run. The

# of sources		Link from router r_0 to r_1			Parameters of the video stream				
TCP	HTTP	Prop. delay (ms)	B.w. (Mbps)	Buffer (pkts)	μ (per sec)	D (ms)	p	R (ms)	T_O
10	40	40	3.7	50	25	120	0.019	210	2
6	30	40	3.7	50	25	120	0.0065	210	2
7	40	5	3.7	100	25	50	0.004	280	3
6	15	5	5	80	50	50	0.0074	160	4

TABLE II

CONSTRAINED STREAMING: VARIOUS SETTINGS FOR THE MODEL VALIDATION IN *ns*.

# of sources		Link from router r_0 to r_1			Parameters of the video stream				
TCP	HTTP	Prop. delay (ms)	B.w. (Mpbs)	Buffer (pkts)	D (ms)	p	R (ms)	T_O	tptr. (pkts per sec.)
9	40	40	3.7	50	120	0.022	220	2	30.8
5	30	40	3.7	50	120	0.006	195	2	66.5
9	40	5	5	100	50	0.015	162	3	46.1
5	30	5	5	100	50	0.014	110	3	71.4

TABLE III

UNCONSTRAINED STREAMING: VARIOUS SETTINGS FOR THE MODEL VALIDATION IN *ns*.

average loss rate of the video stream in the 1000 runs is 0.014. We use $p = 0.014$ in the model and select the runs with loss rate between 0.012 to 0.016. There are a total of 554 such runs. For the selected runs, the average RTT and T_O are 110 ms and 3 respectively; the average TCP throughput is 71.4 packets per second. We set the playback rate of the video to be 51, 55 and 60 packets per second. That is, the available TCP throughput is 40%, 30%, 20% higher than the video playback rate. Fig. 4(a) and (b) depict the fraction of late packets for various playback rates with startup delays of 4 and 6 seconds respectively. Both the results predicted by the model and measured from the simulation are shown in the figures. For both startup delays, the widths of the confidence intervals are large for low playback rates and decrease with the video playback rate. The match between the model and the simulation for a startup delay of 6 seconds is better than that for a startup delay of 4 seconds. We conjecture that this is due to the variation in the RTT encountered by the video stream, which manifests itself more prominently for the startup delay of 4 seconds in the simulation and is not captured by the model.

For a startup delay of 6 seconds, at playback rates of 51, 55 and 60 packets per second, the probabilities of experiencing no late packets throughout an 80-second video are 0.02, 0.04 and 0.09 respectively from the simulation. The upper bounds on these probabilities given by (5) are 0.18, 0.58 and 0.99 respectively. The upper bounds are not very close to the simulation results. This is likely due to the independence assumption used in deriving the bound [11].

IV. MODEL VALIDATION USING EXPERIMENTS OVER THE INTERNET

In this section, we validate the models for constrained and unconstrained streaming using experiments conducted over the Internet. In each experiment, we stream a video using TCP from one site to another site and use *tcpdump* [14] to capture the packet timestamps. The average loss rate p, average RTT R and T_O of this TCP flow are estimated from the *tcpdump* traces. We use Linux-based machines for all the experiments.

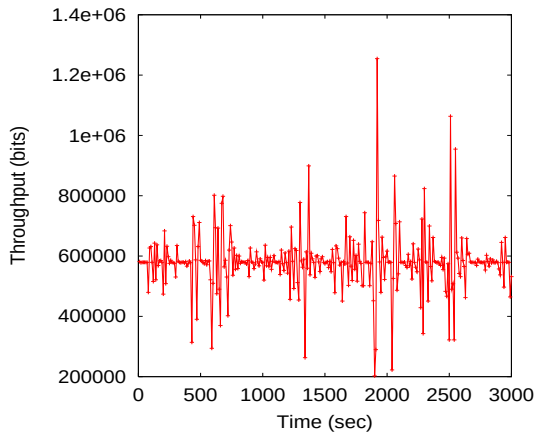


Fig. 5. The TCP throughput of an experiment from USC to the client in the resident house in Amherst, MA.

A. Model validation for constrained streaming

We first focus on constrained streaming. A CBR video is transmitted using TCP from University of Southern California (USC) to a client in a resident house in Amherst, Massachusetts. The resident house uses a cable modem for its Internet connection. The playback rate of the video is 40 or 50 packets per second and each packet consists of 1448 bytes. That is, the bandwidth of the video is approximately 480 or 600 Kbps. We conducted 8 experiments from March 3 to March 7, 2003 at randomly chosen times; each experiment lasting for one hour. For each experiment, we plot the time series of the TCP through-

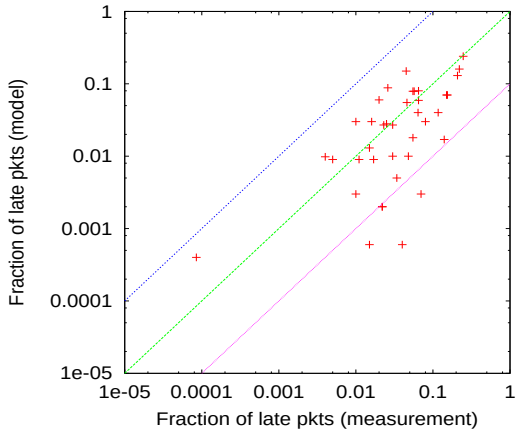


Fig. 6. Constrained streaming: The fraction of late packets obtained from the measurements versus that predicted by the model.

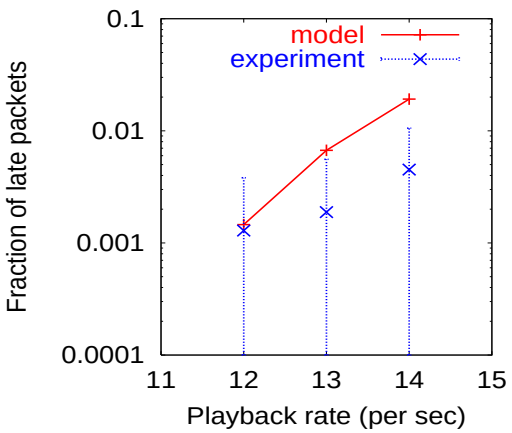


Fig. 7. Unconstrained streaming: The fraction of late packets versus the playback rate of the video for experiments from UMass to Italy.

put, where each point is the average throughput over a 10-second interval. Based on the throughput series, we choose stationary segments of length 500 to 1000 seconds that exhibit variations in throughput, implying the occurrence of congestion. The segments are chosen by visual inspection although more rigorous methods can be used [15]. We use one trace to illustrate our procedure. Fig. 5 plots the TCP throughput averaged over every 10 seconds for one experiment. We choose the first, second and third 1000 seconds of the trace as three segments to validate the model against the measurements. Each segment is treated as a 1000-second video. The loss rate, RTT and T_O are obtained from the data segment and used in the model.

We obtained a total of 11 segments from the experiments. The startup delay varies between 4 to 10 seconds. Fig. 6 presents a scatterplot showing the fraction of late packets for various startup delays obtained from the measurements versus that predicted by the model. The 45 degree line starting at the origin represents a hypothetical perfect match between the measurements and the model. Along the upper and lower 45 degree lines, the fraction of late packets from the model is respectively 10 times higher and lower than that from the measurements. All but 4 scatterplot points fall within an order of magnitude of each other. These 4 points are for the startup delays of 8 or 10 seconds. The reason why the fraction of late packets from these measurements is 10 times higher than that from the model might be because the number of samples in the data segment is not sufficient.

B. *Model validation for unconstrained streaming*

We next compare model prediction to measurements taken over the Internet for unconstrained streaming. In each experiment, we run 8 parallel TCP connections to obtain a group of runs with similar TCP parameters (loss rate, RTT and T_O). The bandwidth from USC to the client in the resident house is too low to run parallel TCP connections. We therefore chose a different network path, where the server is at University of Massachusetts (UMass) and the client is in Universita' dell'Aquila, Italy. Each experiment lasts for 1 hour. We then divide the trace for each TCP flow into multiple segments, each of 100 seconds. Each 100-second segment is treated as a 100-second video. We use $p = 0.031$ in the model and select 266 segments having loss rate between 0.027 and 0.035. For the selected segments, the RTT is 305 ms and $T_O = 1$. The average throughput is 15.2 packets per second. We set the playback rate of the video to be 12, 13 and 14 packets per second. Correspondingly, the available TCP throughput is 27%, 17% and 9% higher than the playback rate of the video. Fig. 7 plots the fraction of late packets for various playback rates when the startup delay is 6 seconds. The fraction of late packets predicted by the model are higher than those from the measurements. This might be because, at the beginning of the video streaming, the window size is always one in the model while it may be larger than one in the measurement data segment.

V. Exploring the parameter space

In this section, we vary the model parameters in constrained and unconstrained streaming to study the impacts of these parameters on performance. In doing so, we provide guidelines as to when the use of TCP leads to satisfactory performance.

We assume the startup delay is 4 to 10 seconds. The loss rate, R and T_O in the model jointly determine the available TCP throughput measured in packets. For convenience, we refer to these three parameters as *TCP pa-*

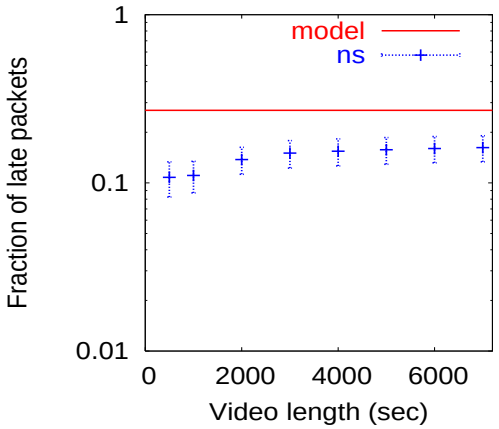


Fig. 8. Constrained streaming: The fraction of late packets versus the length of the video for a startup delay of 6 seconds.

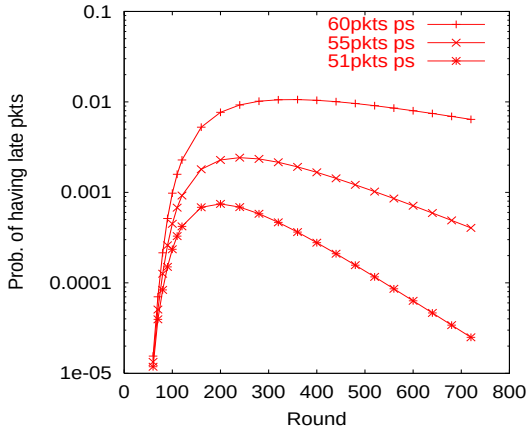


Fig. 9. Unconstrained streaming: Probability of having at least one late packet in a round versus rounds for various playback rates of the video.

rameters. We set the values of the TCP parameters to represent a wide range of scenarios. The loss rate is varied in the range of 0.004 to 0.08, since a loss rate of 0.004 is relatively low and a loss rate of 0.08 is high for streaming. Based on previous studies [16] and our measurements, we vary R in the range of 40 ms to 320 ms. Previous work shows that the median RTT between two sites on the same coast in the US is 50 ms, while the median RTT between west-coast and east-coast sites is 100 ms [16]. We observed a RTT of 300 ms in our experiments (see Section IV). We vary T_O in the range of 1 to 3, based on several measurements from Linux machines in [17] and our measurements.

Denote the available TCP throughput as T packets per second. Then T/μ represents how much the achievable TCP throughput is higher than the video playback rate. In the following, we first explore how the performance of constrained and unconstrained streaming varies with the length of the video. We then investigate the effect of T/μ on the performance and the sensitivity of the performance to the various parameters in the model. Following that, we identify the conditions under which using TCP provides a satisfactory viewing experience. At the end, we summary the key results and describe the implication of the results to the stop-and-wait playout strategy.

A. *The effect of video length on the performance*

We first use the setting in Section III-A to illustrate the fraction of late packets as a function of the video length in constrained streaming. The startup delay is 6 seconds and the length of the video ranges from 500 to 7000 seconds. Fig. 8 plots the fraction of late packets versus the video length from the model and the *ns* simulation. We observe that for videos longer than 2000 seconds, the fraction of late packets for different video lengths from the simulation is similar and closer to the prediction from the model than for shorter videos. Throughout this section, we assume the video for constrained streaming is sufficiently long so that stationary analysis can be used to obtain the fraction of late packets.

In unconstrained streaming, to investigate how the fraction of late packets varies with the video length, we obtain P_i $(i = 1, \ldots, L)$, the probability of having at least one late packet in the ith round from (4) (see Section II-C.2). Fig. 9 plots P_i over the length of the video for the setting in Section III-B. Here the video is 730 rounds (80 seconds). Fig. 9 indicates that the fraction of late packets is low at the beginning of the video, increases to a peak value and then decreases over time. This can be explained as follows. At the beginning of the playback, the probability of having late packet in a round is low due to the packets accumulated in the client local buffer during the startup delay. Subsequently, packets are played out while at the same time being accumulated in the client buffer. The number of early packets in the buffer increases with time since, on average, the available TCP throughput is higher than the playback rate of the video. Therefore, the probability of having late packets in a round reaches a peak value at approximately the 200th round (the 22nd second) and subsequently decreases over time as the number of early packets in the buffer increases.

In Fig. 9, the probability of having late packet in the 730th round decreases to the order of 10^{-5} and 10^{-4} for the playback rates of 51 and 55 packets per second respectively. This indicates that, after 730 rounds, the fraction of late packets is approximately inversely proportional to the length of the video, since the probability of having late packet after 730 rounds is close to 0. This is confirmed by the simulation results. In general, to obtain the fraction of

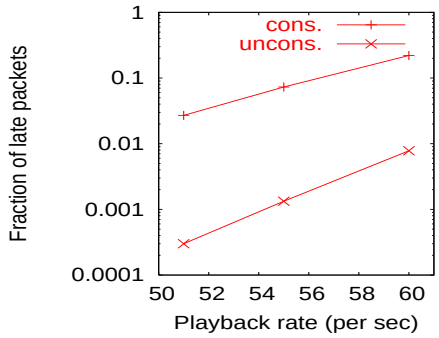


Fig. 10. The performance of constrained and unconstrained streaming when varying the video playback rate and fixing the TCP parameters.

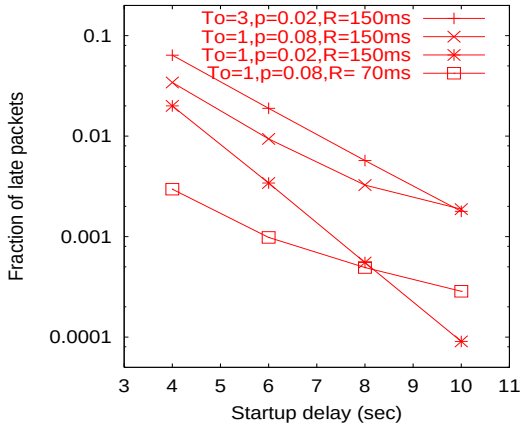


Fig. 11. Constrained streaming: Sensitivity to various parameters (p, R, T_O and the playback rate of the video), $T/\mu = 1.6$.

late packets, f, for a video of L rounds, it is sufficient to obtain the fraction of late packets in the initial l rounds of the video, denoted as f_l, such that P_l is close to 0. Then $f = lf_l/L$. Throughout this section, we assume the video for unconstrained streaming is 80 to 100 seconds.

B. The effect of T/μ on the performance

We now explore the effect of T/μ on the performance of constrained and unconstrained streaming. The fraction of late packets decreases as T/μ increases. This is intuitive since, as T/μ increases, packets are accumulated in the client's local buffer faster relative to the playback rate of the video. We increase T/μ by either decreasing the video playback rate or one of the TCP parameters while fixing the other parameters. We only show one example where we vary the video playback rate and fix the TCP parameters. We use the setting described in Section III-B, where $p = 0.014$, $R = 110$ ms, $T_O = 3$ and the available TCP throughput is 71.4 packets per second. The playback rate of the video is chosen to be 51, 55 and 60 packets per second, corresponding to $T/\mu = 1.4$, 1.3 and 1.2 respectively. Fig. 10 shows the fraction of late packets for the various playback rates with a startup delay of 6 seconds under constrained and unconstrained streaming. For constrained streaming, the video is assumed to be on the order of thousands of seconds. For unconstrained streaming, the length of the video is 80 seconds. We observe that as the playback rate of the video decreases (T/μ increases), the fraction of late packets decreases exponentially in both constrained and unconstrained streaming. For the same playback rate, the fraction of late packets in constrained streaming is higher than that in unconstrained streaming by an order of magnitude. The difference between constrained and unconstrained streaming becomes even more dramatic as the video length increases, since the fraction of late packets is similar for long videos in constrained streaming and decreases with the length of the video in unconstrained streaming.[1] It is not surprising that unconstrained streaming can significantly outperform constrained streaming, since the maximum number of early packets in the latter is no more than the product of the startup delay and the video playback rate, while no such limit exists in the former.

C. Sensitivity of performance to the various parameters

We next fix T/μ and study the sensitivity of the performance to the various model parameters. Fig. 11 shows the fraction of late packets for four sets of TCP parameters in constrained streaming. The playback rate of the video is chosen so that $T/\mu = 1.6$ for all the settings. The startup delay is between 4 and 10 seconds. In Fig. 11, $T_O = 1$, 3; $p = 0.02$, 0.08 and $R = 70$, 150 ms. For $p = 0.02$ and $R = 150$ ms, the fraction of late packets for various startup delays decreases dramatically when T_O decreases from 3 to 1, especially for large startup delays. For $p = 0.08$ and $T_O = 1$, the decrease is close to an order of magnitude when R decreases from 150 ms to 70 ms. For $T_O = 1$ and $R = 150$ ms, the decrease is also large for long startup delays when p decreases from 0.08 to 0.02. The above shows that the performance of constrained streaming is not solely determined by T/μ but also depends on the values of the various parameters in the model. For a fixed value of T/μ, the performance improves when reducing one of the TCP parameters.

Fig. 12 shows the probability of having at least one late packet in a round for four sets of TCP parameters in unconstrained streaming, where rounds are represented using seconds. The playback rate of the video is chosen so that $T/\mu = 1.3$ for all the settings. The startup delay is

[1] We ignore the initial increasing trend since its duration is usually very short (see Section V-A).

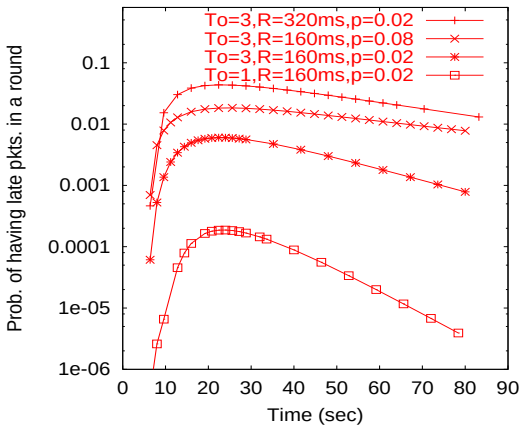


Fig. 12. Unconstrained streaming: Sensitivity to the various parameters (p, R, T_O and the playback rate of the video), $T/\mu = 1.3$.

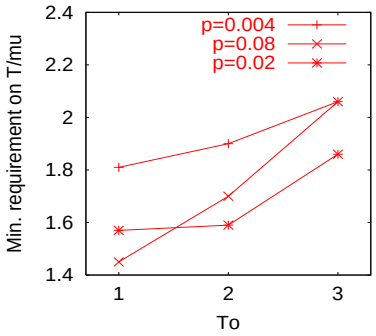


Fig. 13. Constrained streaming: The minimum required value of T/μ for a satisfactory performance.

6 seconds. In Fig. 12, $p = 0.02, 0.08$; $R = 160$, 320 ms and $T_O = 1, 3$. The probability of having late packets in a round is reduced by approximately an order of magnitude for all the rounds when R decreases from 320 to 160 ms for $T_O = 3$ and $p = 0.02$; when p decreases from 0.08 to 0.02 for $R = 160$ ms and $T_O = 3$; and when T_O decreases from 3 to 1 for $R = 160$ ms and $p = 0.02$. This above shows that the performance of unconstrained streaming is sensitive to the various parameters in the model and the performance improves when reducing one of the TCP parameters.

D. Conditions for satisfactory performance

In Section V-C, we observe that, for a fixed value of T/μ, different sets of parameters can lead to dramatically different performance in both constrained and unconstrained streaming. In other words, different sets of parameters place different requirements on the value of T/μ needed to achieve the same performance. We next identify the conditions under which the performance when using TCP is satisfactory.

1) Constrained streaming: We first define a criterion of satisfactory performance for constrained streaming. A viewing experience is defined to be satisfactory if the fraction of late packets is below 10^{-3} when the startup delay is 10 seconds or less. We use this criterion simply as an example since a startup delay no more than 10 seconds is still tolerable and a fraction of late packets below 10^{-3} is low for viewing a video. Other criteria can also be defined. For fixed T_O and p, we find the maximum value of R and, hence, the minimum value of T/μ such that the criteria is satisfied.

Fig. 13 shows the minimum required value of T/μ as a function of T_O and p for a playback rate of 25 packets per second. When T_O increases from 1 to 3, the increment of the minimum required value of T/μ is from 1.5 to 2.1 for $p = 0.08$; from 1.6 to 1.9 for $p = 0.02$ and from 1.8 to 2.1 for $p = 0.004$. For a fixed T_O, the requirement on T/μ for the lowest loss rate (i.e., $p = 0.004$) is the most stringent. We observe similar requirements on T/μ for other video playback rates (figures not shown). In summary, the minimum required value of T/μ for constrained streaming is between 1.5 and 1.1 for all the settings we study.

2) Unconstrained streaming: We explore the performance of unconstrained streaming in two extreme cases. The best case is when all the TCP parameters are at the lowest values, that is, $p = 0.004$, $R = 40$ ms and $T_O = 1$. In this case, the requirement on T/μ to achieve a certain performance is the least stringent. The worst case is when all the TCP parameters are at the highest values, that is, $p = 0.08$, $R = 320$ ms and $T_O = 3$, where the requirement on T/μ to achieve a certain performance is the most stringent. We use a startup delay of 6 seconds and a video of 100 seconds.

In the best case, when T/μ is as low as 1.1, the probability of having late packets reaches a maximum value at the 10th second and then decreases with the length of the video. Furthermore, the probability of having late packets in a round is less than 10^{-6} for all the rounds. This indicates that an available throughput slightly higher than the playback rate of the video is sufficient to achieve a good performance in this case. In the worst case, the average available TCP throughput is 6 packets per second. The video playback rate is set to 3 and 4 packets per second, corresponding to $T/\mu = 2$ and 1.5 respectively. When $T/\mu = 1.5$, the fraction of late packets is 0.02. When $T/\mu = 2$, the performance becomes reasonably good: The fraction of late packets decreases to 0.005 and the probability of having late packets at the 100th second is 0.001. In summary, for all the settings we consider, the minimum requirement on T/μ to achieve a good performance is between 1.1 to 2 in unconstrained streaming.

E. Summary of results and implications to stop-and-wait playout

The key results from our exploration of parameter space are:

- The fraction of late packets is similar for long videos in constrained streaming while decreases with the video length (after a short duration of increasing trend at the beginning of the playback) in unconstrained streaming.
- The performance of constrained and unconstrained streaming improves dramatically when increasing the value of T/μ. Under the same conditions, unconstrained streaming can significantly outperform constrained streaming.
- The performance of constrained and unconstrained streaming is not solely determined by T/μ but is sensitive to the values of the various parameters in the models.
- In the settings we study, the minimum requirement on T/μ for a good performance (by the criteria we defined) is between 1.5 to 2.1 in constrained streaming. For unconstrained streaming, the requirement on T/μ is in the range of 1.1 to 2.

Our models for constrained and unconstrained streaming assume continuous playback at the client, which differs from the stop-and-wait playout strategy commonly used in the commercial streaming system. However, the fraction of late packets obtained from our models provides some insight into the likelihood that a client needs to stop during the playback of the video, and, hence, the performance under the stop-and-wait playout strategy. Furthermore, a stop and wait during the playback can be regarded as extending the startup delay in our models. Therefore, we conjecture that the performance under stop-and-wait strategy will be better than that predicted from our model under the same conditions.

VI. Conclusions and future work

In this paper, we developed discrete-time Markov models for constrained and unconstrained streaming that corresponds to live and stored video streaming respectively. Our validation using *ns* and Internet experiments showed that the performance predicted by the models are accurate. Using the models, we studied the effect of the various parameters on the performance of constrained and unconstrained streaming. In doing so, we provided guidelines as to when using TCP directly for streaming renders satisfactory performance.

As future work, we are pursuing in two directions: (i) Develop an exact model for the stop-and-wait playout strategy. (ii) Explore the use of parallel TCP flows for video streaming when the average throughput of one TCP flow is lower than the video playback rate.

References

[1] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Networking*, 1999.

[2] N. Seelam, P. Sethi, and W. chi Feng, "A hysteresis based approach for quality, frame rate, and buffer management for video streaming using TCP," in *Proc. of the Management of Multimedia Networks and Services 2001*, 2001.

[3] C. Krasic and J. Walpole, "Priority-progress streaming for quality-adaptive multimedia," in *ACM Multimedia Doctoral Symposium 2001*, (Ottawa, Canada), October 2001.

[4] P. de Cuetos and K. W. Ross, "Adaptive rate control for streaming stored fine-grained scalable video," in *Proc. of NOSSDAV*, May 2002.

[5] P. de Cuetos, P. Guillotel, K. W. Ross, and D. Thoreau, "Implementation of adaptive streaming of stored MPEG-4 FGS video over TCP," in *International Conference on Multimedia and Expo (ICME02)*, August 2002.

[6] "Realplayer." $http://www.real.com$.

[7] S. McCanne and S. Floyd, "ns-LBNL network simulator." http://www-nrg.ee.lbl.gov/ns/.

[8] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Tech. Rep. 99-02, Department of Computer Science, University of Massachusetts, Amherst, 1999.

[9] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley, "On the autocorrelation structure of TCP traffic," *Computer Networks Journal Special Issue on Advances in Modeling and Engineering of Long-Range Dependent Traffic*, 2002.

[10] W. Stevens, *TCP/IP Illustrated, Vol. 1*. Addison-Wesley, 1994.

[11] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "A model for TCP-based video streaming," tech. rep., Department of Computer Science, University of Massachusetts, Amherst, 2003.

[12] E. de Souza e Silva and R. M. M. Leao, "The TANGRAM-II environment," in *Proc. of the 11th Int. Conf. on modeling tools and techniques for computer and communication system performance evaluation (TOOLs 2000)*, May 2000.

[13] E. de Souza e Silva and H. R. Gail, "An algorithm to calculate transient distribution of cumulative rate and impulse based reward," *Stochastic models*, vol. 14, no. 3, pp. 509–536, 1998.

[14] "tcpdump." http://www.tcpdump.org/.

[15] J. S. Bendat and A. G. Peirsol, *Random Data Analysis and Measurement Procedures*. John Wiley & Sons, 1986.

[16] B. Huffaker, M. Fomenkov, D. Moore, and K. Claffy, "Macroscopic analyses of the infrastructure: Measurement and visualization of Internet connectivity and performance," in *A Workshop on passive and active measurements*, (Amsterdam), April 2001.

[17] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM*, (Vancouver, CA), pp. 303–314, 1998.