Multimedia Streaming via TCP: An Analytic Performance Study

Bing Wang, Jim Kurose, Prashant Shenoy, Don Towsley

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

UMass CMPSCI Technical Report 04-21

***Abstract*—**
TCP is widely used in commercial media streaming systems, with recent measurement studies indicating that a significant fraction of Internet streaming media is currently delivered over HTTP/TCP. These observations motivate us to develop analytic performance models to systematically investigate the performance of TCP for both live and stored media streaming. We validate the models via *ns* simulations and experiments conducted over the Internet. Our models provide guidelines indicating the circumstances under which TCP streaming leads to satisfactory performance, showing, for example, that TCP generally provides good streaming performance when the achievable TCP throughput is roughly twice the media bitrate, with only a few seconds of startup delay.

I. Introduction

The rapid deployment of broadband connectivity to the home via cable modem and digital subscriber loop (DSL) technologies has resulted in a significant growth in streaming media usage. The conventional wisdom for media streaming is to use UDP, rather than TCP, as the transport protocol. The primary reason for not using TCP is that the backoff and retransmission mechanisms in TCP can lead to undesirable end-to-end delays that violate the timeliness requirement for streaming media. Due to these limitations, much of the research over the past decade focused on developing UDP-based streaming protocols, providing mechanisms for TCP-friendliness and loss recovery (see e.g., [1], [2], [3], [4]).

In spite of the conventional wisdom that TCP is not desirable for streaming and the large body of literature on UDP-based streaming, TCP is widely used in commercial streaming systems. For instance, Real Media and Windows Media, the two dominant streaming media products, both support TCP streaming. Furthermore, a recent measurements study has shown that a significant fraction of commercial streaming traffic uses TCP [5]. This study analyzed 4.5 million session-level logs for two commercial streaming servers over a four month period and found that 72% and 75% of the on-demand and live streaming traffic, respectively, used TCP. Moreover, 27% and 47% of the on-demand and live streaming traffic, respectively, used HTTP. The wide use of HTTP streaming is particularly interesting: HTTP streaming is perhaps the simplest streaming protocol, since no rate adaptation is employed at the application level, unlike other TCP streaming approaches [6], [7], [8], [9]; further, no additional mechanisms are necessary to ensure TCP friendliness or to recover loss, unlike UDP-based streaming.

In this paper, motivated by the wide use of TCP streaming in commercial systems, we seek to answer the following question: *Under what circumstances can TCP streaming provide satisfactory performance?* To answer this question, we study a baseline streaming scheme which uses TCP directly for streaming. This baseline streaming scheme is similar to HTTP streaming and is henceforth referred to as *direct TCP streaming*. We study the performance of direct TCP streaming using analytical models. Our models enable us to systematically investigate the performance of TCP streaming under various conditions, a task that is difficult when using empirical measurements or simulation alone. To the best of our knowledge, this is the first analytical study of using TCP for streaming.

The main contributions of this paper are:

- We build upon the TCP model in [10], [11] to develop discrete-time Markov models for *live* and *stored* video streaming. The models are validated using *ns* [12] simulation and Internet experiments.
- Using the models, we explore the parameter space (i.e., loss rate, round trip time and timeout value in TCP as well as video playback rate) to provide guidelines as to when direct TCP streaming leads to satisfactory performance. Our results show that direct TCP streaming generally provides good performance when the achievable TCP throughput is roughly twice the the video bitrate, with only a few seconds of startup delay.

Our study has the following implication. A large fraction of streaming video clips on the Internet today are encoded at bit rates below 300 Kbps (e.g., [13] finds that around 80% of videos encoded for Windows Media and Real Media are below 300 Kbps). On the other hand, most DSL and cable modem connections support download rates of 750 Kbps - 1 Mbps. In the situations where the end-end available bandwidth is only constrained by the last-mile access link, our performance study thus indicates that direct TCP streaming may be adequate for many broadband users.

The rest of the paper is organized as follows. In Section II, we review related work on TCP-based streaming and TCP modeling. Section III 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 IV and V respectively. Performance study based on the models is presented in Section VI. Finally, Section VII concludes the paper.

II. Related work

TCP-based streaming has several advantages. First, TCP is by definition TCP friendly. Second, reliable transmission provided by TCP removes the need for loss recovery at higher levels. Furthermore, in practice, streaming contents using TCP are more likely to pass through firewalls. A number of existing research efforts that use TCP for streaming [6], [7], [8], [9] combine *client-side buffering* and *rate adaptation* 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. Direct TCP streaming does not deal with long-term fluctuations and only employs client-side buffering. It is hence much simpler than [6], [7], [8], [9]. Furthermore, it does not require layered video as in [8], [9]. In this paper, we focus on the performance of direct TCP streaming. We expect the performance of more sophisticated approaches like [6], [7], [8], [9] to be better. However, the performance of these approaches and the comparison of different approaches are beyond the scope of this study.

There is a vast literature on TCP modeling. We only review some studies that are most related to our work. Most of the models are on the performance of TCP when TCP is used for file transfer. Among them, majority of the models are for long-lived flows (e.g., [14], [15], [10], [16], [17]); some are for short-lived flows (e.g., [18], [19]). In particular, [10] and [11] use Markov models to capture the congestion control and avoidance mechanisms in TCP to study the steady-state TCP throughput and the autocorrelation structure in TCP traffic respectively. Our work differs from all the above in that we consider the real-time requirement when using TCP for streaming. We build upon the TCP model in [10], [11] to develop Markov models for streaming. The reason why we use Markov models is two fold. First, they capture the detailed congestion control and avoidance mechanisms in TCP. The timeout mechanism, which leads to a drastic decrease in congestion window size, is particularly important for modeling streaming using TCP (see Section VI). Secondly, it is convenient to perform transient analysis using Markov model, which is required for stored video streaming (see Section III).

An earlier study [20] combines TCP modeling and video transmission. The author provides a model to obtain the probability distribution of TCP congestion window size, which is further applied to determine the TCP-friendly transmission rate for a non-TCP flow used to transmit video. Our work differs from the above study in that we study TCP-based streaming instead of determining the TCP-friendly transmission rate for UDP-based streaming.

III. 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 video bitrate. 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. The client allows a startup delay on the order of seconds, which is a common practice in commercial streaming products. 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.

Measurement studies show that most of the videos in the Interent are CBR (constant bit rate) videos [13]. We therefore consider a CBR 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 plays back at a constant rate of μ packets per second. A packet arriving later than its playback time is referred to as a *late packet*. We assume a late packet leads to a glitch during the playback and use the *fraction of late packets*, i.e., the probability that a packet is late, to measure the performance. Strictly speaking, fraction of late packets does not correspond directly to viewing quality. To the best of our knowledge, there is no known metric which corresponds directly to viewing quality for videos in general. We therefore use fraction of late packets as a rough metric for the performance.

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 achievable TCP throughput in order to fully utilize the TCP throughput. 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 associated 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 achievable 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 $\mu\tau$.

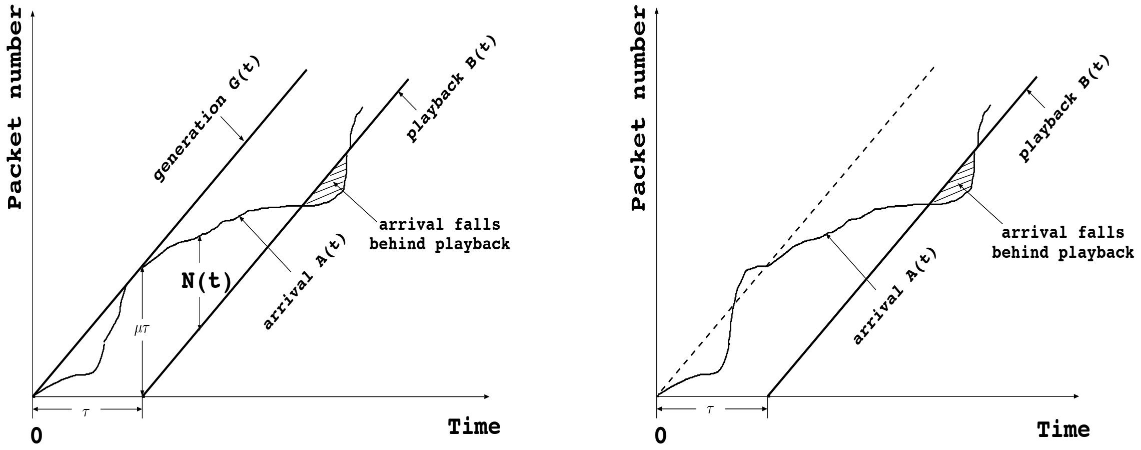
Notation	**Definition**
μ	Playback rate of the video (packets per second)
τ	Startup delay (seconds)
X_i	State of the TCP source in the ith round
S_i	Number of packets transmitted successfully by TCP in the ith round
R	Round trip time (seconds)
L	Length of the video (measured in rounds)
f	Fraction of 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.

As described above, a negative value of $N(t)$ indicates that late packets occur at time t. We need to model $N(t)$ during the playback of the video in order to obtain the fraction of late packets. For this purpose, we extend the model for TCP in [10], [11] to incorporate the specific characteristics of constrained and unconstrained streaming. In Section III-C.1, we construct a Markov model for constrained streaming where the number of early packets is one component in the model. In Section III-C.2, we provide a transient analysis technique for unconstrained streaming. Before describing the models for constrained and unconstrained streaming, we first briefly describe the model in [10], [11].

B. Model for TCP

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 [21]. 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



(a) Constrained streaming.

(b) Unconstrained streaming.

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

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 [10], [11], 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 [10], [11], 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 back-off 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}. For instance, when there is no packet loss from state $X_i = (w, c, l, e, r)$ to $X_{i+1} = (w', c', l', e', r')$, we have $S_i = w$, the window size in the ith round. Detailed description of S_i can be found in [10], [11] and Appendix I. 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 present 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 III-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 $N_i^l \in \{0, 1, \ldots, \mu R\}$, where $N_i^l = 0$ indicates that no packet is late in the ith round. Let the expected number of late packets in the ith round be $E[N_i^l]$. Then

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

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

$$f = \frac{\sum_{i=1}^{L} E[N_i^l]}{\mu R L} \qquad (2)$$

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

$$N_i^b = \begin{cases} 0, & N_i \geq 0 \\ -N_i, & N_i < 0 \end{cases} \tag{3}$$

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 (3) follows directly from the definition of N_i and N_i^b. We obtain $P(N_i^l = k)$ as

$$P(N_i^l = k) = \begin{cases} P(N_i^b = k), & k < \mu R \\ P(N_i^b \geq \mu R), & k = \mu R \end{cases} \tag{4}$$

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, $i = 1, 2, \ldots, L$, by applying (1), (2), (3) and (4). We next describe the models for constrained and unconstrained streaming, focusing on how to derive N_i from the models.

1) Constrained streaming: Constrained streaming can be modeled as a producer-consumer problem. The producer produces packets according to the mechanisms of TCP and stores the packets in a buffer. The consumer starts to consume the packets in the buffer from time τ at a constant rate of μR packets per round. At any time, the number of packets in the buffer is no more than N_{max}, $N_{max} = \mu\tau$. This is from an earlier observation that $N_i \leq N_{max}$ ($i = 1, 2, \ldots, L$) because of the constraint of video generation rate (see Section III-A). To satisfy this constraint, the producer stops producing packets when there are N_{max} packets in the buffer. We therefore use the following model for 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. 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, which is determined by X_i and X_{i+1}. In order to satisfy the condition 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 Appendix I.

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 stationary distribution of N_i using the steady state analysis in the TANGRAM-II modeling tool [22]. Afterwords, we compute the stationary distribution of N_i^l using (3) and (4). Finally, the fraction of late packets is computed from (2).

2) Unconstrained streaming: Unconstrained streaming can also be modeled as a producer-consumer problem. Furthermore, the number of packets in the buffer can be more than N_{max}. Therefore, it appears that solving unconstrained streaming is simpler than solving constrained streaming. This is not true and the reason is as follows. In unconstrained streaming, 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 bitrate, as the length of the video goes to infinity, the number of early packets approaches infinity and, hence, the fraction of late packets approaches 0. The fraction of late packets in the steady state (when the video is regarded as infinitely long) is thus trivial (equal to 0). To obtain the fraction of late packets over a finite video, we therefore resort to transient analysis, which is in general much more complex than steady state analysis.

We develop the following model for 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, $Y_i^u = X_i$. The number of early packets in the ith round, N_i, is excluded from the state space to reduce the size of the state space, and hence computation overhead. We introduce an *impulse reward* into the model to obtain the transient distribution of N_i. An impulse reward associated with a state transition is a generic means to define measure of interest (see [23] 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'. The TANGRAM-II modeling tool [22] provides a functionality to solve for the transient distribution of N_i' based on the algorithm in [23].

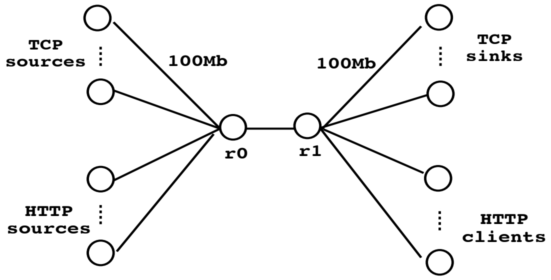


Fig. 2. Validation setting in *ns*: packet losses are caused by buffer overflow on the link from router r_0 to r_1.

We further obtain the transient distribution of N_i from that of N_i' as follows. Observe that N_i' is the total number of early packets in the ith round when the transmission and playback both start at time 0. Recall that N_i is the number of early packets in the ith round when the playback starts at time τ instead of 0. We therefore have the following relationship between N_i and N_i'

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

This relationship therefore allows us to obtain the transient distribution of N_i from that of N_i'. The detailed description of the impulse reward can be found in Appendix II.

To compute the fraction of late packets, we first solve for the transient distribution of N_i through the TANGRAM-II modeling tool [22]. Afterwords, the transient distribution of N_i^l is calculated using (3) and (4). Finally, the fraction of late packets is computed from (2).

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

$$P_i = P(N_i < 0) = P(N_i' < -\mu\tau) \qquad (5)$$

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 Appendix III, we derive an upper bound on β as

$$\beta \leq 1 - \Pi_{i=1}^{L}(1 - P_i) \qquad (6)$$

IV. Model validation using *ns* simulations

In this section, we validate the models for constrained and unconstrained streaming using *ns* simulations [12]. 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. For constrained and unconstrained streaming, we assume the video length to be 7000 and 80 seconds respectively. We vary the video length in Section VI-A. In particular, we show that the model for constrained streaming is accurate for a wide range of video lengths. We also show that, in unconstrained streaming, it is sufficient to model a relatively short video and we provide a method to obtain the fraction of late packets for longer videos.

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 to obtain a confidence interval. For a fixed setting, we found the values of R and T_o among different runs are close. However, due to the randomness in the background traffic, the loss (packet drop) rate for the video stream in different runs may vary significantly, especially in unconstrained streaming, where the video length, and hence, simulation run is short. We thus face the problem of validating a model with a given loss rate against multiple simulation runs with varying loss rates. Since our goal is to validate our model for a given value of p, we select simulation runs with loss rate close to p. In particular, we select the runs with loss rate in the range of $(1 \pm \epsilon)p$, where $\epsilon < 15\%$, for model validation. The 95% confidence intervals for the simulations are obtained from the selected runs.

In each setting, we obtain the fraction of late packets from the model and the simulation, denoted as f_m and f_s respectively. We say the model and the simulation have a good match if f_m falls within the confident interval from the simulation or $\frac{1}{5} \leq \frac{f_m}{f_s} \leq 5$. The reason for the second “loose” criterion can be explained as follows. We use the fraction of late packets to roughly measure the viewing quality. When f_m and f_s satisfy the above criterion, they correspond to similar viewing experience. For instance, fraction of late packets as 0.1 and 0.5 both correspond to bad viewing experience; fraction of late packets as 10^{-4} and 5×10^{-4} both correspond to good viewing experience, etc.

Setting #	# 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
1	10	40	40	3.7	50	25	120	1.90	210	2
2	6	15	5	5	80	50	50	0.74	165	5
3	7	40	5	3.7	100	25	50	0.40	285	3
4	6	30	40	3.7	50	25	120	0.65	210	2

TABLE II

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

Setting #	Range of loss rate $(\%)$	# of selected runs	T (pkts per sec)	T/μ	required τ (sec)
1	$[1.7, 2.1]$	40	31.7	1.27	56
2	$[0.6, 0.8]$	32	64.9	1.30	38
3	$[0.3, 0.5]$	37	49.7	1.99	22
4	$[0.6, 0.8]$	30	57.1	2.28	12

TABLE III

CONSTRAINED STREAMING: RESULTS FOR VARIOUS SETTINGS.

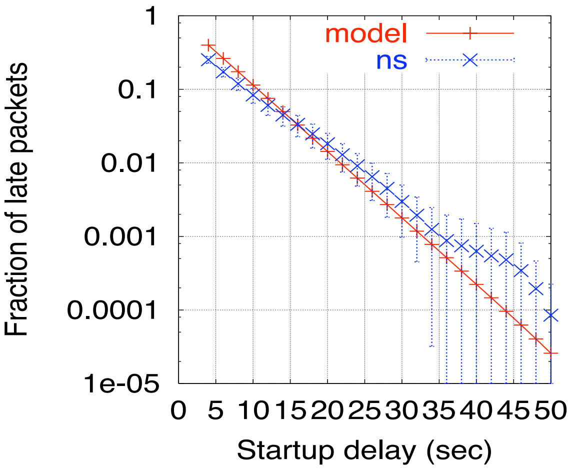


Fig. 3. Constrained streaming (Setting 1): fraction of late packets versus the startup delay for a 7000-second video.

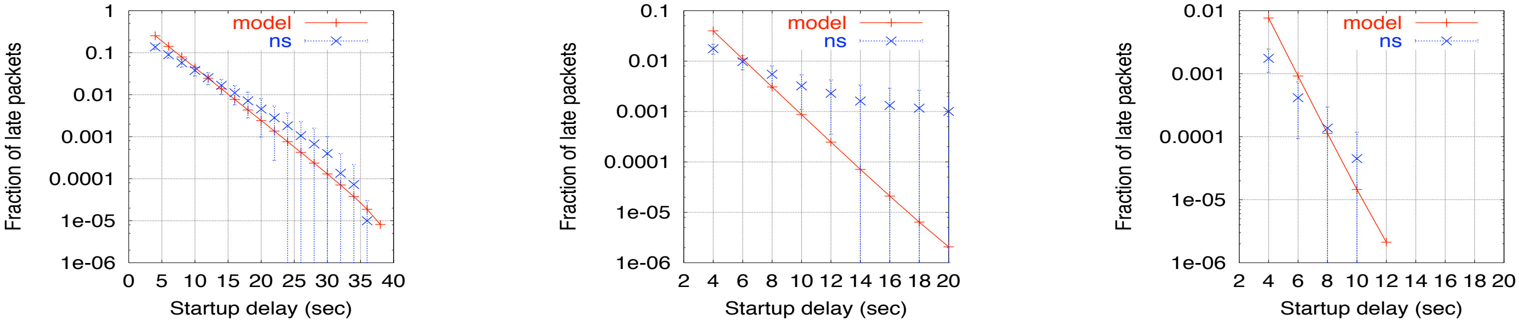
A. Model validation for constrained streaming

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 0.40% to 1.90%; R ranges from 165 to 285 ms and T_O ranges from 2 to 5. Table III lists the range of loss rate and the number of selected runs in each setting. Let T represent the available TCP throughput. Then T/μ represents how much the achievable TCP throughput is higher than the video playback rate. The required startup delay τ is the value of the startup delay at which the fraction of late packets reduces to zero. 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 (Setting 1) 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 lasting for 7000 seconds. The video length is 7000 seconds. The average loss rate of all the runs is 1.9%. We use $p = 1.9\%$ in the model and select runs with loss rates in the range of 1.7% to 2.1% 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. The validation results for the other settings (Setting 2, 3 and 4) are depicted in Fig. 4.

B. Model validation for unconstrained streaming

We validate the model for unconstrained streaming in four settings as listed in Table IV. A TCP flow is used for unconstrained video streaming. The various parameters of this video



(a) Setting 2

(b) Setting 3

(c) Setting 4

Fig. 4. Constrained streaming: fraction of late packets versus the startup delay for a 7000-second video for Settings 2, 3 and 4.

Setting #	# 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.)
1	9	40	40	3.7	50	120	2.2	220	2	30.8
2	5	30	40	3.7	50	120	0.6	195	2	66.5
3	9	40	5	5	100	50	1.5	162	3	46.1
4	5	30	5	5	100	50	1.4	110	3	71.4

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

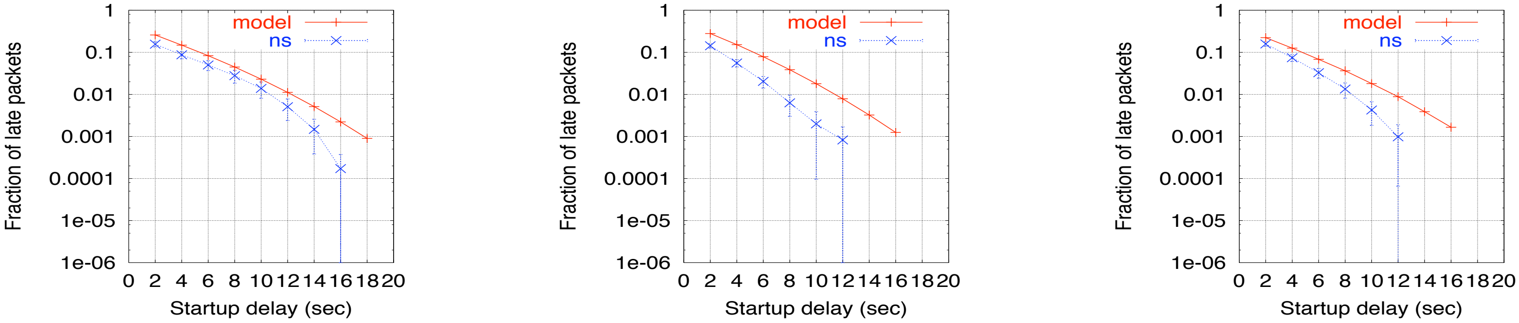
Setting #	Range of loss rate (%)	# of selected runs	μ (pkts ps)	T/μ	required τ (sec)
1	$[2.0, 2.4]$	431	28	1.1	18
2	$[0.4, 0.8]$	535	60	1.1	14
3	$[1.3, 1.7]$	482	42	1.1	14
4	$[1.2, 1.6]$	554	65	1.1	16

TABLE V
UNCONSTRAINED STREAMING: RESULTS FOR VARIOUS SETTINGS FOR MODEL VALIDATION IN *ns*.

stream (including p, R, T_O and the average throughput) are estimated and listed in Table IV. Table V lists the range of loss rate and the number of selected runs in each setting. The playback rate of the video is chosen such that the achievable TCP throughput is 1.1 times of the video playback rate. 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 (Setting 4) 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 average loss rate of the video stream in the 1000 runs is 1.4%. We use $p = 1.4\%$ in the model and select the runs with loss rate between 1.2% to 1.6%. 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 65 packets per second. That is, the available TCP throughput is 10% higher than the video playback rate. Fig. 5 depicts the fraction of late packets versus startup delays. Both the results predicted by the model and measured from the simulation are shown in the figure. Again, we observe a good match between the model and the simulation. 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 (6) 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.

Last, the validation results for other settings (Setting 1, 2 and 3) are plotted in Fig. 6.



(a) Setting 1

(b) Setting 2

(c) Setting 3

Fig. 6. Unconstrained streaming: fraction of late packets versus the startup delay for Settings 1, 2 and 3.

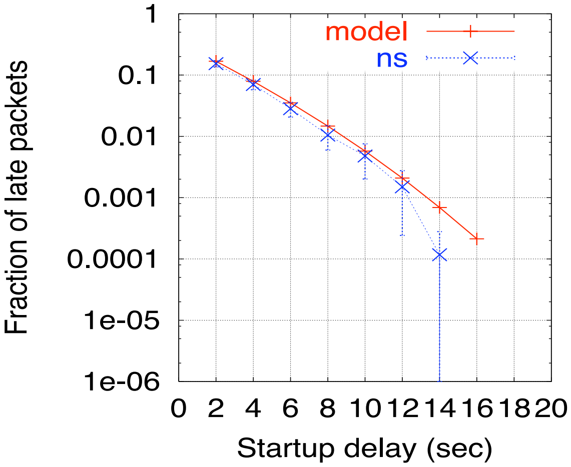


Fig. 5. Unconstrained streaming (Setting 4): fraction of late packets versus the startup delay for a playback rate of 65 packets per second.

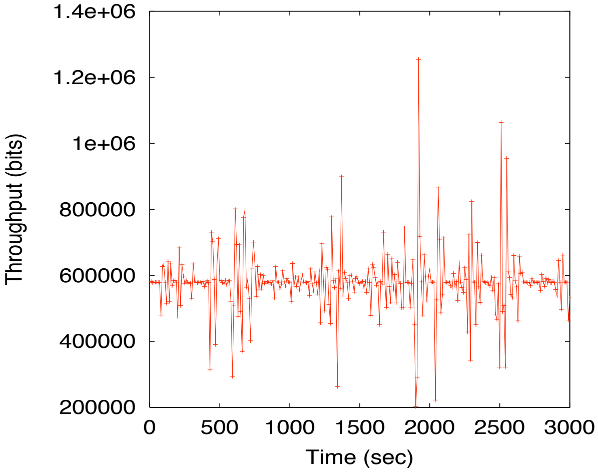


Fig. 7. The TCP throughput of an experiment from USC to the client in the resident house in Amherst, MA.

V. 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* [24] 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.

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) or University of Massachusetts (UMass) 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 22 experiments from February 19 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 throughput, 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 [25]. We use one trace to illustrate our procedure. Fig. 7 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 12 segments from the experiments. The startup delay varies between 4 to 10 seconds. Fig. 8 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 mea-

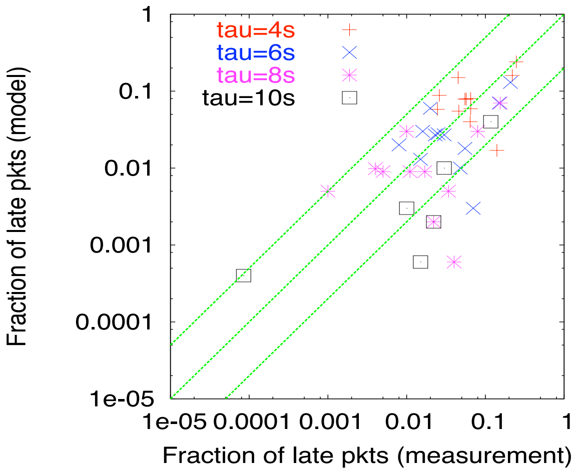


Fig. 8. Constrained streaming: fraction of late packets from the measurements versus that predicted by the model.

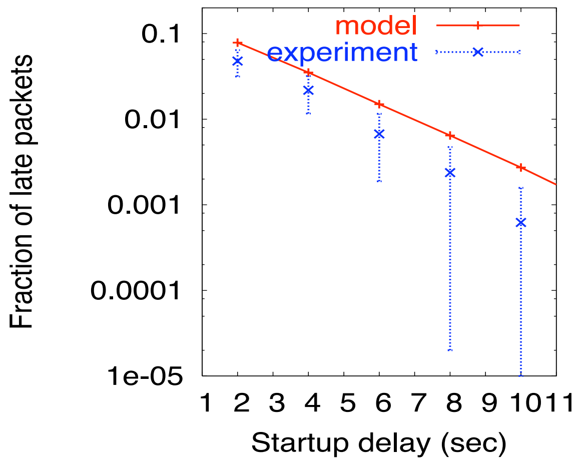


Fig. 9. Unconstrained streaming: The fraction of late packets versus the playback rate of the video for experiments from UMass to Italy for a playback rate of 14 packets per second.

surements and the model. Along the upper and lower 45 degree lines, the fraction of late packets from the model is respectively 5 times higher and lower than that from the measurements. All but 7 scatterplot points fall within the upper and lower 45 degree lines, indicating a match between the model and the Internet experiments. We speculate that the 7 bad matches are due to insufficient number of samples in the data segment.

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). Since the bandwidth for a cable modem connection is too low to benefit from parallel TCP connections, we chose a high-bandwidth university path. 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 $p = 3.1\%$ in the model and select 266 segments having loss rate between 2.7% and 3.5%. For the selected segments, the RTT is 300 ms and $T_O = 1$. The average throughput is 15.2 packets per second. We set the playback rate of the video to be 14 packets per second. Correspondingly, the available TCP throughput is 9% higher than the playback rate of the video. Fig. 9 plots the fraction of late packets for various startup delays. The fraction of late packets predicted by the model is slightly higher than that 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.

VI. Exploring parameter space

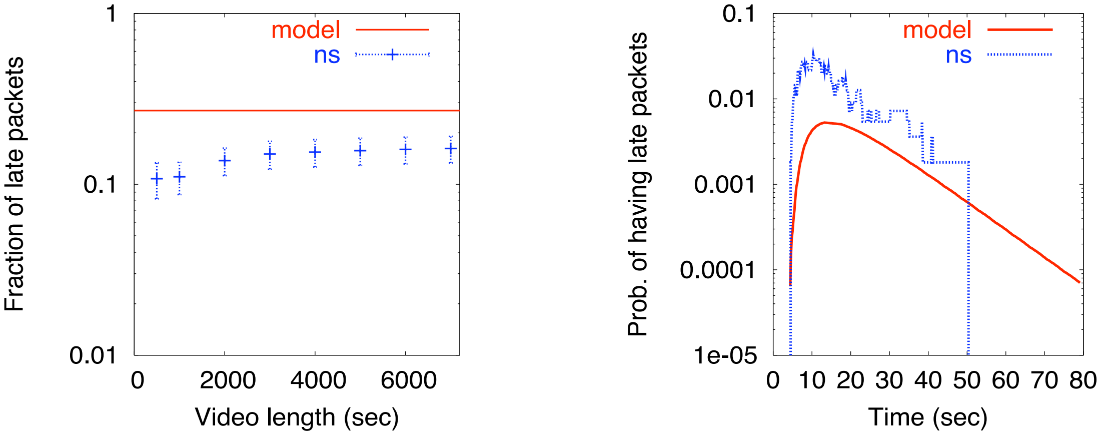
In this section, we vary the model parameters in constrained and unconstrained streaming to study the impacts of these parameters on performance. In doing so, we provide guidelines as to when TCP streaming leads to satisfactory performance.

The loss rate, R and T_O in the model jointly determine the achievable throughput measured in packets. For convenience, we refer to these three parameters as *TCP parameters*. We set the values of the TCP parameters to represent a wide range of scenarios. The loss is varied in the range of 0.4% to 4%. 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 [26]. Consequently, we vary R in the range of 40 ms to 300 ms. We vary T_O from 1 to 4, based on several measurements from Linux machines in [15] and our measurements.

Denote the achievable 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 summarize the key results.

A. *Effect of video length on performance*

We first use the setting in Section IV-A to illustrate the effect of the video length on the performance in constrained streaming. The startup delay is set to 6 seconds and the length of the video ranges from 500 to 7000 seconds. Fig. 10(a) plots



(a) Constrained streaming.

(b) Unconstrained streaming.

Fig. 10. The effect of video length on performance.

the fraction of late packets versus the video length from the model and the *ns* simulation. The model provides a good prediction for various video lengths. 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.

We next use the setting in Section IV-B to investigate how the fraction of late packets varies with the video length in unconstrained streaming. The startup delay is set to 4 seconds. The playback rate is 51 packets per second. Correspondingly, the available TCP throughput is 40% higher than the video playback rate. We obtain P_i $(i = 1, \dots, L)$, the probability that the ith round has at least one late packet (see Section III-C.2). Fig. 10(b) plots P_i over the length of the video from the model and the simulation. In the figure, 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 a 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 achievable throughput is higher than the playback rate of the video. Therefore, the probability of having late packets in a round reaches a peak value and then decreases over time as the number of early packets in the buffer increases.

In Fig. 10(b), the probability of having late packet in the 730th round (i.e., 80th second) decreases to 10^{-4}. 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 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 use videos of 80 seconds for unconstrained streaming.

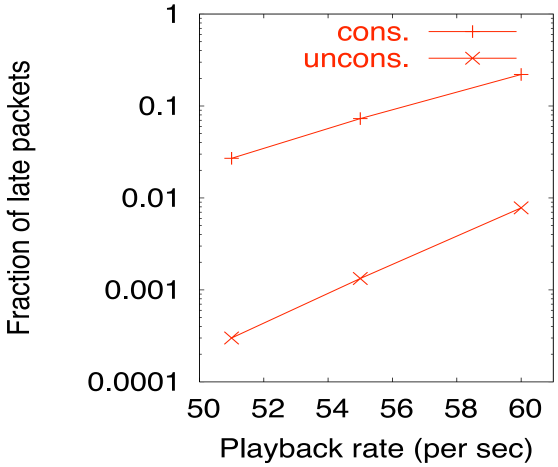


Fig. 11. Performance of constrained and unconstrained streaming when varying the video playback rate and fixing the TCP parameters.

B. Effect of T/μ on 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.

Fig. 11 shows one example, where $p = 1.4\%$, $R = 110$ ms, $T_O = 3$ and the achievable TCP throughput is 71.4 packets

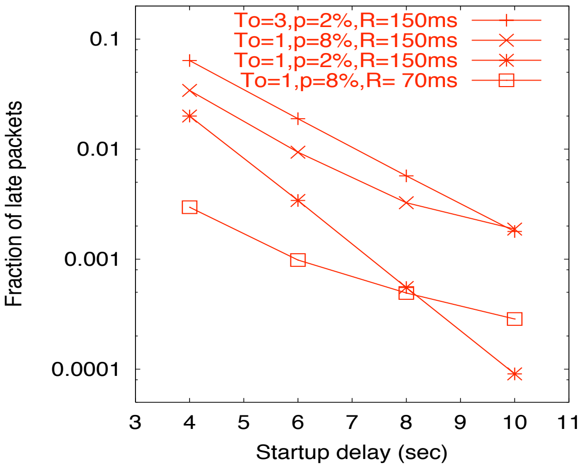


Fig. 12. Constrained streaming: sensitivity to various parameters (p, R, T_O and video playback rate), $T/\mu = 1.6$.

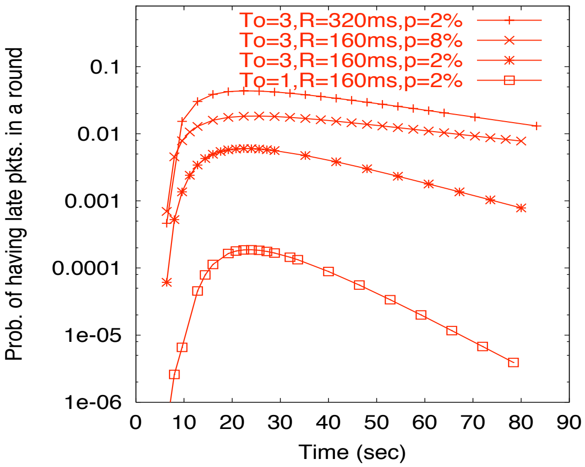


Fig. 13. Unconstrained streaming: sensitivity to the various parameters (p, R, T_O and video playback rate), $T/\mu = 1.3$.

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. The startup delay is 6 seconds. 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 video length 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 parameters*

We next fix T/μ and study the sensitivity of the performance to the various model parameters. Fig. 12 shows the fraction of late packets for 4 sets of TCP parameters in constrained streaming. The playback rate of the video is chosen such that $T/\mu = 1.6$ for all the settings. The startup delay is between 4 and 10 seconds. In Fig. 12, $T_O = 1, 3$; $p = 2\%, 8\%$ and $R = 70, 150$ ms. For $p = 2\%$ 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 $p = 8\%$ 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 8% to 2%. 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. 13 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 6 seconds. In Fig. 13, $p = 2\%, 8\%$; $R = 160, 320$ ms and $T_O = 1, 3$. We observe similar behavior as in constrained streaming: the performance 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 VI-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. In general, viewing quality is satisfactory when the fraction of late packets is low for a short startup delay. People can usually tolerate a few seconds of startup delay. Studies show that the video quality drops when the packet loss rate exceeds 10^{-4} (e.g., [27]).

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

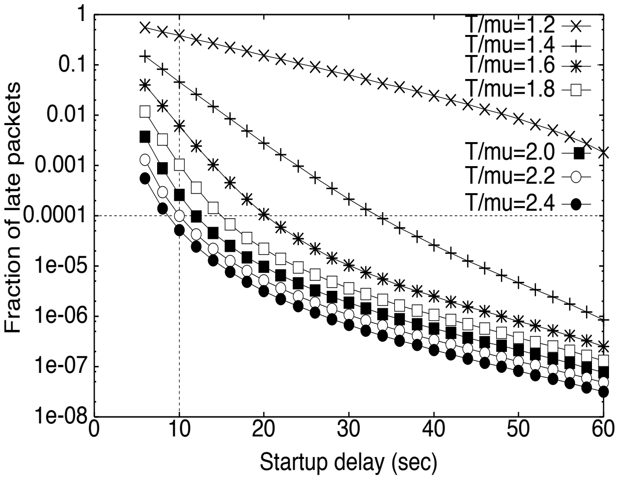


Fig. 14. The fraction of late packets versus the startup delay for $p = 2\%$, $T_O = 4$ and $\mu = 25$ packets per second.

Consequently, we assume that the performance of direct TCP streaming is satisfactory when the fraction of late packets is below 10^{-4} for a startup delay of around 10 seconds.

We choose $p = 0.4\%, 2\%$ or 4%, corresponding to low, medium and high loss rates respectively, and choose $T_O = 1, 2, 3$ or 4. Let T_R denote the achievable TCP throughput in one RTT. Then T_R is determined by p and T_O, and $T = T_R/R$. Since T_R is fixed once p and T_O are fixed, the value of $T/\mu = T_R/(\mu R)$ is varied by varying the product of μ and R. We next explore quantitively the impact of T/μ on the performance of TCP streaming.

1) Constrained streaming: We first fix the playback rate of the video, μ, to be 25 packets per second and vary the value of RTT such that T/μ ranges from 1.2 to 2.4. We observe a diminishing gain by increasing T/μ on the performance: the performance improves dramatically as T/μ increases from 1.2 to 1.6 and less dramatically as T/μ increases from 1.6 to 2.4. One example is shown in Fig. 14, where $p = 2\%$ and $T_O = 4$. This diminishing gain indicates that, to achieve a low fraction of late packets, the required startup delay is very long when T/μ is only slightly higher than 1 and reduces quickly as T/μ increases. However, the reduction becomes less dramatic for large values of T/μ. Fig. 15(a) shows the required startup delay such that the fraction of late packets, f, is below 10^{-4} as a function of T/μ for various loss rates and $2 \leq T_O \leq 4$ (the required startup delay when $T_O = 1$ is much lower for the same loss rate and T/μ). We observe that under various settings, the performance becomes satisfactory when T/μ is roughly 2.

We next set the value of RTT to 50, 100, 200 or 300 ms and vary the playback rate of the video such that T/μ ranges from 1.2 to 2.4. We again observe a dramatic performance gain when T/μ increases from 1.2 to 1.6 and less dramatic gain afterwards. Next, we investigate the required startup delay such that the fraction of late packets is below 10^{-4} when $T/\mu = 2$. Fig. 15(b) shows the required startup delay when $p = 4\%$. The required startup delay for lower loss rates is lower (figures omitted). When $R = 50$ ms (corresponding roughly to two sites on the same coast in the US), the required startup delay is no more than 10 seconds under all settings. When $R = 100$ ms (corresponding roughly to two sites on the two coasts in the US), the required startup delay is no more than 10 seconds under all settings except for very high loss rate ($p = 4\%$) and high T_O values ($T_O \geq 3$). However, for a long RTT, high loss rate and timeout value, the required startup delay is in tens of seconds, as shown in Fig.15(b).

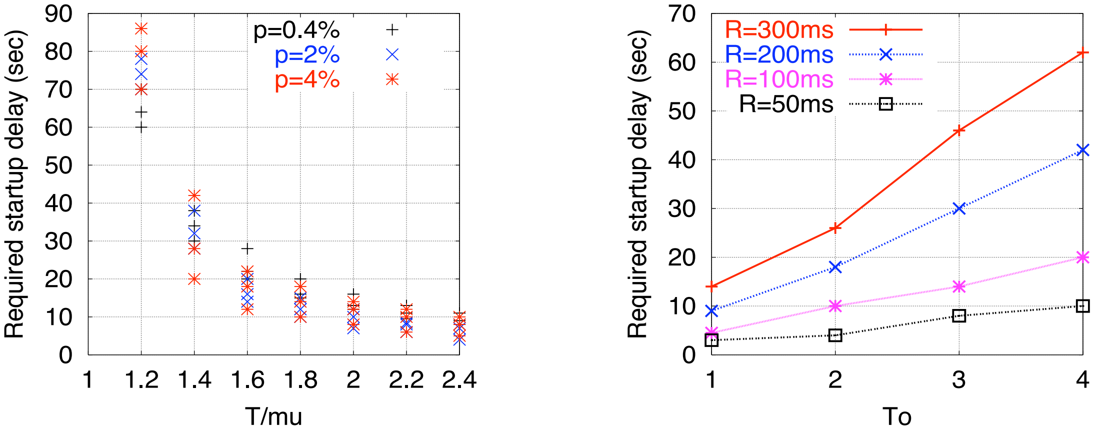
2) Unconstrained streaming: We again first fix the playback rate of the video to 25 packets per second and vary the value of RTT such that T/μ ranges from 1.2 to 2.4. The results are similar as in constrained streaming: diminishing performance gains are observed when increasing T/μ and the performance becomes satisfactory when the achievable TCP throughput is twice the video bitrate. Fig. 16(a) shows the required startup delay such that the fraction of late packets is below 10^{-4} as a function of T/μ for various loss rates and $T_O = 4$ (the required startup delay for lower T_O values is lower). We observe that the required startup delay is bounded within 10 seconds when T/μ increases to 2.

We next set the value of RTT to 50, 100, 200 or 300 ms and vary the playback rate of the video such that T/μ ranges from 1.2 to 2. We obtain the required startup delay such that the fraction of late packets is below 10^{-4} when $T/\mu = 2$. Fig. 16(b) shows the required startup when $p = 4\%$ for various values of RTT. The required startup delay for lower loss rates is lower (figures omitted). Under relatively short RTT, i.e., $R = 50$ or 100 ms, the required startup delay is within 10 seconds for all the settings. However, for a long RTT, high loss rate and timeout value, the required startup delay is in tens of seconds, as shown in Fig.16(b).

E. Summary of results

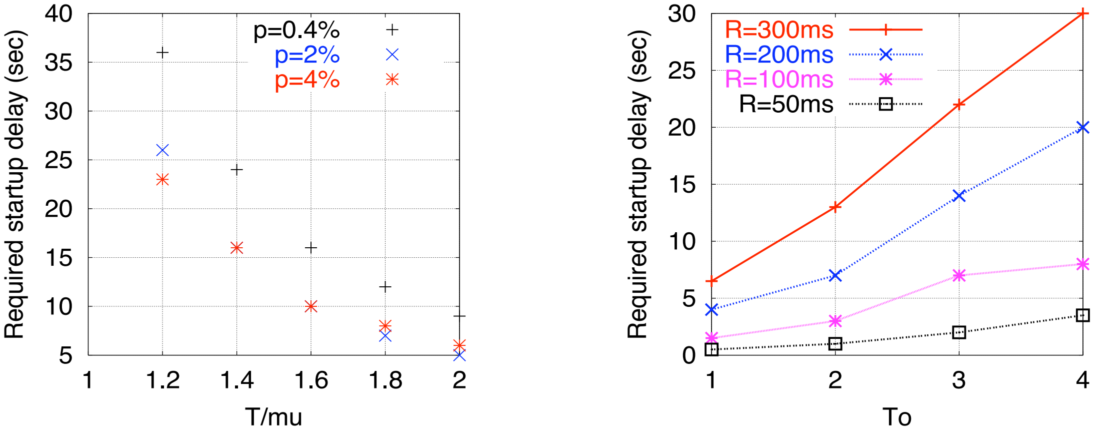
The key results from our exploration of parameter space are:

- The fraction of late packets when the video is beyond a certain length is similar in constrained streaming while it decreases with the video length in unconstrained streaming (after an initial increasing trend at the beginning of the playback).
- The performance of TCP streaming improves as the value of T/μ increases. Furthermore, increasing T/μ beyond a point yields diminishing performance gain.
- The performance of TCP streaming is not solely determined by T/μ but is sensitive to the values of the vari-



(a) (b)

Fig. 15. Constrained streaming: the required startup delay such that $f \leq 10^{-4}$ (a) when $\mu = 25$ packets per second, $2 \leq T_O \leq 4$; (b) when $p = 4\%$ and $T/\mu = 2$.



(a) (b)

Fig. 16. Unconstrained streaming: the required startup delay such that $f \leq 10^{-4}$ (a) when $\mu = 25$ packets per second, $T_O = 4$; (b) when $p = 4\%$ and $T/\mu = 2$.

ous parameters in the models. However, the performance is generally good when the achievable TCP throughput is roughly twice the video bitrate, when allowing a few seconds of startup delay.

- For large RTTs, high loss rates and timeout values, to achieve a low fraction of late packets, either a long startup delay or a large T/μ (greater than 2) is required.

Our study has the following implication. A large fraction of streaming video clips on the Internet today are encoded at bit rates below 300 Kbps [13]. On the other hand, most DSL and cable modem connections support download rates of 750 Kbps - 1.5 Mbps. In the situations where the end-end available bandwidth is only constrained by the last-mile access link, our performance study thus indicates that direct TCP streaming may be adequate for many broadband users.

VII. Conclusions

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 various parameters on the performance of constrained and unconstrained streaming. In doing so, we provided guidelines as to when direct TCP streaming renders satisfactory performance, showing, for example, that TCP generally provides good streaming performance when the achievable TCP throughput is roughly twice the media bitrate, with only a few seconds of startup delay. Note that our model can be easily extended to the setting where loss rate varies during the playout of the video by incorporating the various loss rate values into the Markov models. Last, we use fraction of loss rate as the performance metric throughout the paper. Performance study using more complicated and user-oriented metrics is left as future work.

VIII. Acknowledgment

We thank E. de Souza e Silva and D. R. Figueiredo for many helps with numerically solving Markov models. We also thank L. Golubchik, C. Papadopulous and F. L. Presti for providing us accounts for the Internet experiments. This research was supported in part by the National Science Foundation under NSF grants ANI-9977635, NSF ANI-9980552, CCR-9984030 and EEC-0313747 001.

References

[1] R. Rejaie, M. Handley, and D. Estrin, "Quality adaptation for congestion controlled video playback over the Internet," in *SIGCOMM*, pp. 189–200, September 1999.

[2] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *SIGCOMM 2000*, (Stockholm, Sweden), pp. 43–56, August 2000.

[3] T. P. Nguyen and Z. Avideh, "Distributed video streaming with forward error correction," in *International Packetvideo Workshop*, 2002.

[4] C. Boutremans and J. Y. Le Boudec, "Adaptive joint playout buffer and FEC adjustment for Internet telephony," in *Proceedings of IEEE INFOCOM'2003*, (San-Francisco, CA), April 2003.

[5] J. van der Merwe, S. Sen, and C. Kalmanek, "Streaming video traffic: Characterization and network impact," in *Proceedings of the Seventh International Web Content Caching and Distribution Workshop*, August 2002.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] 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.

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

[13] M. Li, M. Claypool, R. Kinicki, and J. Nichols, "Characteristics of streaming media stored on the internet," Tech. Rep. WPI-CS-TR-03-18, CS Department, Worcester Polytechnic Institute, May 2003.

[14] M. Mathis, J. Semke, and J. Mahdavi, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communications Review*, vol. 27, no. 3, 1997.

[15] 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.

[16] V. Misra, W. Gong, and D. Towsley, "Stochastic differential equation modeling and analysis of TCP-windowsize behavior," in *In Proceedings of PERFORMANCE99*, (Istanbul, Turkey), 1999.

[17] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of TCP/IP with stationary random losses," in *SIGCOMM*, pp. 231–242, 2000.

[18] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *INFOCOM (3)*, pp. 1742–1751, 2000.

[19] M. Mellia, I. Stoica, and H. Zhang, "TCP model for short lived flows," *IEEE Communication Letters*, vol. 6, February 2002.

[20] S. Bohacek, "A stochastic model of TCP and fair video transmission," in *Proc. IEEE INFOCOM*, 2003.

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

[22] 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.

[23] 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.

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

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

[26] 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.

[27] O. Verscheure, P. Frossard, and M. Hamdi, "MPEG-2 video services over packet networks: Joint effect of encoding rate and data loss on user-oriented QoS," in *Proc. of NOSSDAV*, July 1998.

[28] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource allocation for multimedia streaming over the internet," *IEEE Transactions on Multimedia*, September 2001.

[29] J. D. Esary, F. Proschan, and D. W. Walkup, "Association of random variables, with applications," *Annals of mathematical statistics*, vol. 38, pp. 1446–1474, October 1967.

Appendix I
Model for constrained streaming

In constrained streaming, the state of the model in the ith round, Y_i^c, is 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. As in [10], [11], X_i, is represented as a tuple: $X_i = (W_i, C_i, L_i, E_i, R_i)$. W_i is the window size for round i. C_i models the delayed ACK behavior of TCP; $C_i = 0$ and $C_i = 1$ indicate that 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 back-off exponent in round i. R_i indicates if a packet being sent in the timeout phase is either a retransmission ($R_i = 1$) or a new packet ($R_i = 0$).

Let $p_{w,c,l,e,r,n;w',c',l',e',r',n'} = P(W_{i+1} = w', C_{i+1} = c', L_{i+1} = l', E_{i+1} = e', R_{i+1} = r', N_{i+1} = n' \mid W_i = w, C_i = c, L_i = l, E_i = e, R_i = r, N_i = n)$ be the probability associated with the state transition from state $(W_i = w, C_i = c, L_i = l, E_i = e, R_i = r, N_i = n)$ to state $(W_{i+1} = w', C_{i+1} = c', L_{i+1} = l', E_{i+1} = e', R_{i+1} = r', N_{i+1} = n')$. Let $r_{w,c,l,e,r,n;w',c',l',e',r',n'}$ be the time taken for this state transition. Let R_{TO} be the value of the first retransmission timer. It is rounded as a multiple of the RTT R. Denote the maximum window size as W_{max}. The state transition probabilities and the times taken for the transitions are listed in Table VI. In the table, there are 5 groups of p's and r's corresponding respectively to situations (1) no packets are lost in a round; (2) one or more packets are lost in a round; (3) one or more packets are lost in a short round; (4) exponential back-off; (5) packet playback.

Appendix II
Model for unconstrained streaming

In unconstrained streaming, the state of the model in the ith round $Y_i^u = X_i$, where $X_i = (W_i, C_i, L_i, E_i, R_i)$. The im-

$p_{w,0,0,0,0,n;w,1,0,0,n'}$	=	$(1-p)^w$,	$1 \leq w \leq W_{max}$ $n' = \min(N_{max}, n+w)$
$p_{w,1,0,0,0,n;w+1,0,0,0,n'}$	=	$(1-p)^w$,	$1 \leq w \leq W_{max}$, $n' = \min(N_{max}, n+w)$
$p_{w,1,0,0,0,n;w,0,0,0,n'}$	=	$(1-p)^w$,	$w = W_{max}$, $n' = \min(N_{max}, n+w)$
$r_{w,0,0,0,0,n;w,1,0,0,n'}$	=	R,	$1 \leq w \leq W_{max}$
$r_{w,1,0,0,0,n;w+1,0,0,0,n'}$	=	R,	$1 \leq w \leq W_{max}$
$r_{w,1,0,0,0,n;w,0,0,0,n'}$	=	R,	$1 \leq w \leq W_{max}$
$p_{w,c,0,0,0,n;w-l,0,l,0,0,n'}$	=	$p(1-p)^{w-l}$,	$2 \leq w \leq W_{max}, c = 0, 1, 1 \leq l \leq w$, $n' = \min(N_{max}, n+w-l)$
$p_{w,c,0,0,0,n;1,0,0,1,1,n}$	=	p,	$2 \leq w \leq W_{max}, c = 0, 1$
$r_{w,c,0,0,0,n;w-l,0,l,0,0,n'}$	=	R,	$2 \leq w \leq W_{max}, c = 0, 1, 1 \leq l \leq w$,
$r_{w,c,0,0,0,n;1,0,0,1,1,n}$	=	R_{TO},	$2 \leq w \leq W_{max}, c = 0, 1$
$p_{1,0,l,0,0,n;1,0,0,1,1,n'}$	=	1,	$n' = \min(N_{max}, n+1-p)$
$p_{2,0,l,0,0,n;1,0,0,1,1,n'}$	=	1,	$n' = \min(N_{max}, n+p(1-p)+2(1-p)^2)$
$p_{w,0,l,0,0,n;1,0,0,1,1,n'}$	=	$\sum_{i=1}^{2} p(1-p)^i$,	$3 \leq w \leq W_{max}$ $n' = \min(N_{max}, n + \frac{\sum_{i=0}^{2} ip(1-p)^i}{\sum_{i=0}^{2} p(1-p)^i})$
$p_{w,0,l,0,0;\lfloor (w+l)/2 \rfloor,0,0,0,0}$	=	$\sum_{i=3}^{w-1} p(1-p)^i + (1-p)^w$,	$3 \leq w \leq W_{max}$, $n' = \min(N_{max}, n + \frac{\sum_{i=3}^{w-1} ip(1-p)^i + w(1-p)^w}{\sum_{i=3}^{w-1} p(1-p)^i + (1-p)^w})$
$r_{w,0,l,0,0,n;1,0,0,1,1,n'}$	=	$R_{TO} - R$,	$1 \leq w \leq W_{max}$
$r_{w,0,l,0,0;\lfloor (w+l)/2 \rfloor,0,0,0,0}$	=	R,	$3 \leq w \leq W_{max}$
$p_{1,0,0,i,r,n;1,0,0,\min\{i+1,7\},1,n}$	=	p,	$1 \leq i \leq 7, r = 0, 1$
$p_{1,0,0,i,1,n;1,0,0,i,0,n+1}$	=	$1-p$,	$1 \leq i \leq 7$
$p_{1,0,0,i,0,n;2,0,0,0,0,n+1}$	=	$1-p$,	$1 \leq i \leq 7$
$r_{1,0,0,i,r,n;1,0,0,\min\{i+1,7\},1,n}$	=	$2^{(i-1)} R_{TO}$,	$1 \leq i \leq 7, r = 0, 1$
$r_{1,0,0,i,1,n;1,0,0,i,0,n+1}$	=	R,	$1 \leq i \leq 7$
$r_{1,0,0,i,0,n;2,0,0,0,0,n+1}$	=	R,	$1 \leq i \leq 7$
$r_{w,c,l,e,r,n;w,c,l,e,r,n'}$	=	R,	$n' = n - \mu R$

TABLE VI

CONSTRAINED STREAMING: DEFINITION OF THE STATE TRANSITION PROBABILITIES AND THE TIMES TAKEN FOR THE TRANSITIONS.

pulse reward $\rho_{w,c,l,e,r;w',c',l',e',r'}$ is associated with a transition from state $(W_i = w, C_i = c, L_i = l, E_i = e, R_i = r)$ to state $(W_{i+1} = w', C_{i+1} = c', L_{i+1} = l', E_{i+1} = e', R_{i+1} = r')$. This impulse reward is defined in Table VII. In the table, there are 4 groups of ρ's corresponding respectively to situations (1) no packets are lost in a round; (2) one or more packets are lost in a round; (3) one or more packets are lost in a short round; (4) exponential back-off.

APPENDIX III
DERIVATION OF AN UPPER BOUND ON β FOR UNCONSTRAINED STREAMING

Let β be the probability that at least one late packet occurs during the playback of the video. Let α be the probability of having no late packet throughout the playback of the video. Then $\beta = 1 - \alpha$. We provide an upper bound on β by giving a lower bound on α. Let A_i be the total number of packets reaching the client up to the ith round. Let B_i be the total number of packets played back by the client up to the ith round. Then A_i and B_i are respectively the discrete-time version of $A(t)$ and

$\rho_{w,0,0,0,0;w,1,0,0}$	=	$w - \mu R,$	$1 \leq w \leq W_{max}$
$\rho_{w,1,0,0,0;w+1,0,0,0}$	=	$w - \mu R,$	$1 \leq w \leq W_{max}$
$\rho_{w,1,0,0,0;w,0,0,0}$	=	$w - \mu R,$	$1 \leq w \leq W_{max}$
$\rho_{w,c,0,0,0;w-l,0,l,0,0}$	=	$w - l - \mu R,$	$2 \leq w \leq W_{max}, c = 0, 1, 1 \leq l \leq w,$
$\rho_{w,c,0,0,0;1,0,0,1,1}$	=	$-\mu R,$	$2 \leq w \leq W_{max}, c = 0, 1$
$\rho_{1,0,l,0,0;1,0,0,1,1}$	=	$1 - p - \mu R_{TO},$	
$\rho_{2,0,l,0,0;1,0,0,1,1}$	=	$p(1-p) + 2(1-p)^2 - \mu R_{TO},$	
$\rho_{w,0,l,0,0;1,0,0,1,1}$	=	$\frac{\sum_{i=0}^{2} ip(1-p)^i}{\sum_{i=0}^{2} p(1-p)^i} - \mu R_{TO},$	$1 \leq w \leq W_{max}$
$\rho_{w,0,l,0,0;\lfloor (w+l)/2 \rfloor,0,0,0,0}$	=	$\frac{\sum_{i=3}^{w-1} ip(1-p)^i + w(1-p)^w}{\sum_{i=3}^{w-1} p(1-p)^i + (1-p)^w} - \mu R,$	$3 \leq w \leq W_{max}$
$\rho_{1,0,0,i,r;1,0,0,\min\{i+1,7\},1}$	=	$-\mu 2^{(i-1)} R_{TO},$	$1 \leq i \leq 7, r = 0, 1$
$\rho_{1,0,0,i,1;1,0,0,i,0}$	=	$1 - \mu R,$	$1 \leq i \leq 7$
$\rho_{1,0,0,i,0;2,0,0,0,0}$	=	$-\mu R,$	$1 \leq i \leq 7$

TABLE VII

UNCONSTRAINED STREAMING: DEFINITION OF IMPULSE REWARD.

$B(t)$ introduced in Section III-A. It is clear that $N_i = A_i - B_i$ and

$$B_i = \begin{cases} \mu(iR - \tau), & \text{if } iR \geq \tau \\ 0 & \text{o.w.} \end{cases}$$

Let S_k be the number of packets sent out successfully by TCP in the kth round. Then $A_i = \sum_{k=1}^{i} S_k$. Since the time unit in the model is the length of a round, we have

$$\begin{aligned} \alpha &= P(N_1 \geq 0, N_2 \geq 0, \ldots, N_L \geq 0) \\ &= P(A_1 \geq B_1, A_2 \geq B_2, \cdots, A_L \geq B_L) \end{aligned}$$

It is clear that A_i is a nondecreasing function of S_i. If S_i's are associated, then by [29], we have

$$\alpha \geq \Pi_{i=1}^{L} P(A_i \geq B_i) = \Pi_{i=1}^{L}(1 - P_i)$$

where $P_i = P(N_i < 0) = P(A_i < B_i)$. We next sketch a proof that S_i's are associated by only considering the congestion control behavior in TCP. The proof when also considering time out behavior is similar. We first define random variables χ_i as

$$\chi_i = \begin{cases} 1, & \text{congestion occurs in round } i \\ 0, & \text{o.w.} \end{cases}$$

Then by the mechanism of TCP, we have

$$S_i = \max(S_{i-1} + 1, W_{max})(1 - \chi_i) + S_{i-1}\chi_i/2$$

It is observed that packet losses in TCP follow a Poisson process [16]. We therefore assume χ_i's are independent. Then by Property P_4 in [29], S_i's are associated.

We obtain an upper bound on β from the lower bound on α as

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