Predictive Buffering for Multi-Source Video Streaming over the Internet

P. Y. Ho and Jack Y. B. Lee
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong
{pyho5, yblee}@ie.cuhk.edu.hk

***Abstract*—The current best-effort Internet does not guarantee the bandwidth availability between a receiver and a sender, and so renders any quality-of-service (QoS) control difficult, if not impossible. This paper proposes a novel predictive buffering algorithm for streaming video not from one, but from multiple senders to a receiver over the best-effort Internet. In particular, the predictive buffering algorithm estimates the mean and variance of the aggregate throughput of multiple senders, and then use these estimated parameters to predict the future bandwidth availability. By appealing to the Central Limit Theorem, the future bandwidth availability will tend to be normally distributed, irrespective of the distribution of the measurement bandwidth availability. This insight enables the buffering algorithm to determine, at runtime, the minimum buffering time required to ensure playback continuity. Extensive trace-driven simulations show that this predictive buffering algorithm can achieve buffer delays that are remarkably close to the optimal buffer time.**

I. INTRODUCTION

The current Internet does not provide any end-to-end quality-of-service (QoS) control and thus presents a significant challenge to bandwidth-sensitive applications such as streaming video and TV contents over the Internet. The fluctuations in bandwidth availability can easily lead to frequent video playback interruptions that are extremely annoying to the end users.

To tackle this challenge researchers have developed novel adaptation mechanisms [1-4] to dynamically adjust the video bit-rate to match the varying bandwidth availability. However, this often requires the use of special compression algorithms (e.g., FGS [1-2]) or real-time media transcoders [4] that may not be feasible or available in some applications.

Without these advanced codecs or transcoders, today's content providers typically prepare a few versions of the same content in different bit-rates to cater for users of different connection bandwidth. Given the complexity and the time required to encode multiple versions of the same video, it is not surprising that there will only be a small number of versions of the same video content provided. Thus the selected video is often either of too low or too high a bit-rate for the client. The former case is trivial as streaming will likely be successful. The latter case will be far more complicated as the client now does not have sufficient bandwidth to streaming the video in the conventional manner. Some existing video players will simply download the video and begins playback only after substantial portion of the video has been downloaded. However, due to the inherent variations in network bandwidth availability, even this conservative strategy may not be able to ensure continuous playback, especially for long video contents.

This work tackles this problem by developing a novel predictive buffering algorithm that can determine at runtime the buffering time required to ensure playback continuity, especially for longer videos (e.g., over a few minutes) and when the video bit-rate exceeds the available network bandwidth. The proposed predictive buffering algorithm is designed around two principles.

First, during the initial buffering period the client can measure the mean and variance of the available bandwidth over a given interval (e.g., 1 s). Assuming that the past and future available bandwidth is a stationary random process of unknown distribution, then the sum of future bandwidth availability over the next n intervals will approach normal as $n \to \infty$ due to the Central Limit Theorem. Thus knowing the distribution of the future bandwidth, the client can determine the minimum buffer time to ensure playback continuity.

Second, the stationarity assumed of the future bandwidth availability obviously may not be true in practice, as the available bandwidth from a sender to a receiver is simply unpredictable. However, our investigation reveals that if there are multiple senders transmitting data to the client simultaneously, then the aggregate available bandwidth will become far more stationary. Therefore, by employing sufficient number of senders, each transmitting a portion of the data, the stationarity assumption can then be satisfied and we can invoke the first principle to determine the minimum buffering time accordingly.

The proposed predictive buffering algorithm is evaluated using extensive trace-driven simulations, with two sets of traffic traces obtained from different networks and different time frames. The results confirm the relation between the aggregate bandwidth stationarity and the number of senders in the aggregate data flow, and also show that the predictive buffering algorithm can achieve buffer delays that are remarkably close to the optimal buffer time.

The rest of the paper is organized as follows: Section II reviews some previous related work; Section III presents the

details of the predictive buffering algorithm; Section IV discusses the practical issues of our algorithm; Section V investigates the bandwidth model and the predictive buffering algorithm using trace-driven simulations; and Section VI summarizes the paper and outlines some future work.

II. Background and Related Work

Streaming video from multiple sources to a receiver has previously been investigated by a number of researchers [5-10]. Compared to single-source streaming, multi-source streaming has several potential advantages, such as increasing the throughput by combining the bandwidth of multiple senders [5-7]; adapting to network bandwidth variations by shifting the workload among the multiple senders [8-9]; and reducing bursty packet loss by splitting the data transmission among the multiple senders [5-6].

For example, Nguyen and Zakhor [5-6] developed rate allocation and packet partition algorithms with Forward Error Correction (FEC) to minimize the packet loss rate and the probability of late packet arrivals. Xu *et al*. [7] proposed an algorithm for media data assignment to reduce buffering delay. Kwon and Yeom [8] proposed a dynamic rate allocation and packet partition scheme to adapt to the senders' varying throughput. Agarwal and Rejaie [9] proposed an adaptive layered streaming algorithm to compensate for variations in the measured available bandwidth from all congestion controlled senders.

The above studies exploited the availability of multiple sources and the diversity of multiple network paths to improve streaming performance. In another study, Reisslein and Ross proposed a novel call admission scheme [12] that can provide statistical QoS guarantee in streaming prerecorded variable-bit-rate (VBR) videos over ATM. In their study the network bandwidth is known but the video bit-rate can vary due to the VBR encoding and interactive playback controls. To guarantee QoS they proposed to multiplex multiple video streams over the network and then model the bit-rate of the multiplexed aggregate video flow as a stochastic process, and then apply the Central Limit Theorem and Large Deviation theory to obtain probabilistic bounds.

In comparison, the intra-flow bandwidth aggregation model developed in this paper also appeals to the Central Limit Theorem (CLT) to obtain probabilistic bounds. However, there are two fundamental differences. First, Reisslein and Ross's work [12] solved the problem of varying video bit-rate but with constant network bandwidth, while our work solved the problem of constant video bit-rate but with varying network bandwidth. Second, the varying video bit-rate in Reisslein and Ross's work, although modeled as a random process, is known *a priori* as they are prerecorded. By contrast, our work does not assume *a priori* knowledge of the varying available bandwidth, and thus we need to develop an estimation algorithm to measure and estimate the parameters of the stochastic process.

In another related study, Hui and Lee [10-11] proposed to model the aggregate available bandwidth of multiple independent senders as a normal distribution by appealing to the CLT. Based on this model they developed download [10] as well as adaptive streaming algorithms [11] to provide probabilistic QoS guarantee in streaming video over best-effort networks.

Comparing to this work, Hui and Lee's study drew on the observation that if the senders are independent, then the sum of their available bandwidth *at any given time* will tend to be normally distributed when there are sufficient number of senders. By contrast, we argued in this work that even if the sum of the available bandwidth across multiple senders is not normally distributed, the sum of the future available bandwidth *over a period of time* will still be normally distributed according to the CLT, provided that the aggregate available bandwidth is sufficiently independent temporally and is stationary. For clarity, we refer to the model in Hui and Lee's work as the inter-flow bandwidth aggregation model (i.e., sum over multiple senders) and refer to the model proposed in this work the intra-flow bandwidth aggregation model (i.e., sum over time).

III. Predictive Buffering Algorithm

In the following we first formulate the system model and then present the predictive buffering algorithm. Note that we do not consider the issue of data assignment (i.e., how to split data across the senders) in this study and refer the readers to the related work (e.g., [5] and [7]).

A. System Model

To begin a new video session, a client will send requests to n senders to initiate data transfer. We assume that the video data are delivered from each sender to the receiver using a transport protocol with congestion control mechanisms such as TCP or TCP-friendly streaming protocols (e.g., TFRC [13]) such that the bandwidth available to the video session will vary according to the instantaneous load of the network path.

The client upon receiving the initial video data will begin the buffering period, and then start playback once sufficient amount of video data are buffered. Specifically, let C_i be the total amount of data received from all n senders in time interval i after the buffering process begins; R be the video bit-rate and w be the time to start playback. To ensure continuous playback we must ensure that the amount of data received at any time must not be less than the amount of data consumed, i.e.,

$$\sum_{j=1}^{i} C_j \geq R(i-w), \forall i > w \qquad (1)$$

or else buffer underflow will occur, causing playback interruptions. The challenge is to find, *at run time*, the smallest buffering period w that satisfies (1).

B. Predictive Buffering Algorithm

At each time interval, the client will check to see if sufficient data have been received to sustain continuous video playback for the rest of the video session. Let L be the total video length in number of time intervals and B_i be the amount of data received up to the time interval i. Then the client can guarantee continuous playback for the entire video session if the

following constraint is satisfied:

$$B_i + \sum_{j=i+1}^{i+k} C_j \geq Rk, \forall k = 1,2,\ldots,L \qquad (2)$$

where the L.H.S. is the amount of data already buffered plus the amount of data to be received in the future k time intervals, and the R.H.S. is the amount of data to be consumed in the future k time intervals if playback is to begin from time interval i.

Otherwise the client will buffer for another time interval and then check (2) again, and repeat the process until (2) is satisfied. However the precise future bandwidth availabilities $\{C_j \mid j \geq i+1\}$ are obviously not known *a priori* and so we need to devise a way to estimate it.

Specifically, let the aggregate bandwidth $\{C_i$'s$\}$ be independent and *arbitrarily distributed*. The only assumption needed is that the future available bandwidth $\{C_j \mid j=i+1,i+2,\ldots\}$ maintains the same mean and variance as the past available bandwidth up to the current time interval i: $\{C_j \mid j=1,2,\ldots,i\}$. In other words, the client assumes that the aggregate available bandwidth of the n senders is stationary with respect to their mean and variance. As the $\{C_i$'s$\}$ are independent, the CDF of the summation term in the L.H.S. of (2) will be equal to the convolution of the CDFs of the k aggregate bandwidths $\{C_j \mid j=i+1,i+2,\ldots,i+k\}$, denoted by $F_k(\cdot)$. Now as the $\{C_i$'s$\}$ are independent with the same mean μ and variance σ^2 the CDF $F_k(\cdot)$ will approach normal with mean $k\mu$ and variance $k\sigma^2$ as $k \to \infty$ according to the Central Limit Theorem – intra-flow bandwidth aggregation.

Thus the minimum buffering time needed to guarantee playback continuity with a given probability of Δ can be computed from

$$w = \min_i \{F_k(Rk - B_i) < (1-\Delta), \forall k = 1,2,...,L\} \qquad (3)$$

where the mean and variance of $F_k(\cdot)$ are estimated using the measured mean and variance of the aggregate bandwidth $\{C_i$'s$\}$ during the initial buffering period.

IV. Practical Issues

There are two subtle issues in the predictive buffering algorithm presented in Section III. The first one is related to the two assumptions on the underlying stochastic process governing the aggregate available bandwidth and the second one is related to the estimation of the mean and variance of the aggregate bandwidth $\{C_i$'s$\}$.

A. Stationarity and Independence

In computing the CDF $F_k(\cdot)$ in (3) it was assumed that the k aggregate bandwidth $\{C_i$'s$\}$ are independently and arbitrarily distributed, but with the same mean μ and variance σ^2. Nevertheless, it is clearly very difficult, if not impossible, to predict the means and variances of future available bandwidth for individual senders, and so the key is to combine multiple independent senders such that the non-stationarity of individual senders will be partially cancelled out, thus leading to a more stationary stochastic process (with respect to the aggregate mean and variance) than that of a single sender.

Specifically, consider n senders with the available bandwidth of sender j in time interval i denoted by $c_{i,j}$. Then with a buffering period of w time intervals, the measured mean bandwidth of sender j in the intervals $[1,w]$ will be equal to

$$\mu_j(1,w) = E[c_{i,j} \mid i = 1,...,w] \qquad (4)$$

The future bandwidths, say from time intervals w+1 to 2w, obviously may vary randomly and even the mean may also vary. In particular, if the stochastic process is non-stationary, e.g., with a decreasing mean bandwidth $\mu_j((x+1)w,(x+2)w) < \mu_j(xw,(x+1)w)$, then the mean future bandwidth will drift further and further away from the initial measured mean, thus violating the stationarity assumption.

However, if there are multiple independent senders, their deviations in the future mean bandwidth will also be independent. Consequently some of the senders will exhibit increases in their mean bandwidth and the others decreases in their mean bandwidth, thereby partially cancelling out the deviations. The case for variance is similar and thus is not repeated here.

In addition to stationarity, we also assumed that the $\{C_i$'s$\}$ are independent in devising the predictive buffering algorithm. Our investigation of real-world traffic traces show that the aggregate available bandwidth does exhibit some temporal correlations within a short time scale but the correlations diminish rapidly over longer time scales. We will return to the stationarity and independence issues in Section V and investigate their properties using trace-driven simulations.

B. Parameter Estimation

During the initial buffering period the client measures the mean and variance of the aggregate available bandwidth. Being measurements of a stochastic process the measurement accuracy will depend on the number of samples used, i.e., the length of the measurement period. This latter point leads to another subtle issue as the length of the measurement period is simply equal to the buffering period, which can vary significantly depending on the ratio of the video bit-rate to the mean aggregate available bandwidth as well as the variances of the available bandwidth.

For example, if the available bandwidth is substantially lower than the video bit-rate then the buffering period will likely be longer, thus allowing more accurate measurement of the required parameters. On the other hand, if the available bandwidth is comparable to the video bit-rate then the buffering period as computed from (3) can be very short. In this case if the measured parameters are inaccurate then the computation of (3) will become inaccurate as well, possibly resulting in playback interruptions.

To guard against this problem, we employ the method of confidence interval [14] in estimating the mean and variance during the buffering period. Specifically, when the sample size w is more than 30, we can assume that the sample mean distribution of μ is normally distributed. The $(1-\alpha)$ confidence interval of sample mean is given by

$$(\mu - z_{\alpha/2}\frac{\sigma}{\sqrt{w}}, \mu + z_{\alpha/2}\frac{\sigma}{\sqrt{w}}) \quad (5)$$

where σ is the samples' standard deviation and $z_{\alpha/2}$ is equal to 2.5 for $\alpha = 0.1$ (i.e., 99% confidence). Thus the client can use the lower limit of the confidence interval as the sample mean.

In extreme cases with sample size $w < 30$, the sample mean distribution is replaced by the Student's t-distribution with the corresponding $(1-\alpha)$ confidence interval given by

$$(\mu - t_{\alpha/2,w-1}\frac{\sigma}{\sqrt{w}}, \mu + t_{\alpha/2,w-1}\frac{\sigma}{\sqrt{w}}) \quad (6)$$

where the value of $t_{\alpha/2,w-1}$ is given in the t-table.

Our results show that using this confidence interval method can effectively prevent inaccurate parameter estimations without significantly increasing the estimated buffering time.

V. Performance Evaluation

In this section we evaluate the performance of the predictive buffering algorithm using trace-driven simulations. There are two sets of trace data in the simulations.

The first set is obtained from the NLANR PMA archive [15], which captured the packet-level trace data at an Internet gateway at Bell Labs in 2002. We divide the one-day trace into separate one-hour sub-traces and use them as cross traffic in the simulation topology depicted in Fig. 1.

The simulator is implemented using NS2 [16], with up to N senders {S1, S2, …, SN} transmitting data simultaneously to the receiver R using TCP as the transport. The senders do not perform additional rate control and simply transmit data as fast as TCP allows. We choose TCP for its ability to automatically adapt to the network load (i.e., the cross traffic) to obtain a fair share of the available bandwidth for transporting video data. Other transport protocols such as TFRC [13] can also be used as long as they have built-in congestion control algorithm. The predictive buffering algorithm operates independently from the actual transport protocol used.

The second set of trace data is obtained from our measurements conducted in the PlanetLab [17]. Specifically, a test program was installed to the PlanetLab nodes (total 286 nodes) for the measurements. In each measurement, N+1 nodes are randomly drawn from the pool of PlanetLab nodes, with one node acting as the receiver and the remaining N nodes acting as senders. All N senders then simultaneously transmit data to the receiver using TCP. Note that there is also a pre-measurement test designed to identify and eliminate sending nodes with excessively high bandwidth to prevent causing performance degradations to other users sharing the same PlanetLab nodes. After the pre-measurement test each measurement would last for 2 hours.

A. Bandwidth Stationarity

As discussed in Section IV-A the number of senders in the aggregate data flow is expected to exert a significant impact on the stationarity of the aggregate available bandwidth. To test this intuition we ran trace-driven simulations using the two trace data sets and then study how far the mean available bandwidth will deviate from the initial estimations.

Specifically, we consider periods of duration of w time intervals, i.e., $[1,w]$, $[w+1,2w]$, …, with the mean bandwidth of the initial period, i.e.,

$$\mu(1,w) = \sum_{j=1}^{N} \mu_j(1,w) \quad (7)$$

as the estimated mean of the aggregate available bandwidth of the data flow. We then compute the mean bandwidth of the subsequent periods to see how far they have deviated from the estimated mean:

$$D_{avg} = v^{-1}\mu(1,w)^{-1}\sum_{i=1}^{v}\left|\mu\left(iw+1,(i+1)w\right) - \mu\left(1,w\right)\right| \quad (8)$$

where v is the total number of w-time-interval periods. Similar calculations are also performed for the standard deviation of the aggregate available bandwidth.

If the aggregate available bandwidth is a stationary stochastic process we would expect D_{avg} to be small, and vice versa. Fig. 2 plots the values of D_{avg} with w=100 for 1 to 8 senders. For both sets of trace data we can clearly observe the consistent decrease in D_{avg} when the number of senders is increased from 1 to 8. This suggests that by aggregating the bandwidth of more senders, the resultant combined data flow does exhibit higher level of stationarity. The results also show that the PlanetLab traces are generally more stationary then that of the NLANR PMA traces.

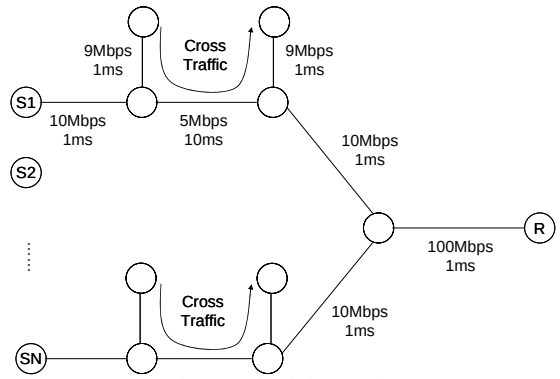

Figure 1. Simulation topology

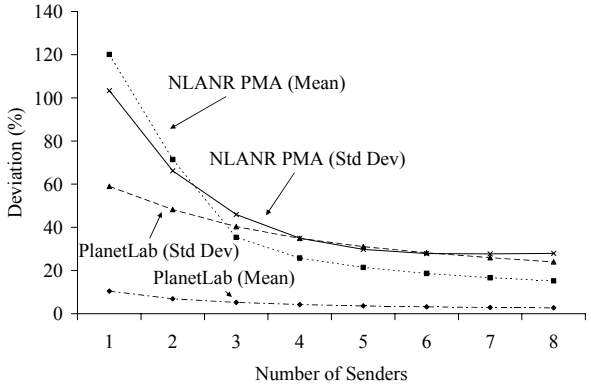

Figure 2. Deviation of mean and standard deviation from initial estimations

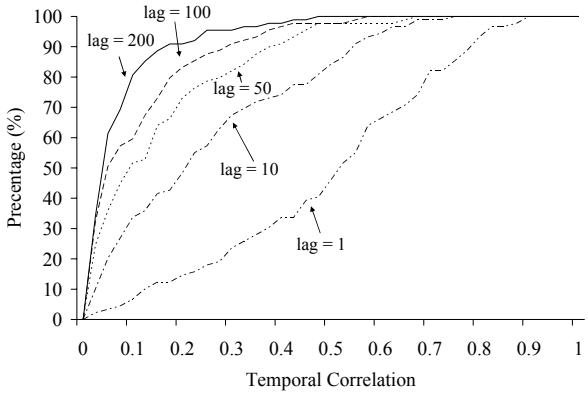

Figure 3. The CDF of temporal correlation with different lag values

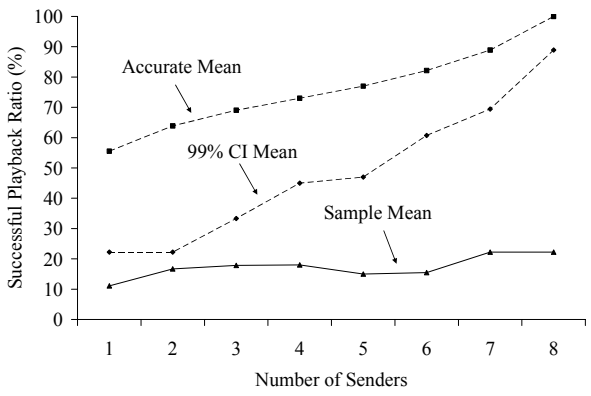

Figure 4. Comparison of successful playback ratio when using Sample Mean, 99% CI Mean, and Accurate Mean (PlanetLab traces)

B. Temporal Correlations

Another issue discussed in Section VI-A is the independence of the aggregate available bandwidth across different time intervals. To investigate this issue we compute the correlations between the aggregate available bandwidth of different lag τ, where lag is the temporal distance between them, i.e., between C_j and $C_{j+\tau}$.

Fig. 3 shows the CDF of the absolute value of the temporal correlations of the PlanetLab trace data, using a time interval of 1 second. A correlation value of 0.2 and lower is considered to be independent. Not surprisingly there are a fair amount of temporal correlations between adjacent samples (e.g., lag of 1) but then the correlations diminish rapidly for larger lags. Considering that a typical video session often last for hundreds, if not thousands, of seconds only a small portion of the C_j's will be correlated and so the impact should be small. Nevertheless we are conducting additional measurements and calculations to more thoroughly quantify the impact of partial temporal correlation to the buffering algorithm's performance.

C. The Predictive Buffering Algorithm

In this section we evaluate the performance of the proposed predictive buffering algorithm using trace-driven simulations. The video length is 1800 seconds and the video bit-rate varies from 1 to 1.3 times the mean aggregate available bandwidth (i.e., R/μ = 1, 1.1, 1.2 and 1.3). Thus other than the case of R/μ=1 all other cases suffer from insufficient bandwidth and so rely on the predictive buffering algorithm to determine the minimum buffer time needed to ensure continuous video playback. In case the client runs into buffer underflow due to data not arriving in time for playback, it will suspend playback and then rerun the predictive buffering algorithm to buffer sufficient video data before resuming playback. An alternative approach (not used in this work) would be to continue playback despite the missing data and then attempt to conceal the visual degradation through error concealment techniques. In this latter approach playback performance will then be measured by the visual quality (e.g., PSNR) instead.

Fig. 4 plots the successful playback ratio using the PlanetLab traces with video bit-rate ratio equals to 1.1 (i.e., R/μ = 1.1). Successful playback ratio is the proportion of simulation runs with no playback interruption (i.e., buffer underflow) during the entire video playback session. There are three curves in the figure: the *Sample Mean* curve is plotted with the mean aggregate available bandwidth (i.e., $\mu(1,w)$) of the initial buffering period as input to computing the minimum buffering time using (3); the *99% CI Mean* curve is plotted with the lower limit of the 99% confident interval mean (c.f. (5) and (6)) as input to (3); and the *Accurate Mean* curve is plotted with the real actual mean of the aggregate flow as input to (3). This last case is not realizable as it requires *a priori* knowledge of future available bandwidth.

The first observation is that the performance when using the sample mean is significantly lower than the case when the 99% CI mean is used. This is because in this simulation the video bit-rate is only 1.1 times the mean available bandwidth and so the resultant buffering time is relatively short, thereby leading to inaccurate measurement of the bandwidth parameters. In our other simulations with higher video bit-rate ratios the difference will become substantially smaller as the buffering period lengthens.

The second observation is that the performance increases as the number of senders in the aggregate flow increases. With 8 senders the performance of the predictive buffering algorithm using the 99% CI mean already approaches the case when the accurate mean is known. This is a direct result of the improved stationarity of the aggregate available bandwidth when there are many senders.

Using the 99% CI mean we investigate further the performance of the predictive buffering algorithm at video bit-rate ratios ranging from 1 to 1.3 using the NLANR PMA traces (Fig 5-7) and the PlanetLab traces (Fig. 8-10). To provide a finer scale for performance comparison we plot in Fig. 6 and 9 the average pause count – the average number of buffer-underflow-induced playback interruptions per streaming session, and in Fig. 7 and 10 the average underflow time – the average total duration of playback suspension per streaming session.

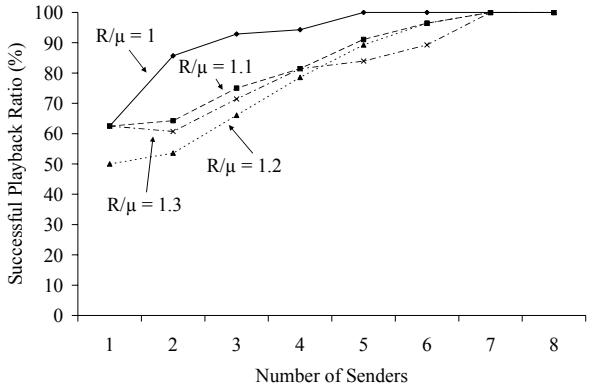

Figure 5. Successful playback ratio for NLANR PMA traces

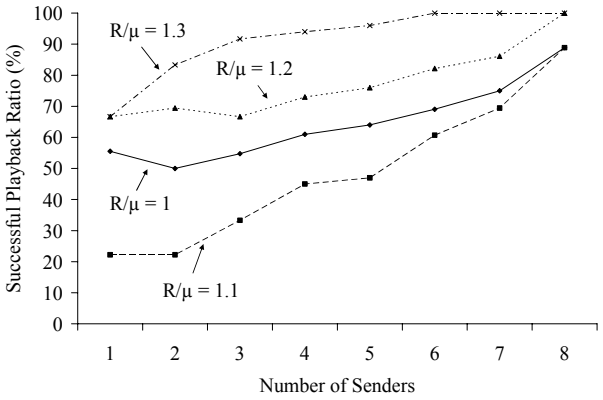

Figure 8. Successful playback ratio for PlanetLab traces

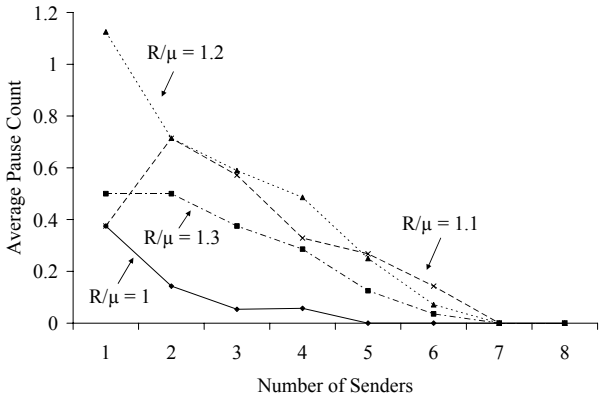

Figure 6. Average pause count for NLANR PMA traces

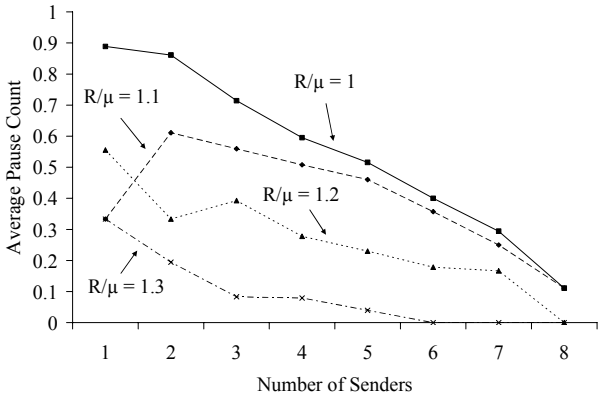

Figure 9. Average pause count for PlanetLab traces

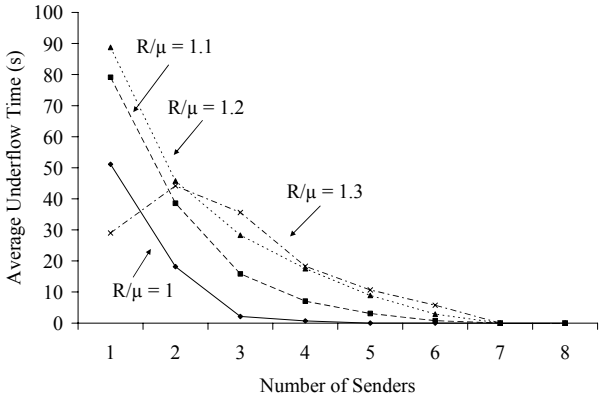

Figure 7. Average underflow time for NLANR PMA traces

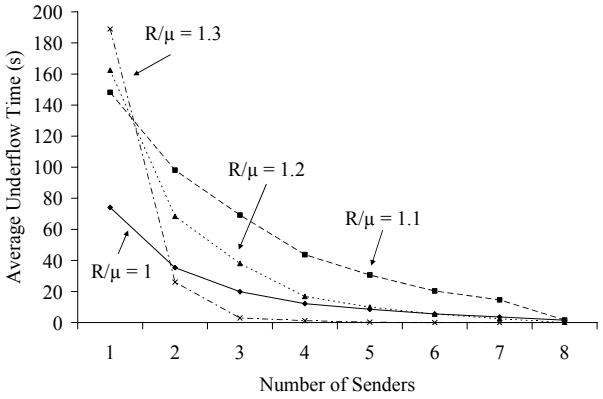

Figure 10. Average underflow time for PlanetLab traces

The results show that increasing the number of senders generally results in better performance, i.e., higher successful playback ratios, fewer playback pauses, and shorter underflow time for all 4 cases of video bit-rate ratios in both traces. In fact the algorithm achieves a successful playback ratio of 100% when there are 7 or more senders for the NLANR PMA traces.

The previous results focus on the video playback performances. In Fig. 11 we plot the average buffering time computed by the predictive buffering algorithm for 8 different sets of simulation configurations. Runs 1 to 4 are from the PlanetLab traces with R/μ = 1, 1.1, 1.2 and 1.3 respectively, and runs 5 to 8 are from the NLANR PMA traces with R/μ = 1, 1.1, 1.2 and 1.3 respectively. In the same figure we also plot the upper bound – which is the time to download the entire video, and the lower bound – which is the minimum buffering time required for continuous video playback assuming all the future bandwidth availabilities are known *a priori*. This latter bound is again not realizable in practice but provides a useful benchmark to evaluate the absolute performance of the predictive buffering algorithm.

There are four observations. First, there is an increasing trend from sets 1 to 4, and from sets 5 to 8. This is due to the increasing video bit-rate ratios (R/μ) used, which longer buffering times are needed to compensate for the higher video bit-rate ratios. Second, the computed buffering time is remarkably close to the lower bound, meaning that the predictive buffering algorithm can achieve near-optimal buffering time and maintain a high successful playback ratio. Third, the differences between the Sample Mean values and the 99% CI mean values are negligible. This shows that by using the confidence interval mean we can achieve substantially better successful playback ratio (c.f. Fig. 4) and yet with only negligible increase in the buffering time. Finally, the average buffering time for PlanetLab traces (runs 1 to 4) is generally closer to the lower bound than the one for the NLANR PMA traces (runs 5 to 8). This is because the NLANR PMA traces generally exhibit significantly more and larger variations in the available bandwidth. Thus to compensate for the larger bandwidth variations the predictive buffering algorithm must extend the buffering time longer to ensure continuous playback.

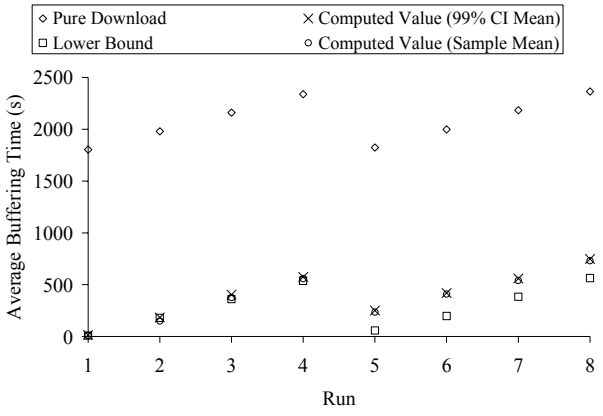

Figure 11. Average buffering time for different simulation runs

D. Comparison with Simple Buffering

Finally, to determine the additional performance gain due to the predictive buffering algorithm, we conducted another set of simulations using the same traffic traces and simulation parameters, but using a simple buffering algorithm based on mean bandwidth estimation. Specifically, in this simple buffering algorithm, the future bandwidth availability is assumed to be a constant C which is assumed to equal to the sample mean of the aggregate bandwidth during the initial buffering (measurement) period. With this assumption, the client will need to satisfy the following constraint to ensure successful playback:

$$w'C-(R-C)L\geq 0 \qquad (9)$$

where w' is the time to start playback, R is the video bit-rate and L is the video length.

Fig. 12-15 plot the percentage reduction in the average pause count and average underflow time when replacing the simple buffering with the proposed predictive buffering algorithm. In both traffic traces (NLANR PMA and PlanetLab) the proposed predictive buffering algorithm can further improve playback performance compared to the simple buffering algorithm. The extend of the performance gains, however, differs substantially between the two traffic traces. In particular, the performance gains of the predictive buffering algorithm over simple buffering in the PlanetLab traces are significantly higher than those in the NLANR PMA traces.

To explain this observation we plot in Fig. 16 the measured average available bandwidth C versus the length of the initial estimation duration in simple buffering. Comparing the two traces we observe that the NLANR PMA traces exhibit significantly lower estimated available bandwidth even for estimation duration as long as 1000 s. Consequently, the simple buffering algorithm will underestimate the available bandwidth C, and thus ends up buffering more than sufficient video data before beginning playback. This results in high successful playback rates that leave little room for improvement by the predictive buffering algorithm. By contrast, the same underestimation does not exist in the PlanetLab traces and thus the predictive buffering algorithm substantially outperforms the simple buffering algorithm.

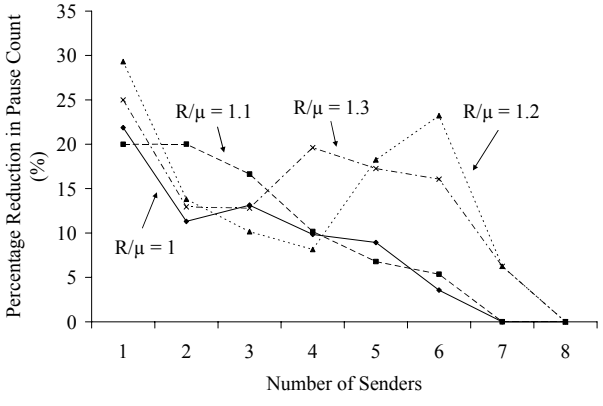

Figure 12. Percentage reduction in pause count for NLANR PMA traces

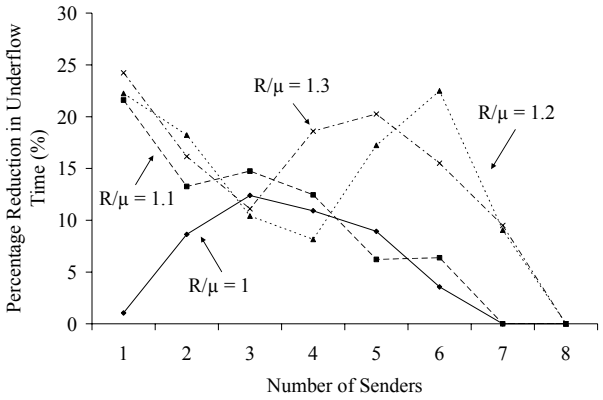

Figure 13. Percentage reduction in underflow time for NLANR PMA traces

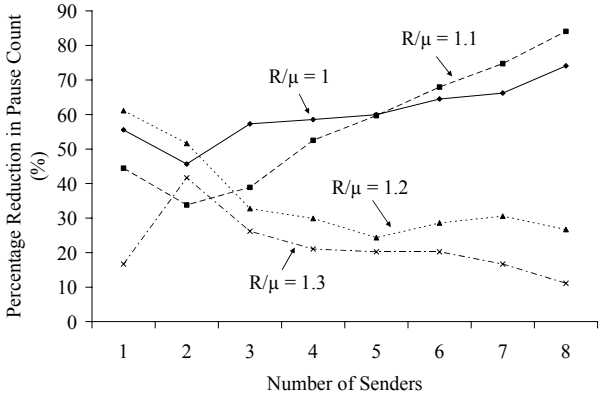

Figure 14. Percentage reduction in pause count for PlanetLab traces

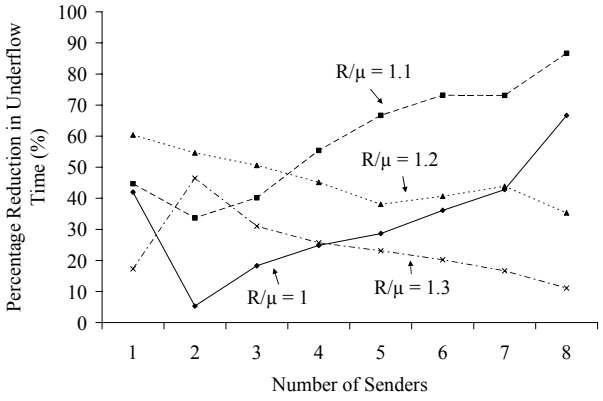

Figure 15. Percentage reduction in underflow time for PlanetLab traces

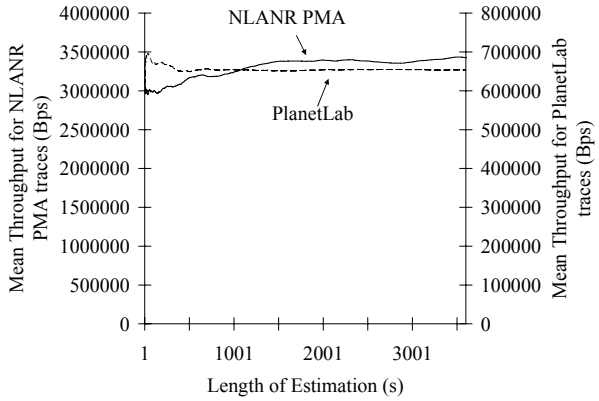

Figure 16. Mean throughput versus the length of estimation for NLANR PMA traces and PlanetLab traces

VI. SUMMARY AND FUTURE WORK

In this work we developed a new predictive buffering algorithm for streaming video from multiple senders to a receiver. The proposed algorithm incorporates the impact of variations in the available bandwidth and uses that knowledge to inform the buffering operation. The trace-driven simulation results show that the predictive buffering algorithm can achieve very high successful playback ratio while keeping the buffering time close to the optimal.

Beyond this work there are still a number of open problems that warrant further study. First, due to resource constraints our current measurements conducted in the PlanetLab are limited to durations of approximately 2 hours. Although the current results show that the aggregate available bandwidth in this time scale is relatively stationary, whether it is still true for even longer durations or for non-PlanetLab hosts remains an open question. This will impact applications where very long video sessions are streamed, such as video conference proceedings.

On the other hand, the predictive buffering algorithm can also be integrated with content adaptation [1-4] and playback rate adaptation algorithms [11, 18] to further increase the system's resilience to bandwidth fluctuations and to support the streaming of live videos.

Second, in practical applications some of the senders may be highly correlated, e.g., if they share the same network bottleneck, and in such cases one will either need a way to identify and eliminate such senders, or to extend the algorithm to compensate for the correlations. Other related open problems include sender-limited rather than network-limited bandwidth variations, the discovery and selection of senders, the interference between multiple video sessions with overlapping senders, and so on.

REFERENCES

[1] P. de Cuetos and K.W. Ross, "Adaptive Rate Control for Streaming Stored Fine-Grained Scalable Video," *Proc. NOSSDAV*, May 2002, pp.3-12.

[2] P. de Cuetos, P. Guillotel, K.W. Ross and D. Thoreau, "Implementation of Adaptive Streaming Of Stored MPEG-4 FGS Video Over TCP," *Proc. ICME* 2002, pp.405-408.

[3] S. Jacobs and A. Eleftheriadis, "Streaming Video using Dynamic Rate Shaping and TCP Congestion Control," *Journal of Visual Comm and Image Representation*, Vol. 9, No. 3, 1998, pp.211-222.

[4] L. S. Lam, Jack Y. B. Lee, S. C. Liew, and W. Wang, "A Transparent Rate Adaptation Algorithm for Streaming Video over the Internet," *Proc. 18th International Conference on Advanced Information Networking and Applications*, Fukuoka, Japan, March 29-31, 2004.

[5] Nguyen and A. Zakhor, "Distributed Video Streaming over the Internet," *SPIE Conference on Multimedia Computing and Networking*, San Jose, California, January 2002

[6] Nguyen and A. Zakhor, "Distributed Video Streaming with forward error correction," *Packet Video Workshop*, PA, USA, April 2002

[7] D.Y. Xu, M. Hefeeda, S. Hambrusch and B. Bhargava, "On Peer-to-Peer Media Streaming," *Proc. Int'l Conf on Distributed Computing Systems 2002*, Vienna, Austria, pp.363-371, July 2002.

[8] J. B. Kwon and H. Y. Yeom, "Distributed Multimedia Streaming over Peer-to-Peer Network," *Proc. 9th Int'l Conf on Parallel and Distributed Computing*, Klagenfurt, Austria, August 2003.

[9] V. Agarwal and R. Rejaie, "Adaptive Multi-source Streaming in Heterogeneous Peer-to-Peer Networks," *SPIE Conf on Multimedia Computing and Networking*, San jose, California, January 2005.

[10] S. C. Hui and Jack Y. B. Lee, "Modeling of Aggregate Available Bandwidth in Many-to-One Data Transfer," *Proc. of the 4th Int'l Conf on Intelligent Multimedia Computing and Networking*, July 21-26, 2005, Utah, USA.

[11] S. C. Hui and Jack Y. B. Lee, "Playback-Adaptive Multi-Source Video Streaming," *Proc. of the 4th Int'l Conf on Intelligent Multimedia Computing and Networking*, July 21-26, 2005, Utah, USA.

[12] M. Reisslein and K.W. Ross, "Call Admission for Prerecorded Sources with Packet Loss," *IEEE Journal Selected Areas in Communications*, vol. 15, pp.1167-1180, August 1997.

[13] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP friendly rate control (TFRC): Protocol specification," *RFC 3448*, January 2003.

[14] L. Lapin, *Modern Engineering Statistics*, Duxbury Press, 1997.

[15] NLANR PMA data set: http://pma.nlanr.net/Traces/long/bell1.html.

[16] NS2 official homepage at http://www.isi.edu/nsnam/ns/.

[17] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: An Overlay Testbed for Broad-Coverage Services," *ACM SIGCOMM Computer Communication Review*, vol. 33(3), pp. 3-12, July 2003.

[18] M. Kalman, E. Steinbach, and B. Girod, "Adaptive Media Playout for Low-Delay Video Streaming Over Error-Prone Channels," *IEEE Trans on Circuits and Systems for Video Technology*, vol.14(6), June 2004, pp.841-851.