

Playout Buffer and Rate Optimization for Streaming over IEEE 802.11 Wireless Networks

MINGZHE LI, MARK CLAYPOOL, and ROBERT KINICKI
Worcester Polytechnic Institute, USA

Most streaming rate selection and buffer optimization algorithms are developed for wired networks and can perform poorly over wireless networks. Wireless MAC layer behavior, such as rate adaptation, retransmissions, and medium sharing, can significantly degrade the effectiveness of current streaming algorithms. This article presents the Buffer and Rate Optimization for Streaming (BROS) algorithm to improve streaming performance. BROS uses a bandwidth estimation tool designed specifically for wireless networks and models the relationship between buffer size, streaming data rate, and available bandwidth distribution. BROS optimizes the streaming data rate and initial buffer size, resulting in a high data rate but with few frame losses and buffer underflow events, while still keeping a small initial buffer delay. BROS is implemented in the Emulated Streaming (EmuS) client-server system and evaluated on an IEEE 802.11 wireless testbed with various wireless conditions. The evaluation shows that BROS can effectively optimize the streaming rate and initial buffer size based on wireless network bandwidth conditions, thus achieving better performance than static rate or buffer selection and jitter removal buffers.

Categories and Subject Descriptors: C.2.m [Computer-Communication Networks] Miscellaneous

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Multimedia networking, playout buffer, streaming rate, wireless networks

ACM Reference Format:

Li, M., Claypool, M., and Kinicki, R. 2009. Playout buffer and rate optimization for streaming over IEEE 802.11 wireless networks. *ACM Trans. Multimedia Comput. Commun. Appl.* 5, 3, Article 26 (August 2009), 25 pages.
DOI = 10.1145/1556134.1556143 <http://doi.acm.org/10.1145/1556134.1556143>

1. INTRODUCTION

In best-effort networks, streaming media applications use streaming rate selection and playout buffers to reduce degradations in performance caused by changes in the available bandwidth along the path of the streaming flow. However, most streaming rate selection and buffer optimization algorithms are developed for wired networks and can perform poorly over wireless networks. Wireless networks often introduce rate adaptation, retransmissions, Forward Error Correction (FEC), and channel access control, and other behaviors not normally found in wired networks. This mismatch in design leads to significant degradation in the effectiveness of rate selection and playout buffer techniques.

Typical streaming rate selection for media scaling [Delgrossi et al. 1993] is based on loss rate, round-trip time, or a bandwidth estimate to adjust the streaming data rate to reduce the network impact on

Authors' address: M. Li, M. Claypool (corresponding author) R. Kinicki, Department of Computer Science, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609-2280; email: claypool@cs.wpi.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2009 ACM 1551-6857/2009/08-ART26 \$10.00 DOI 10.1145/1556134.1556143 <http://doi.acm.org/10.1145/1556134.1556143>

ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 5, No. 3, Article 26, Publication date: August 2009.

media performance. However, these measurements do not always provide clear indications of wireless network conditions and target rates for adaptation. For example, retransmissions and rate adaptation at the wireless MAC layer may reduce the loss rate while increasing the round-trip times measured by the applications. Thus, wireless network conditions hidden from the application can cause bad media scaling decisions. Moreover, media scaling action is usually taken during degraded performance and therefore is not effective in avoiding performance degradation. Recent research [Kuang and Williamson 2004; Bai and Williamson 2004; Li et al. 2005a, 2005c] shows that media scaling performance is limited when the optimal streaming rate is not correctly selected over wireless networks.

Similarly, widely deployed client-side playout buffer techniques adapt infrequently to network characterizations in wireless networks [Jacobs and Eleftheriadis 1996]. Generally, client-side buffering provides the essential functionality of removing delay jitter and reducing playback disruption caused by oscillations in the transmission rate at the cost of initial startup delay [Birney 2004; Li et al. 2001]. The oscillations in transmission rate may be caused by the transport protocol, such as with TCP or TFRC [Floyd et al. 2000], or by the rate adaptation in a wireless network. Most current playout buffer techniques use either a fixed-sized buffer or choose the minimum buffer size necessary to remove delay jitter. However, for wireless networks with dynamic capacity changes, insufficient buffer sizes decided by jitter removal or static buffer size can produce an increased number of rebuffer events or substantial delays [Kuang and Williamson 2004; Li et al. 2005a, 2005c].

This article presents the Buffer and Rate Optimization for Streaming (BROS) algorithm to improve streaming multimedia application performance by reducing frame loss, buffer underflow events, and buffer delay in wireless networks. BROS applies a low-cost bandwidth estimation approach, the Wireless Bandwidth Estimation tool (WBest) [Li et al. 2008], at the application layer to provide information on the effective capacity, available bandwidth, and variance in bandwidth for the bottleneck wireless link. BROS uses a Markov chain model of the buffer size, streaming data rate, and available bandwidth distribution to concurrently optimize the streaming rate selection and the playout buffer size.

BROS is incorporated into the Emulated Streaming (EmuS) client-server system in Linux and evaluated on an IEEE 802.11 wireless testbed over a variety of wireless conditions that include an idle channel, channels with contending and crossing traffic, and rate adaptation during poor connection conditions. The evaluation shows that BROS can effectively optimize the streaming rate and initial buffer size based on wireless network bandwidth conditions, thus achieving better performance than static rate selection and static or jitter removal buffers. Analysis indicates that BROS can reduce buffer underflow probability by nearly 100%, frame lost rate by about 97%, and the total buffer delay from 78% to 87%, compared with typical static and jitter removal approaches.

The article is organized as follows: Section 2 presents select related work; Section 3 explains the buffer model based on the available bandwidth distribution; Section 4 discusses BROS and related issues; Section 5 describes the experimental setup; Section 6 analyzes the experimental results; Section 7 provides conclusions; and Section 8 suggests possible future work.

2. RELATED WORK

This section reviews related work in the areas of rate selection and playout buffer optimization for streaming media over wireless networks.

2.1 Streaming Rate Selection

Typical techniques to improve the rate selection for streaming media in wireless networks include novel transport layer approaches [Cen et al. 2003; Feng and Sechrest 1995; Floyd et al. 2000; Yang et al. 2004], cross-layer approaches [Hsu et al. 1999; Kazantzidis and Gerla 2003; Li and van der Schaar 2004], and

bandwidth estimation approaches [van Beek et al. 2004; Bolot and Turetti 1994; Demircin and van Beek 2005; van Beek and Demircin 2005].

2.1.1 Transport Protocols. Novel transport protocols for streaming media often seek to mimic the congestion response of TCP, the de facto Internet transport protocol. TCP Friendly Rate Control (TFRC) [Floyd et al. 2000], used in the Datagram Congestion Control Protocol (DCCP) [Kohler et al. 2006], provides for rate-based streaming for smooth, low delay, and TCP-friendly packet transmission. However, TCP (and TCP-friendly protocols) can perform poorly over 802.11 wireless networks because many of the TCP mechanisms assume a wired network infrastructure. Performance can be improved by using wireless loss differentiation techniques, such as in Cen et al. [2003], allowing streaming applications to select rates according to the TCP-friendly rate for the observed wireless network conditions. Using similar techniques, Chen and Zakhor [2005] and Yang et al. [2004] propose new transport protocols that use a TCP-friendly rate in a wireless network to control the streaming rate. However, as discussed by Kazantzidis and Gerla [2003], using a TCP-friendly rate control provides only “trial-and-error” scaling, which is unreliable, converges slowly, and cannot be used for initial streaming rate selection. In general, our work can complement transport protocol enhancements by providing a much faster estimate of available bandwidth as well as an estimate of variance that can be used for sizing playout buffers.

2.1.2 Cross-Layer. Cross-layer approaches take advantage of MAC and physical layer information, such as the rate adaptation, amount of forward error correction, and retransmissions, to control the selection of the streaming rate. For example, Kazantzidis and Gerla [2003] propose a link-network feedback architecture to provide cross-layer information for the streaming media applications to adapt their streaming rates. In addition, Yang et al. [2004] combine the cross-layer and TCP-friendly rate control approach to propose a new protocol that utilizes link layer loss information to improve the TCP-friendly rate control. However, cross-layer approaches are difficult to deploy because they require modifications to end hosts and protocol stacks, and can involve multiple vendor implementations.

2.1.3 Bandwidth Estimation. Bandwidth estimation approaches use application measurements to guide rate selection. A survey by Prasad et al. [2003] provides a review of bandwidth estimation literature focusing on underlying techniques and methodologies. Early work by Bolot and Turetti [1994] adjusts streaming rates based on loss rate information fed back from the receiver, but without actually estimating available bandwidth. Commercial streaming media applications, such as Windows Media Service, use packet pair techniques to estimate the capacity and choose an appropriate streaming rate [Birney 2004]. Most recent research from van Beek et al. [2004], Demircin and van Beek [2005], and van Beek and Demircin [2005] applies packet pairs/trains and receiver-side statistical bandwidth estimation techniques to guide the rate selection in wireless networks. The advantage of the bandwidth estimation approach is that it usually does not depend upon lower-layer information or new protocol stacks, thus it can be relatively easy to deploy. In addition, with careful design, bandwidth estimation can avoid the “trial-and-error” problem caused by a TCP-friendly approach to rate selection. However, Li et al. [2008], Lakshminarayanan et al. [2004], and Angrisani et al. [2006] show that traditional bandwidth techniques designed for wired networks cannot accurately estimate the bandwidth in wireless networks. Moreover, these techniques usually provide capacity estimation, while streaming rate selection needs a broader range of bandwidth information, such as the available bandwidth and the variance in available bandwidth, when dealing with contending and crossing traffic over the WLAN.

2.2 Playout Buffer Optimization

There are a variety of strategies proposed to improve the effectiveness of client-side buffering that include optimizing the buffer size based on jitter removal [Mundur et al. 1999; Ramjee et al. 1994;

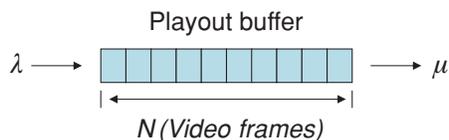


Fig. 1. Buffer model.

Moon et al. 1998; Fujimoto et al. 2002], slowing down the media playout rate at the client to reduce the consumption rate and help prevent buffer underflow [Yuang et al. 1996; Steinbach et al. 2001; Laoutaris and Stavrakakis 2001; Kalman et al. 2004], and mitigating the buffer size required by VBR media codecs [Yuang et al. 1997; Feng and Sechrest 1995]. Since VBR streaming smoothers usually have a static size based on the media codec, and the adaptive playout rate can be used as an additive for reducing the buffer underflow events in most streaming systems, most buffer optimization algorithms are primarily for jitter removal.

To study the buffer size required for removing the jitter in networks, Yuang et al. [1996, 1997], Steinbach et al. [2001], Kalman et al. [2004], and Laoutaris and Stavrakakis [2001] present Markov chain models based on Poisson arrivals. However, in wireless networks, the arrival of streaming traffic cannot be simply modeled as a Poisson distribution because the capacity changes due to wireless rate adaptation algorithms produces variance in the streaming traffic arrival rate. Therefore, jitter removal buffer algorithms are not sufficient to avoid buffer underflow in wireless networks.

Stockhammer et al. [2004] provide a model for setting a delay buffer based on a deterministic VBR channel. However, they provide an example specific to wide-area wireless networks only and solely validate their findings through simulation.

3. MODEL

3.1 Bandwidth Estimation

Our previous research developed a low-cost technique, called Wireless Bandwidth Estimation (WBest) [Li et al. 2008], to measure available bandwidth in a wireless network. Based on a model of packet dispersion in 802.11 networks, WBest first uses packet pairs to estimate the effective capacity, followed by a packet train to estimate the overall available bandwidth in the wireless network. For output, WBest provides a Cumulative Distribution Function (CDF), $F_A(\cdot)$, of the available bandwidth which can be used to estimate the overall available bandwidth (to assist in choosing a streaming rate) and to estimate the variance in available bandwidth (to assist in choosing a buffer size). While the temporal properties of successive available bandwidth measurements are not necessarily captured in the CDF, the intent is to provide a quick snapshot of the network and not necessarily to provide an accurate network model. WBest adds only modest delay to the start of a streaming session, taking less than 0.5 seconds to complete in all typical wireless environments.

While BROS currently employs WBest only once, prior to starting the video streaming, to capture the time-varying nature of the current wireless conditions within a CDF estimate, if the volatility of the WLAN (Wireless Local Area Network) is high, WBest could be event-triggered during the streaming process to allow a subsequent BROS adjustment to significant changes in the wireless conditions. This enhancement is left as future work.

3.2 Model Definitions and Assumptions

Figure 1 depicts a typical client-side playout buffer system with a buffer size of N frames, arrival rate λ , and playout rate μ . Based on buffer occupancy, we create a Markov model of $N + 1$ states as shown

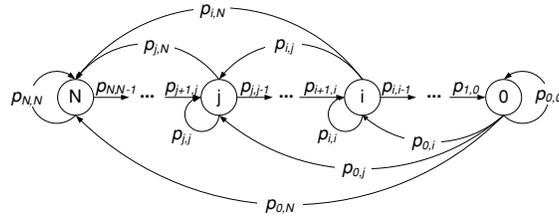


Fig. 2. States of the buffer model.

in Figure 2. State N is defined as having a buffer with N frames, where state 0 is a buffer underflow.¹ The matrix P in Eq. (1) presents the transition probability of states i and j , where $0 \leq i, j \leq N$.

$$P_{i,j} = [p_{i,j}] = \begin{bmatrix} p_{0,0} & p_{0,1} & p_{0,2} & \cdots & p_{0,N} \\ p_{1,0} & p_{1,1} & p_{1,2} & \cdots & p_{1,N} \\ p_{2,0} & p_{2,1} & p_{2,2} & \cdots & p_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{N,0} & p_{N,1} & p_{N,2} & \cdots & p_{N,N} \end{bmatrix}. \quad (1)$$

Most of the previously developed playout buffer models use a Poisson arrival process [Yuang et al. 1997, 1996; Steinbach et al. 2001]. Poisson arrivals can be used as a lower bound on system performance when analyzing the buffer behavior [Laoutaris and Stavrakakis 2001]. However, for real streaming applications over wireless networks, the buffer space used to smooth interarrival time variance is small relative to the buffer space needed to smooth the variance in available bandwidth. In networks with a large available bandwidth variance, the expected arrival rate of the streaming packets is impacted by the available bandwidth. For example, if the available bandwidth is less than the streaming rate, the expected arrival rate at the playout buffer will also be less than the streaming rate. Therefore, the transition probability model is based on both the available bandwidth and the streaming rate.

To define the probability matrix, the following assumptions are made. First, as discussed in Kalman et al. [2004]; Lin et al. [1984], packet loss is modeled as a reduction in available bandwidth. Given a playout buffer of a few seconds, a lost packet will have multiple retransmission opportunities. For typical interpacket loss rates of less than 20% [Conklin et al. 2001], the probability that a packet is received after a few retransmission attempts is nearly one [Kalman et al. 2004]. Thus, as shown in Figure 3, lost packets are treated as delayed due to insufficient bandwidth. When bandwidth does become available, delayed packets are sent in bursts at a rate equal to the available bandwidth until the buffer is filled again. Second, once a streaming rate is selected, we can model the multimedia content as a constant data rate R and a constant frame rate μ for both streaming and decoding. While frame sizes do depend upon the encoding and type of encoded frame, such as I, B, and P frames in MPEG encoding, a constant frame size $S = R/\mu$ is used to simplify our model. For real streaming systems, the constant frame size assumption can be accommodated by applying an additional buffer, at the sender or receiver, to smooth out the variable frame sizes due to I, B, and P frame size differences.

¹Client-side playout buffers can be quite large, often big enough to buffer an entire video clip. Thus, a buffer *overflow* is extremely unlikely and is not considered further.

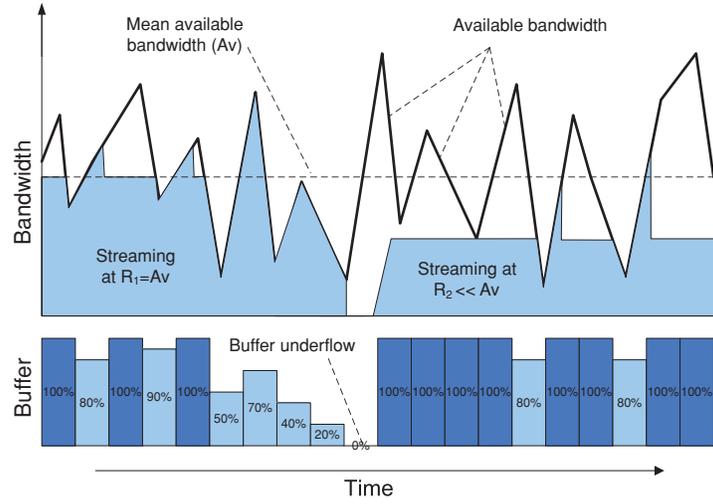


Fig. 3. Streaming rate and available bandwidth.

Therefore, assuming an available bandwidth constraint on the streaming traffic, the frame arrival rate λ_i in frame per seconds is

$$\lambda_i = \begin{cases} A_i/S, & A_i \leq R; \\ A_i/S, & A_i > R, i < N; \\ \mu, & A_i > R, i = N; \end{cases} \quad (2)$$

where A_i is the available bandwidth in bits per seconds at state i . Assuming a constant arrival rate λ_i within each frame time $t = 1/\mu$, the expected number of frames n_i that arrive in each frame slot can be approximated by $n_i = \lambda_i/\mu$. This facilitates a model of the transition probability between states based on the distribution of the available bandwidth A . For example, given the Cumulative Distribution Function (CDF) $F_A(\cdot)$ of the available bandwidth, the probability of $1 < n_i \leq 2$ can be determined by $F_A(A_i = 2R) - F_A(A_i = R)$, which is the transition probability from state i to $i+1$, where $A_i > R, i < N$. Similarly, the transition probability matrix can be defined for the buffer model, henceforth referred to as the full model.

$$p_{i,j} = \begin{cases} F_A[(j-i+1)R] - F_A[(j-i)R], & 0 \leq i < j < N, \\ & j-i \leq A_i/R; \\ 1 - F_A[(j-i)R], & 0 \leq i < j, j = N, \\ & j-i \leq A_i/R; \\ 1 - \sum_{j=0, j \neq i}^N p_{i,j}, & 0 \leq i \leq N, j = i; \\ F_A(R), & 0 < i \leq N, \\ & j = i-1; \\ 0, & elsewhere. \end{cases} \quad (3)$$

The steady state probability distribution of the buffer occupancies, $\Pi = [\pi_0, \pi_1, \dots, \pi_N]$ can be directly computed by solving the stationary equation $\Pi = \Pi P$ and $\sum_{k=0}^N \pi_k = 1$. The probability of 0 buffer occupancy is the expected buffer underflow probability. However, since the full model characterization

does not lend itself to a closed-form solution, it is cumbersome to use for real systems due to the required massive matrix computations.

As shown in the right side of Figure 3, lowering the streaming rate below the average available bandwidth reduces the buffer requirement to avoid buffer underflow. Therefore, by focusing on situations where the multimedia streaming rate is close to the average available bandwidth, the full model can be simplified to yield a closed-form solution for the buffer underflow probability. When $A_i/R \approx 1$, then $p_{i,i+1} \gg p_{i,i+2}, \dots, p_{i,N}$, the transition matrix P can be further reduced to a simplified buffer model.

$$p_{i,j} = \begin{cases} 1 - F_A(R), & 0 \leq i < N, j = i + 1 \\ & j - i \leq A_i/R; \\ 1 - \sum_{j=0, j \neq i}^N p_{i,j}, & 0 \leq i \leq N, j = i; \\ F_A(R), & 0 < i \leq N, j = i - 1; \\ 0, & elsewhere. \end{cases} \quad (4)$$

Therefore, the closed-form solution for the simplified buffer model becomes

$$\pi_i = \gamma^i \pi_0, \quad (5)$$

where $\gamma = (1 - F_A(R))/F_A(R)$. The buffer underflow probability π_0 for a given buffer size of N frames is

$$\pi_0 = \frac{1 - \gamma}{1 - \gamma^{1+N}}, \quad \gamma \neq 1. \quad (6)$$

A streaming system with $\gamma \leq 1$ means streaming at a rate greater than the average available bandwidth and usually results in a high buffer underflow probability. Therefore, the streaming rate selection algorithm selects the initial streaming rate such that $\gamma > 1$. Moreover, if a streaming application demands an upper bound on the buffer underflow in terms of π_0 , γ can be computed from π_0 using Eq. (6), and the streaming data rate can be looked up from the inverse CDF of the available bandwidth $F_A^{-1}(\cdot)$ by

$$R = F_A^{-1}\left(\frac{1}{\gamma + 1}\right). \quad (7)$$

As discussed in Kalman et al. [2004], the Mean Time Between Buffer Underflows (MTBBU) can be used as a measure of performance for playout buffering. Each discrete frame slot can be treated as an independent Bernoulli trial with the outcome being either underflow with probability π_0 or no underflow with probability $1 - \pi_0$. Thus, the MTBBU, M_U , is distributed geometrically over the succession of frame slots as

$$M_U = \left(\frac{1}{\pi_0}\right) \cdot \left(\frac{1}{\mu \cdot 60}\right), \quad (8)$$

where μ is the playout rate in frames per second and 60 is the number of seconds in a minute. Given M_U and the CDF of the available bandwidth, Eq. (9) is the required buffer size N in frames.

$$N = \left\lceil \frac{\log(1 + (\gamma - 1)(M_U \cdot \mu \cdot 60))}{\log \gamma} - 1 \right\rceil \quad (9)$$

In practice, the minimum client-side buffer, N' , may also include extra buffer space required for video decoding or playback. For example, an extra buffer B_{min} may be needed for handling VBR video or encoding dependencies. This research assumes $B_{min} = 1$, which means only the frame that is currently

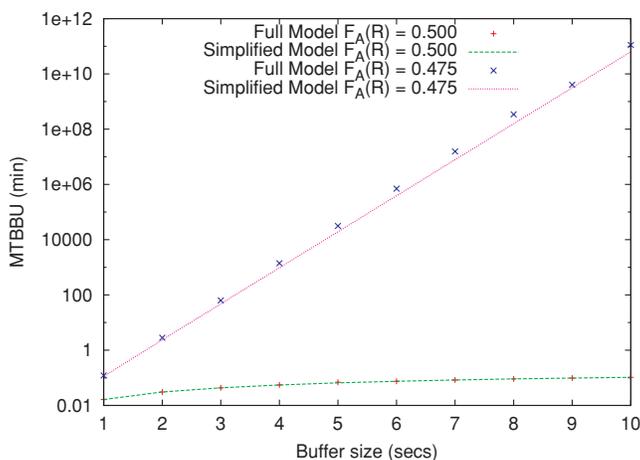


Fig. 4. Comparison of full and simplified buffer models.

being played out is considered. Eq. (10) shows the expression that can be used to predict the minimum buffer size in practice.

$$N' = N + B_{min} = N + 1 \quad (10)$$

3.3 Model Validation

To study the impact of the simplifications to the full buffer model, outputs from both the full model and simplified model are compared. Figure 4 depicts the MTBBU of the full model, which is computed using numerical matrix computation based on Eq. (3), and the MTBBU of the simplified model based on Eq. (4). The CDF of the available bandwidth is based on the trace of 900 samples of bandwidth estimations using WBest in a wireless testbed with dynamic rate adaptation. It is not possible to consistently send data at a rate above the median of the available bandwidth distribution (0.5 in the CDF), as doing so represents an offered load greater than the service rate of the network link. However, in general, the more bits used for streaming the better the quality. Hence, a general guideline for a streaming rate is close to, but less than, the median of $F_A(R)$.

In all cases, the simplified model is close to the full model, especially when the buffer size is small or the streaming rate is close to the median available bandwidth. Moreover, for the $F_A(R) = 0.475$ cases, the simplified model has a lower MTBBU than does the full model due the approximation of $p_{i,i+1} \gg p_{i,i+2}, \dots, p_{i,N}$. However, this small error encourages a conservative estimate for selecting a playout buffer size (e.g., to get the same MTBBU, the simplified model demands a slightly larger buffer size). The simplified model greatly reduces the amount of matrix computation required over the full model.²

The model is further validated with the EmuS system in an IEEE 802.11 wireless testbed using the setup as discussed in Section 5. A 10-minute multilayer video is streamed with a one-second fixed buffer at different streaming rates, and the MTBBUs (in minutes) are recorded. The wireless testbed is configured with 802.11b and a poor reception signal strength of -89 dBm that causes MAC layer retransmissions and frequent rate adaptation. The available bandwidth is monitored by WBest before each run and the average available bandwidth is around 1.12Mbps. A wireless sniffer is used to capture

²Our numerical solution computation takes about 1.5 seconds on a typical PC to solve the full buffer model for 300 frames.

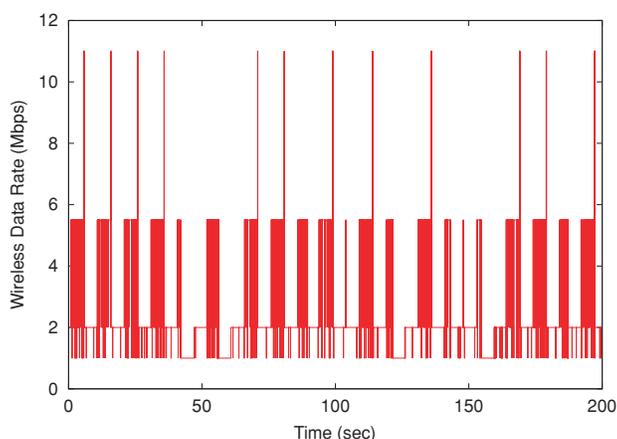


Fig. 5. Typical rate adaptation.

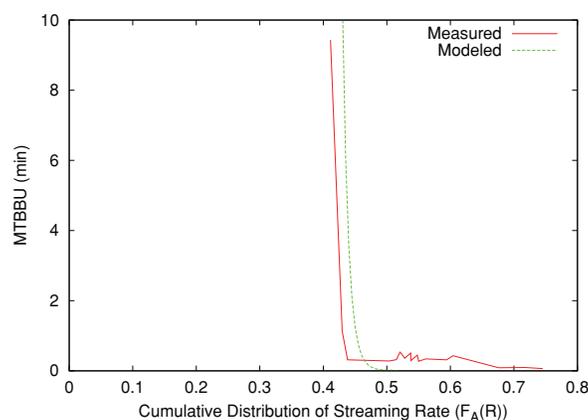


Fig. 6. MTBBU and the CDF of the streaming rate.

the rate adaptation of the wireless network, as shown in Figure 5. The streaming rate, R , as a CDF of available bandwidth, $F_A(R)$, is computed and the relationship between $F_A(R)$ and MTBBU is depicted in Figure 6. The MTBBU for the one-second buffer is also computed based on the simplified buffer model.

As seen in Figure 6, the measured results fit well with our buffer model, with the measured and modeled curves having the same shape at nearly the same places. The measured curve shifted slightly to the left indicates that for the same streaming rate, wireless uncertainties such as bursty loss and black outs result in a small MTBBU, and thus a higher buffer underflow probability than in the model. Moreover, the measured MTBBU for high streaming rates, such as for rates with $F_A(R) > 0.5$, is slightly larger than the modeled MTBBU. This is because in the streaming client-server system implemented, when the buffer underflows, the rebuffer mechanism stops playback of the video until the buffer is filled again, which effectively increases the buffer size for each buffer underflow and intentionally raises the MTBBU for periods after the underflow.

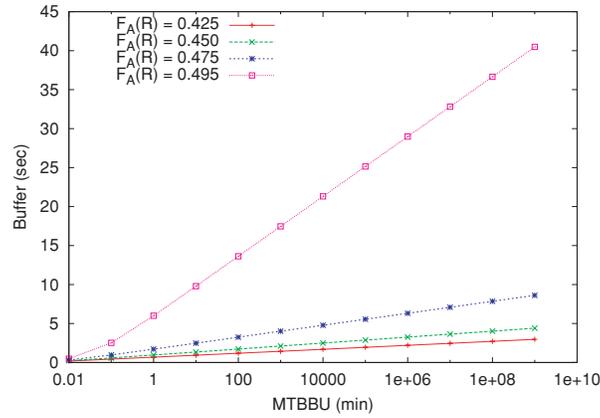


Fig. 7. Required buffer size versus MTBBU.

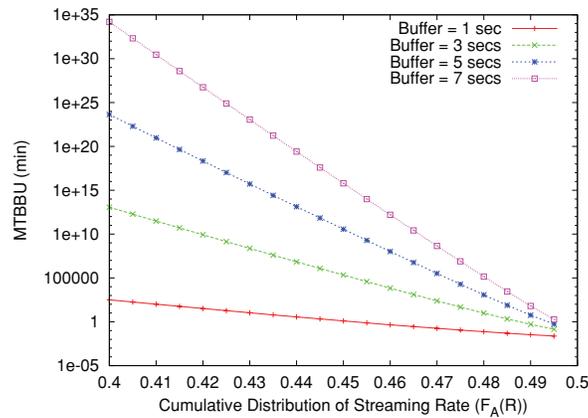


Fig. 8. MTBBU versus streaming rate.

3.4 Model Results

The streaming buffer model can be used to estimate the minimum buffer required to achieve a MTBBU with a given rate and known network conditions, such as the CDF of the available bandwidth $F_A(\cdot)$. Figure 7 shows the minimum buffer size required for various selections of the streaming rate as a fraction of the available bandwidth. The available bandwidth distribution ($F_A(\cdot)$) was provided by WBest on the same wireless setup with rate adaptation used in Section 3.3. From Figure 7, the higher the streaming rate, the larger the buffer needed to achieve a desired MTBBU. Moreover, as the streaming rate gets closer to the median of the available bandwidth, the benefits to MTBBU for an increasing buffer size get smaller.

The buffer model can also be used to explore the relationship between MTBBU, buffer size, and streaming rate from a different aspect. Figure 8 depicts the relationship of MTBBU and streaming rate for 4 different fixed-sized buffers: 1, 3, 5, and 7 seconds. For a given buffer size, reducing the streaming rate increases the MTBBU, thus providing fewer buffer underflow events. For a modest buffer size (i.e., 5 seconds), a small decrease in the streaming rate selected results in a greatly increased MTBBU.

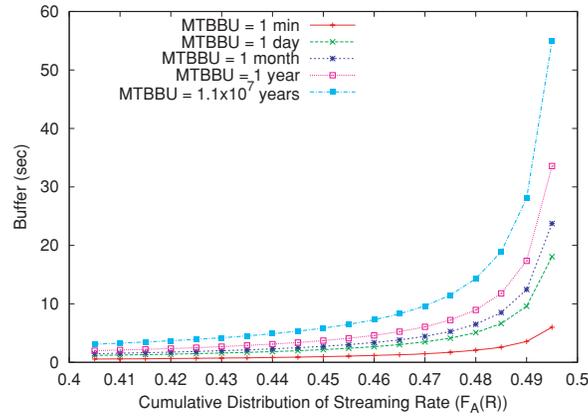


Fig. 9. Buffer size versus streaming rate.

Similarly, for a small buffer size (i.e., one second), a small decrease in the streaming rate has much less effect on the MTBBU.

Figure 9 shows the relationship between buffer size and streaming rate for a given MTBBU, a value that could be provided by the content service provider. The higher MTBBU values are more sensitive to the buffer size as the streaming rate selected is increased, with dramatic increases in required buffer size as the streaming rate approaches the median (0.5, on the far right). The “knee” for the curves occur when the streaming rate selected is around 0.485 of the available bandwidth. Streaming rates higher than this show a marked increase in the buffer size requirements. However, also note at the far left of the graph that streaming rates only slightly lower, around 0.4 of the available bandwidth, can have a marked decrease in the MTBBU for only a slight (around 1 second) increase in the buffer size.

4. BUFFER AND RATE OPTIMIZATION FOR STREAMING (BROS)

4.1 Algorithm

Current media servers such as Windows Media Services and Real Networks use a seemingly arbitrary initial period fixed at several seconds to fill the client-side playout prior to starting the playout of the video at the client. This strategy highlights the trade-off between how much initial delay the client will tolerate and the desire to avoid a major loss in client satisfaction when the server stalls the video to refill the playout buffer in response to a buffer underflow.

This section presents the BROS algorithm that takes a more formal quality-of-service approach to this inherent trade-off. Starting from a user-specified maximum tolerable buffer size and target MTBBU, the simplified buffer model of Section 3 and WBest are used to determine the optimum streaming rate and the minimum buffer size needed to meet a client’s QoS requirements. WBest is employed by BROS because it specifically estimates available bandwidth when the last hop to the client involves a wireless network.

Algorithm 1 provides the details of the BROS (Buffer and Rate Optimization for Streaming) mechanism. After the client specifies the two QoS constraints, BROS proceeds in four stages. At line 2, WBest is run to estimate the average and variance of available bandwidth. As detailed in Section 4.2, these estimates are then used to create $F_A(\cdot)$, an accurate estimate of the CDF of the available bandwidth when the last hop is wireless. Line 3 provides shorthand for the second stage that determines the maximum affordable streaming rate, Th_R . Using the simplified model, M_U is used in Eq. (8) to determine π_0 . π_0 and N_{max} are inserted into Eq. (6) to yield γ . With γ , $F_A(\cdot)$ can be then used in Eq. (7) to provide Th_R .

Algorithm 1. Buffer and Rate Optimization for Streaming (BROS).**Require:** Encoding rates R_1 to R_{MAX} , ordered lowest to highest**Require:** Client tolerable delay: $N_{max} > 0$ **Require:** Target MTBBU: $M_U > 0$

```

// Perform WBest Bandwidth Estimation
1: CDF of Available Bandwidth:  $F_A(\cdot) \leftarrow WBest$ 
// Select streaming rate ( $R$ )
2: Rate Threshold:  $Th_R \leftarrow (N_{max}, F_A(\cdot), M_U)$ 
3: for  $i = 1, MAX$  do
4:   if  $R_i < Th_R \leq R_{i+1}$  then
5:      $R \leftarrow R_i$ 
6:   end if
7: end for
8: if  $Th_R \leq R_0$  then
9:   Stop {Not enough bandwidth for streaming}
10: else
11:   // Optimize buffer ( $N'$ )
12:    $N' \leftarrow N(M_U, F_A(R)) + 1$ 
13: end if

```

The third stage of BROS is described in lines 4–8. Assuming the streaming server has MAX distinct encoding rates, this loop selects the highest available encoded streaming rate less than Th_R . Once R is chosen, the last stage at line 13 involves using R and $F_A(\cdot)$ to determine a new value for γ which is then used with M_U to compute a new N and N' using Eqs. (9) and (10). If there is no encoded streaming rate less than Th_R , the algorithm exits.³

BROS is run at the streaming server. The streaming client typically provides the maximum tolerable buffer size and target MTBBU when the initial connection to the server is made. WBest is run between server and client, with the client returning the results to the server. When the server selects the streaming rate and buffer size, this information is conveyed back to the client and then streaming commences. The client buffers the streaming data until the playout buffer is full and then starts playout.

4.2 Approximate CDF of Available Bandwidth

Even though WBest provides the mean and standard deviation of the available bandwidth estimates, it is difficult to get a closed-form equation for the CDF of the available bandwidth. To simplify the algorithm and reduce the time of recording (and transmitting) multiple bandwidth estimate samples that make up the CDF, the estimated mean μ_A and standard deviation σ_A of the available bandwidth are used to approximate the CDF of the available bandwidth using a normal distribution. The approximate CDF of the available bandwidth is $F_A(\cdot) = N(\mu_A, \sigma_A^2)$. To validate this approximation with measurements, four different setups in our IEEE 802.11 wireless testbed are used: an idle channel, a channel with 5Mbps downstream crossing traffic, a channel with 5Mbps upstream contending traffic, and a channel that experiences link rate adaptations. For each setup, WBest is repeated 30 times and the CDF of the available bandwidth is recorded. The normal CDFs are then generated based on the median of the mean and standard deviation of the estimated available bandwidth from the 30 WBest runs. The CDF of the available bandwidth and the approximated available bandwidth CDF using the normal distribution are compared by the relative error of the 10th, 25th, 50th, 75th, and 90th percentile values. The relative error is computed as $(F_A^{-1}(x) - F_{\hat{A}}^{-1}(x))/F_A^{-1}(x)$, where $F_A^{-1}(x)$ and $F_{\hat{A}}^{-1}(x)$ are the

³Note, this case also occurs when there is no available bandwidth.

Table I. Relative Error of Approximating Available Bandwidth Using Normal Distribution

Setup	Percentile				
	10th	25th	50th	75th	90th
Idle channel	0.04	< 0.01	<0.01	-0.07	-0.17
Contending traffic	0.34	0.16	-0.06	-0.23	-0.30
Crossing traffic	0.14	0.14	-0.01	-0.17	-0.31
Rate adaptation	0.03	0.05	-0.01	-0.01	0.03

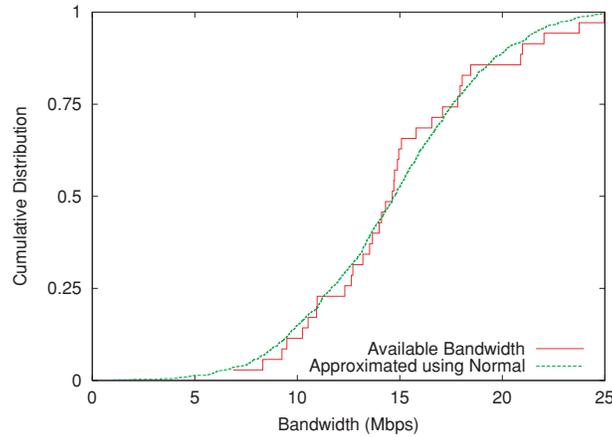


Fig. 10. Approximating available bandwidth using normal distribution.

inverse CDFs of the available bandwidth and approximated available bandwidth, and x is the CDF value, such as 0.25 and 0.75. Thus, a positive relative error denotes an underapproximation, while a negative error denotes an overapproximation.

As shown in Table I, the results confirm that using a normal distribution can closely approximate the available bandwidth. Moreover, the normal distribution tends to have a lower value than the available bandwidth distribution for the same CDF value in the region from zero to the median. Since BROS sensibly only selects a streaming rate lower than the median available bandwidth, this implies that using a normal distribution instead of the sampled available bandwidth distribution results in a conservative estimation of the rate selected, which is helpful to avoid buffer underflow.

Figure 10 provides a closer look at the effectiveness of using a normal distribution to approximate the available bandwidth for the case of rate adaptation. Even though rate adaptation has link capacity changes in fixed intervals, the interactions with the MAC layer retries causes the available bandwidth to closely follow a normal distribution. Thus under conditions of low signal strength, it may take multiple retries to transmit at a higher data rate, while taking fewer retries to transmit at a lower data rate, thus “smoothing” the fixed-capacity steps to follow a normal distribution.

5. EVALUATION

5.1 Wireless Testbed

The Buffer and Rate Optimization for Streaming (BROS) algorithm is implemented and evaluated using an IEEE 802.11 wireless testbed. As shown in Figure 11, the wireless testbed consists of a streaming server (wbestserver) that performs bandwidth estimation and streams video to a client, a traffic server (tgserver) that generates crossing and contending traffic, a wireless AP, and three clients (Clients

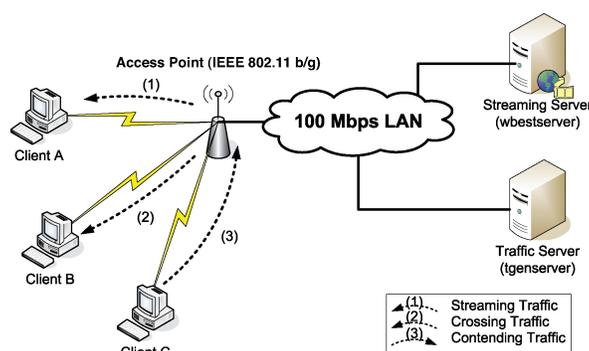


Fig. 11. Network path with last mile wireless network.

A, B, and C). The AP in the testbed is a Cisco Air-AP1121G⁴ supporting IEEE 802.11b/g. Both servers are PCs with P4 3.0 GHz CPUs and 512MBytes RAM and the three clients are PCs with P4 2.8 GHz CPUs and 512MBytes RAM. All the testbed PCs run SUSE⁵ 9.3 Linux with kernel version 2.6.11. The servers connect to the AP with a wired 100Mbps LAN, and the clients connect to the AP with a IEEE 802.11b/g WLAN using Allnet⁶ ALL0271 54Mbps wireless PCI card with a prism GT chipset.⁷

As labeled in Figure 11, Client A is configured as the streaming client to receive streaming traffic (1) from the streaming server. Client B, Client C, and the traffic server generate crossing traffic (2) or contending traffic (3) using the Multi-Generator Toolset⁸ (mgen) v4.2b6 over UDP. Client B or Client C is also configured as a wireless sniffer⁹ to monitor the traffic in the testbed when not being used as a traffic generator.

5.2 Emulated Streaming (EmuS) Client/Server System

To evaluate BROS, we developed an emulated streaming client-server system, called *Emulated Streaming (EmuS)*, with initial buffer and rate selection features. EmuS has features common to many commercial streaming systems, including multiple encoded video layers, configurable initial buffer size, and an RTSP-like communication mechanism. EmuS uses two channels, a control channel running over TCP and a data channel running over UDP. Even without a real media codec, EmuS provides framing and packetization functionalities to emulate a real streaming system. EmuS uses CBR encoding with equal-sized frames and a fixed-size smoothing buffer. While actual video is typically VBR, smoothing buffers are often used at the sender to avoid packet bursts or at the receiver to aid decoding. The client detects frame losses, based on out-of-order sequence numbered frames and sends NACK messages back to the server asking for retransmission until the frame is too late to be played out. The client also applies frame repair, which repeats the last good frame for each lost frame to mitigate the impact of frames loss. When buffer underflow occurs, the client stops playback and starts rebuffering frames from the server until the playout buffer has filled. Once the buffer is filled, video playback resumes. The client records statistical information that includes buffer underflow counts, frame rate, frame losses, and packet retransmission counts.

⁴<http://www.cisco.com/en/US/products/hw/wireless/ps4570/index.html>

⁵<http://www.novell.com/linux/>

⁶<http://www.allnet-usa.com/>

⁷<http://www.conexant.com/products/entry.jsp?id=885>

⁸<http://pf.itd.nrl.navy.mil/mgen/>

⁹<http://perform.wpi.edu/wsniiffer/>

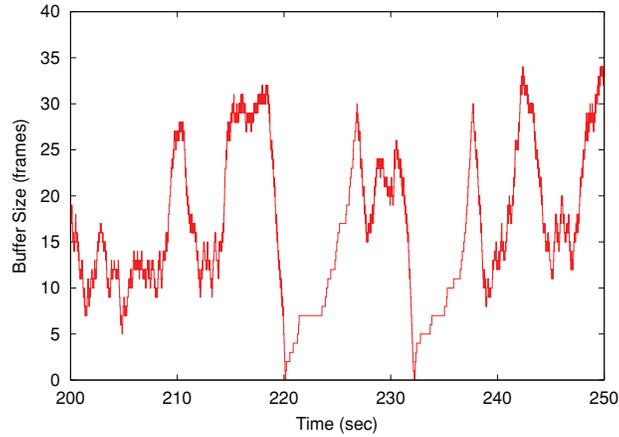


Fig. 12. Typical buffer size fluctuation.

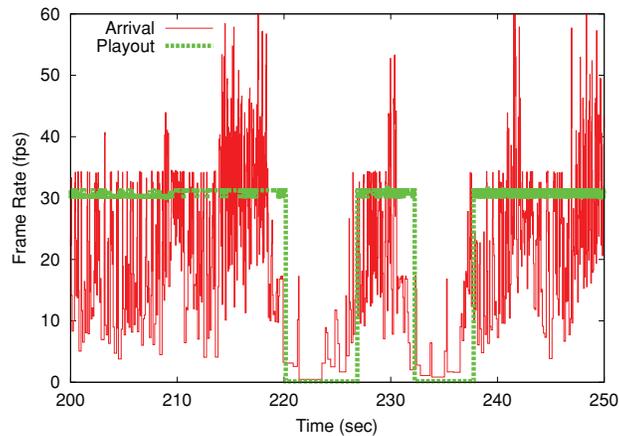


Fig. 13. Frame arrival and playout.

Figure 12 and Figure 13 depict the actual buffer occupancy, and arrival/playout frame rates, respectively, recorded for a typical EmuS streaming session, shown for a 50-second time slice starting about 2.5 minutes after the session starts. Note that the buffer occupancy fluctuates in Figure 12 in response to fluctuations in the available bandwidth, evidenced by the fluctuations in the arrival frame rate in Figure 13. As long as the playout buffer remains above 0, the frame playout rate in Figure 13 remains at 30 frames per second. However, at time 220 seconds, the buffer drops to zero (a buffer underflow event) and frame playout stops until the buffer is filled at about time 227. At time 232, the buffer again drops to zero and frame playout stops until the buffer fills at about time 239.

As discussed in Section 4, a proper MTBBU value needs to be indicated by a client in order to compute the buffer size based on streaming rate and vice versa. As shown in Figure 9, as long as the streaming rate is kept under 0.485 of the available bandwidth, a modest increase in buffer size can result in a significantly increased MTBBU. In light of this, for all experiments, an extremely low target MTBBU of $\pi_0 = 1 \times 10^{-16}$ is used based on the underflow probability in Eq. (8). In practice, for future experiments, both BROS (and EmuS) allow configuration of the targeted MTBBU to fit the application requirements.

Table II. Evaluation Setups

Case	Configuration
1. Idle channel	No other traffic $RSSI > -38dBm$ $mean = 6.4Mbps, stdev = 0.4Mbps$
2. Contending traffic	1.2 Mbps UDP: Client C to tgenserver $RSSI > -38dBm$ $mean = 4.7Mbps, stdev = 2.3Mbps$
3. Crossing traffic	1.2 Mbps UDP: tgenserver to Client B $RSSI > -38dBm$ $mean = 5.2Mbps, stdev = 1.9Mbps$
4. Rate adaptation	No other traffic, $-89dBm \leq RSSI \leq -85dBm$ $mean = 3.5Mbps, stdev = 1.6Mbps$

5.3 Experiment Design

The time-variant behavior of the available bandwidth can be dramatically affected by other traffic on the WLAN and the current wireless conditions. Thus, our specific experimental tests include contending and crossing traffic situations and a situation specifically to yield significant rate adaptation changes by the 802.11 layer. Running WBest first gathers the “macroscopic” nature of the current WLAN situation, capturing the impact on available bandwidth due to contending traffic, crossing traffic, and rate adaptation. The premise is that these large, macroscopic effects on time variation in available bandwidth outweigh any smaller, microscopic random time variation in wireless conditions.

Four test cases are executed to evaluate BROS running on EmuS in different wireless network setups. BROS is compared with a fixed buffer size, fixed streaming rate, and jitter-removal buffer algorithms.

The encoded layers of the streaming media used in our experiments follow the default multilayer encoding profile for Windows Media Encoder,¹⁰ which has 10 encoding layer rates from 28Kbps to 1.1Mbps, and 4 encoding rates for higher layers at 2.1, 3.6, 5.1, and 6.8Mbps. These encoding rates are meant to represent a “typical” set of encoding rates, but other rates could easily be substituted. All encoding layers have a frame rate of 30 frames per second. The video media length is 2 minutes, which is the median length of streaming video available on the Web [Li et al. 2005b]. The wireless testbed is set to 802.11b with an average available bandwidth of 6.4Mbps as measured by WBest. The four distinct test scenarios are: an idle channel (with good reception quality), a channel with crossing traffic, a channel with contending traffic, and a channel with frequent rate adaptations (with poor reception quality). The set up details are listed in Table II. For each test case, the streaming traffic is sent downstream from the wbestserver to Client A. The rate adaptation case is set up by manually reducing the transmission power of both the AP and Client A to generate rate adaptation events, as in Figure 5, and the accompanying MAC layer retries. Table II provides a summary view of the network conditions in that case, with the medians of the mean and standard deviation of the available bandwidth of all test runs for each case.

For each test case, the streaming sessions listed in Table III are run 30 times in sequence. First, the sessions with BROS are run with a maximum tolerable buffer size of 5 seconds. Second, the sessions with a 5-second fixed¹¹ buffer size¹² are run, first with a streaming rate higher than the mean available bandwidth (H), then with a streaming rate close to the mean available bandwidth (M), and lastly with

¹⁰<http://www.microsoft.com/windows/windowsmedia/default.msp>

¹¹As in most streaming systems, “fixed” does not refer to the amount of buffering time that elapses before streaming commences, but rather refers to the temporal amount of video data that is buffered before streaming commences.

¹²The default buffer size chosen by Windows Streaming Media is 5 seconds [Birney 2004].

Table III. Sessions for Each Case

Sessions		Case 1,2,3		Case 4	
		Rate(Mbps)	Buf(sec)	Rate(Mbps)	Buf(sec)
BROS		Auto	Auto	Auto	Auto
Fixed buffer	H	6.8	5	5.1	5
	M	5.1	5	3.6	5
	L	3.6	5	2.1	5
Jitter buffer	H	6.8	B_{jit}	5.1	B_{jit}
	M	5.1	B_{jit}	3.6	B_{jit}
	L	3.6	B_{jit}	2.1	B_{jit}

a streaming rate less than the mean available bandwidth (L). Third, traces for the fixed buffer tests are used to determine the performance of the jitter-removal buffer algorithm.

The jitter removal buffer B_{jit} is computed using an approach similar to that discussed in Mundur et al. [1999]. Most jitter-removal buffer algorithms propose variable sampling and averaging algorithms to estimate the network jitter. However, since the frame arrival traces for the entire session are used, no jitter prediction is actually needed. Instead, jitter is measured as the difference between the 95th percentile and 5th percentile of frame interarrival times, denoted by $D_{.95} - D_{.05}$. Therefore, the buffer size for jitter removal B_{jit} is computed as

$$B_{jit} = \frac{1}{\mu} \left[\frac{D_{.95} - D_{.05}}{1/\mu} + 1 \right], \quad (11)$$

where μ is the frame rate. With the traces of frame arrival data gathered from the fixed buffer sessions, the jitter-removal buffer algorithm is applied to decide the buffer size needed to remove the jitter for each session. Then the session is trace driven using the frame arrival data with B_{jit} .

5.4 Performance Metrics

To assess the quality of a streaming video system as perceived by a user, there are many factors that need to be taken into consideration. Since determining perceived quality through subjective user studies is often impractical, many objective video quality metrics have been developed to estimate the quality of streaming video. Peak Signal to Noise Ratio (PSNR) is commonly used to assess the visual quality of encoded video. However, PSNR does not effectively represent perceived quality during frame loss and in no way represents perceived quality for network delay or for rebuffering events at the client. Other metrics less commonly used include the Video Quality Metric (VQM) [Pinson and Wolf 2004], Moving Pictures Quality Metric (MPQM) [van den Branden Lambrecht and Verscheure 1996], Structural Similarity Index Measurement (SSIM) [Wang et al. 2002], and Noise Quality Measure (NQM) [Damera-Venkata et al. 2000]. Like PSNR, these quality metrics are primarily designed to assess the visual quality of the encoding/decoding system and do not effectively represent degraded quality due to startup delays or rebuffering events.

Within EmuS, video quality is evaluated using the following relevant metrics:

- (1) *Initial Buffer Delay*: the number of seconds video frames are queued at the client before playout. While the delay from the server to the client also includes latency plus queuing along the network path, this network delay is not controllable by most streaming systems. Moreover, our experiments are on a LAN with small (typically well under 10 milliseconds) of delay. Hence, delay is dominated by the playout buffer on the client.
- (2) *Total Buffer Delay*: the total number of seconds video frames are queued at the client during the playout session. The total buffer delay includes the initial buffer delay and the delay caused by

rebuffering after buffer underflow events. Total buffer delay defines the total time that playout stops for the whole playout session.

- (3) *Buffer Underflow*: the fraction of time the playout buffer reaches zero, causing the client to stop playing frames. When a buffer underflow occurs, there is a significant interruption in the playback since the frame playout stops until the playout buffer is completely refilled.
- (4) *Frame Loss*: the percentage of frames that are not played by the client. Frame loss primarily happens when frames are lost in the network and are not retransmitted before the following frame is ready to be played.
- (5) *Streaming Rate*: the rate the server sends data to the client. The streaming rate directly correlates to the encoding bitrate used to compress the raw video frames. While PSNR could be used to assess the quality of the encoded bitrate, since all comparisons in this work use the same codec, the streaming rates used can be directly compared.

An adaptive streaming system tries to maximize the streaming rate, while minimizing frame loss, initial, and total buffer delay, and buffer underflow. Thus, there are trade-offs in that a higher streaming rate (resulting in better encoding quality) will have a higher delay (to remove jitter), more frame losses, and/or more buffer underflows and vice versa. Unfortunately, to the best of our knowledge there is no way to combine these individual metrics into a single, unified measure of quality. Instead, in our analysis, for a fixed streaming rate, the frame loss rate, initial and total buffer delay, and buffer underflows are compared, and for a given buffer size the frame loss rate, buffer underflows, and streaming rate are compared.

6. RESULTS

BROS is evaluated in four different wireless LAN scenarios running 120 two-minute video sessions and 90 trace-based sessions for each scenario. The total data collected includes more than 16 hours of streaming, recording buffer underflow events, frame arrival and playout rates, frame loss and buffer delays. BROS is evaluated by comparing the rate selection, buffer optimization results, and the related streaming quality metrics [Wang et al. 2003]: streaming rate, buffer underflow events, frame losses, and initial and total buffer delays.

Figure 14 depicts the rate selection results for the four test cases. The average streaming rate selected for the 120 sessions for each case is shown along with the standard deviation. With the input of bandwidth estimation information from WBest (shown in Table II), the BROS algorithm correctly selects the maximum available encoding rate lower than the available bandwidth. Moreover, BROS also takes into consideration the variance of the available bandwidth. For example, in the crossing traffic tests, even though the median available bandwidth is on average higher than the encoding rate of 5.1Mbps, BROS selects the next lower encoding rate, 3.6Mbps, because the variance in the available bandwidth causes the rate of 5.1Mbps to have a higher buffer underflow probability.

Figure 15 depicts the buffer optimization results, in comparison with the size of the jitter-removal buffer (the fixed-sized buffer can also be compared as always having a 5-second buffer size on the y -axis). With BROS, the initial buffer size is greatly reduced from a fixed-size, 5-second buffer to about 1 or 2 seconds. The jitter-removal buffers are substantially smaller, consistently less than 0.5 seconds even for the largest jitter-removal buffer. While a small buffer is attractive for reducing the startup delay for playout, the quality is also impacted by the resulting impact on frame loss rate (see next). In addition, BROS adjusts the buffer based on the variance in the available bandwidth (see Table II). Comparing the contending and crossing cases, even though the available bandwidth and the selected streaming are close to each other, the buffer sizes BROS chooses are different. The contending traffic

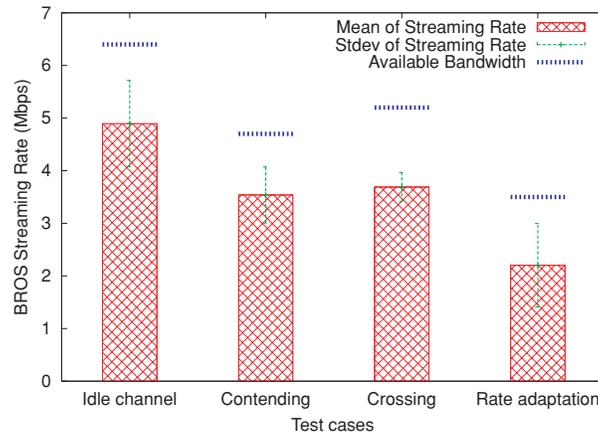


Fig. 14. BROS streaming rate.

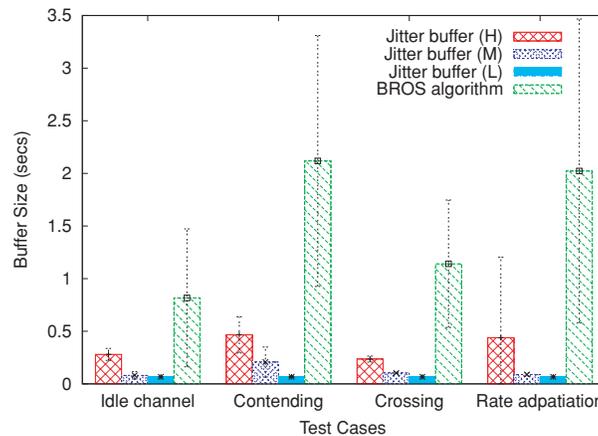


Fig. 15. BROS buffer size.

case has a higher average buffer size than the crossing traffic case because of the higher variance in available bandwidth.

Figures 16 through 19 provide the results for the streaming quality metrics for each case, showing the averages over all sessions of the fraction of buffer underflows, frame loss rates, and buffer delays. By combining the optimization of the streaming rate selection and the initial buffer size, BROS performs better than the manual rate selection and jitter buffer approaches in terms of buffer underflow, frame loss, and buffer delays. As a quick visual summary, lower is better for all metrics depicted and BROS is the lowest or nearly the lowest for all graphs shown.

The jitter buffer sessions always have a high fraction of buffer underflow events because of the small buffer sizes chosen. The fixed buffer size sessions with high (H) and medium (M) streaming rates also have high buffer underflow fractions since the streaming rates are generally too high to be consistently sustained for the duration of the video session. While the buffer underflow fractions for the low streaming rate sessions (L) with fixed buffer size have a buffer underflow fraction close to that of BROS, BROS has a total buffer delay reduced by more than 50%.

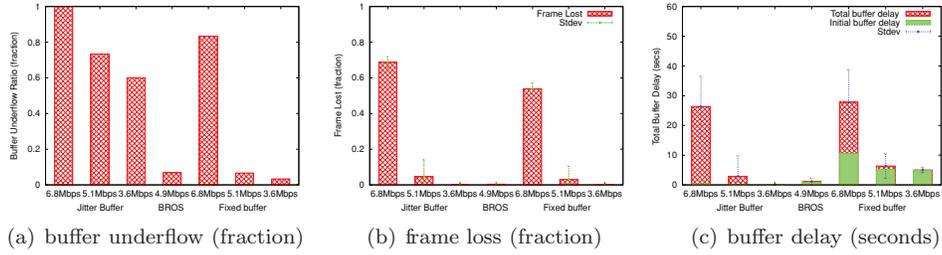


Fig. 16. Summary results for evaluation case 1 (Idle channel).

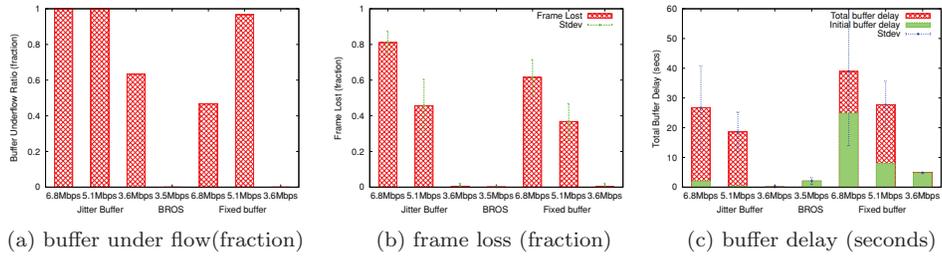


Fig. 17. Summary results for evaluation case 2 (contending traffic).

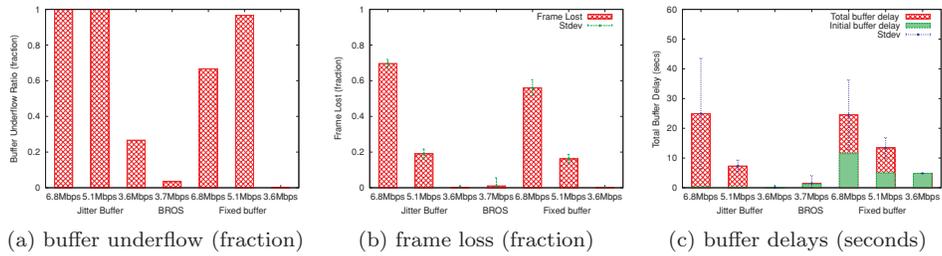


Fig. 18. Summary results for evaluation case 3 (crossing traffic).

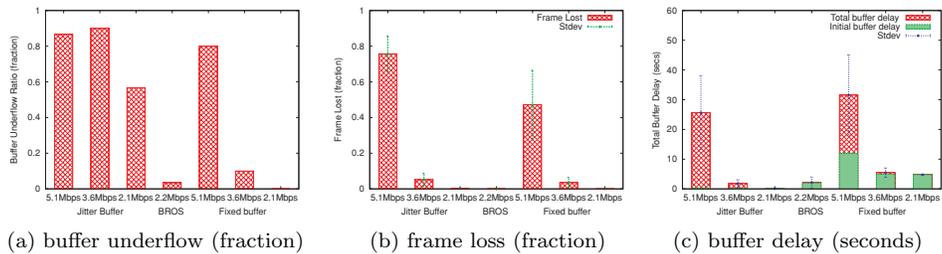


Fig. 19. Summary results for evaluation case 4 (rate adaptation).

Table IV. Buffer Underflow (fraction)

		Case 1	Case 2	Case 3	Case 4
BROS		0.07	0	0.03	0.03
Fixed buffer	H	0.83	0.47	0.67	0.80
	M	0.07	1.00	1.00	0.10
	L	0.03	0	0	0
Jitter buffer	H	1.00	1.00	1.00	0.87
	M	0.73	1.00	1.00	0.90
	L	0.60	0.63	0.26	0.56

Table V. Frame Loss Rate (percent)

		Case 1	Case 2	Case 3	Case 4
BROS		0.39%	0.16%	1.06%	0.05%
Fixed buffer	H	53.9%	61.5%	56.0%	47.2%
	M	3.00%	36.8%	16.2%	3.57%
	L	0.04%	0.32%	0.03%	0.01%
Jitter buffer	H	68.7%	81.0%	69.7%	75.7%
	M	4.66%	45.7%	19.1%	5.20%
	L	0.14%	0.37%	0.14%	0.09%

Table VI. Buffer Delay (seconds)

		Case 1		Case 2		Case 3		Case 4	
		Initial	Total	Initial	Total	Initial	Total	Initial	Total
BROS		0.82	1.06	2.06	2.06	1.13	1.50	1.95	2.12
Fixed buffer	H	10.6	27.8	24.9	38.9	11.5	24.6	11.9	31.6
	M	5.33	6.34	7.97	27.7	4.97	13.4	4.97	5.47
	L	4.85	5.02	4.85	4.85	4.84	4.84	4.79	4.79
Jitter buffer	H	0.31	26.4	1.99	26.7	0.26	24.9	0.32	25.7
	M	0.05	2.89	0.31	18.6	0.07	7.31	0.05	1.81
	L	0.05	0.06	0.03	0.08	0.03	0.04	0.03	0.03

Moreover, BROS has much lower frame losses than all the sessions with high (H) and medium (M) streaming rate with both-fixed size or jitter buffers. While the low (L) streaming rate sessions with fixed or jitter buffer have frame loss rates similar to BROS, BROS has a lower initial buffer delay than the fixed buffer size sessions and a lower buffer underflow than the jitter buffer sessions.

In general, BROS significantly reduces the average buffer delays for the fixed buffer size and jitter buffers, for both the initial buffer delay and total buffer delay. For instance, even though the jitter buffer sessions with high (H) and medium (M) streaming rates usually have a small initial delay due to the small buffer size, the total buffer delays are high because of the large number of buffer underflow events in each session. The jitter buffer sessions with low (L) streaming rate have the lowest buffer delay. However, the buffer underflow fraction is unacceptable for most streaming media applications.

Tables IV, V, and VI summarize the streaming quality results by comparing the averages for all the sessions in each case in terms of the buffer underflow fraction, frame loss percent, and buffer delays. BROS shows the best overall performance for the quality metrics.

Tables VII summarizes the streaming rate selected by BROS in comparison with the streaming rates used for evaluating fixed and jitter buffer sessions. The streaming rates selected by BROS are usually higher than the low streaming rate (cases 1, 3, and 4). This demonstrates that BROS preserves a reasonable streaming rate and a good image quality, while significantly reducing the buffer underflow fraction, frame loss percent, and buffer delays at the same time.

BROS significantly reduces the buffer underflow event, frame loss, and total buffer delay by selecting the appropriate initial streaming rate and buffer size corresponding to the available bandwidth

Table VII. Streaming Rate (Mbps)

		Case 1	Case 2	Case 3	Case 4
BROS		4.9	3.5	3.7	2.2
Fixed and Jitter Buffer	H	6.8	6.8	6.8	5.1
	M	5.1	5.1	5.1	3.6
	L	3.6	3.6	3.6	2.1

information gathered. For example, BROS effectively selects proper streaming rates in between the medium (M) and low (L) streaming rates and adjusts the buffer sizes to achieve a better streaming quality than fixed and jitter buffer cases. Comparing BROS with the medium (M) streaming rate for fixed and jitter-removal buffers, BROS effectively reduces the average buffer underflow fraction by 99% and 100%, frame loss rates by 97% and 98%, and total buffer delay by 87% and 78%, respectively. Comparing BROS with the low (L) streaming rate for a fixed buffer, BROS effectively reduces the total buffer delay by 55% to 83%, while maintaining a comparable low buffer underflow fraction and frame loss rate. Comparing BROS with the low (L) streaming rate for jitter buffer, even though BROS has slightly higher total buffer delays, BROS effectively reduces the buffer underflow fraction by 88% to 100%, and maintains comparable low frame loss rates.

7. CONCLUSION

This article proposes BROS, an algorithm designed to select the proper streaming rate and initial buffer size based on the available bandwidth estimations using WBest to reduce the buffer underflow events and initial and total buffer delays, and improve the frame loss rate for multimedia streaming application over wireless networks. A core contribution is a model of the client-side initial buffer size for streaming multimedia applications, as a function of streaming rate and the distribution of available bandwidth in the wireless networks. One advantage of the buffer model over existing jitter or Poisson arrival models is that it considers changes in available bandwidth which typically have a larger impact on streaming performance than interarrival jitter. For evaluation purposes, BROS is implemented in a streaming system (EmuS) and compared against streaming systems with a fixed streaming rate and buffer sizes, and jitter-removal buffers. EmuS involves a wireless testbed that facilitated evaluation under four distinct WLAN scenarios. The following conclusions can be drawn.

- (1) Existing buffer models that consider only the impact of jitter are not adequate to remove the effects of changes in available bandwidth in IEEE 802.11 networks. Moreover, the assumption of constant average arrival rate in previous research is not applicable in environments with volatility in available bandwidth such as IEEE 802.11 WLANs. This causes the current jitter buffer models to underpredict the buffer size needed to avoid buffer underflow in wireless networks.
- (2) Multimedia streaming application performance is significantly impacted by the initial streaming rate and buffer size. By selecting the streaming rate and playout buffer size based on current available bandwidth conditions, BROS provides better performance.
- (3) BROS currently uses the WBest algorithm to estimate the mean and standard deviations of available bandwidth to infer the available bandwidth distribution. However, BROS is flexible enough to be used with other bandwidth estimation tools that provide similar bandwidth information. This makes it easy to improve the bandwidth estimation techniques independently of the BROS algorithm.

Overall, BROS can significantly reduce buffer underflows, frame losses, and buffer delays by optimizing the initial streaming rate and buffer size.

8. FUTURE WORK

Strong interference or mobility may still impact the available bandwidth during streaming sessions, thus resulting in unexpected buffer underflow events. Therefore, to further improve streaming rate selection and buffer optimization for the entire streaming sessions in an unsteady environment, BROS can be applied periodically during playback or triggered by each rebuffer event. However, adjustments to streaming rates and buffer sizes would need to be done with care since such changes have negative effects on perceived quality [Zink et al. 2003]. One possible area of future work would be to use the streaming multimedia data to estimate available bandwidth during the session. This can reduce the traffic overhead caused by WBest probing.

BROS can be further improved to optimize application-layer media repair approaches, such as Forward Error Correction (FEC) and packet retransmissions. For example, the available bandwidth information can be used to decide the level of FEC or number of streaming packet retries. For the streaming sessions that are constrained by available bandwidth, limiting the amount of FEC and retransmission traffic can reduce the packet delay or loss caused by media repair procedures.

Currently, BROS suggests not to stream when WBest reports no available bandwidth in the wireless networks. However, since the wireless network is a contention domain, streaming applications can still get the throughput of a fair share of the effective capacity. Therefore, instead of suggesting not to stream, BROS could be further improved to decide the streaming rate inferred as a function of the fair share of the effective capacity, perhaps utilizing cross-layer knowledge of the number of contending nodes in the wireless network.

REFERENCES

- ANGRISANI, L., BOTTA, A., PESCAPE, A., AND VADURSI, M. 2006. Measuring wireless links capacity. In *Proceedings of the 1st International Symposium on Wireless Pervasive Computing*, 1–5.
- BAL, G. AND WILLIAMSON, C. 2004. The effects of mobility on wireless media streaming performance. In *Proceedings of Wireless Networks and Emerging Technologies (WNET)*, 596–601.
- BIRNEY, B. 2004. Reducing start-up latency with Windows Media 9 series. Microsoft Online Documents.
- BOLOT, J.-C. AND TURLETTI, T. 1994. A rate control scheme for packet video in the internet. In *Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies Infocom*, 1216–1223.
- CEN, S., COSMAN, P. C., AND VOELKER, G. M. 2003. End-to-end differentiation of congestion and wireless losses. *IEEE/ACM Trans. Netw.* 11, 5(1), 703–717.
- CHEN, M. AND ZAKHOR, A. 2005. Rate control for streaming video over wireless. *IEEE Wirel. Comm.* 12, 4, 32–41.
- CONKLIN, G. J., GREENBAUM, G. S., LILLEVOLD, K. O., LIPPMAN, A. F., AND REZNIK, Y. A. 2001. Video coding for streaming media delivery on the Internet. *IEEE Trans. Circ. Syst. Video Technol.* 11, 3, 269–281.
- DAMERA-VENKATA, N., KITE, T. D., GEISLER, W. S., EVANS, B. L., AND BOVIK, A. C. 2000. Image quality assessment based on a degradation model. *IEEE Trans. Image Process.* 9, 4, 636–650.
- DELGROSSI, L., HALSTRICK, C., HEHMANN, D., HERRTWICH, R. G., KRONE, O., SANDVOSS, J., AND VOGT, C. 1993. Media scaling for audiovisual communication with the Heidelberg transport system. In *Proceedings of the 1st ACM International Conference on Multimedia*, 99–104.
- DEMIRCI, M. U. AND VAN BEEK, P. 2005. Bandwidth estimation and robust video streaming over 802.11E wireless lans. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 1250–1253.
- FENG, W.-C. AND SECHREST, S. 1995. Smoothing and buffering for delivery of prerecorded compressed video. In *Proceedings of IST/SPIE Multimedia Networking and Computing Conference*, 234–242.
- FLOYD, S., HANDLEY, M., PADHYE, J., AND WIDMER, J. 2000. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM Conference*, 43–56.
- FUJIMOTO, K., ATA, S., AND MURATA, M. 2002. Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications. In *Proceedings of the IEEE Conference and Exhibition on Global Telecommunications (GlobeCom)*. vol. 3, 2451–2457.
- HSU, C.-Y., ORTEGA, A., AND KHANSARI, M. 1999. Rate control for robust video transmission over burst-error wireless channels. *IEEE J. Select. Areas Comm.* 17, 5, 756–773.

- JACOBS, S., AND ELEFThERiADiS, A. 1996. Providing video services over networks without quality of service guarantees. In *Proceedings of the World Wide Web Consortium Workshop*.
- KALMAN, M., STEiNBACH, E., AND GiROD, B. 2004. Adaptive media playout for low delay video streaming over error-prone channels. *IEEE Trans. Circ. Syst. Video Technol.* 14, 6, 841–851.
- KAZANTZiDiS, M. AND GERLA, M. 2003. The impact of link layer assisted multimedia adaptation in wireless network. In *Proceedings of the IEEE International Conference on Information Technology, Research and Education (ITRE)*, 326–330.
- KOHLER, E., HANDLEY, M., AND FLOYD, S. 2006. Designing DCCP: Congestion control without reliability. In *Proceedings of ACM SIGCOMM Conference*, 27–38.
- KUANG, T. AND WILLIAMSON, C. L. 2004. Hierarchical analysis of real-media streaming traffic on an IEEE 802.11b wireless LAN. *Comput. Comm.* 27, 6, 538–548.
- LAKSHMINARAYANAN, K., PADMANABHAN, V. N., AND PADHYE, J. 2004. Bandwidth estimation in broadband access networks. In *Proceedings of the Internet Measurement Conference (IMC 2004)*, 314–321.
- LAOUTARIS, N. AND STAVRAKAKIS, I. 2001. Adaptive playout strategies for packet video receivers with finite buffer capacity. In *Proceedings of the IEEE International Conference on Communications (ICC)*. Vol. 3, 969–973.
- LI, F., CHUNG, J., LI, M., WU, H., CLAYPOOL, M., AND KiNiCKi, R. 2005a. Application, network and link layer measurements of streaming video over a wireless campus network. In *Proceedings of the 6th Passive and Active Measurement Workshop (PAM)*, 189–202.
- LI, K., KRASiC, C., WALPOLE, J., SHOR, M. H., AND PU, C. 2001. The minimal buffering requirements of congestion controlled interactive multimedia applications. In *Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems (IDMS '01)*, 181–192.
- LI, M., CLAYPOOL, M., AND KiNiCKi, R. 2008. WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks. In *Proceedings of the 33rd IEEE Conference on Local Computer Networks (LCN)*.
- LI, M., CLAYPOOL, M., KiNiCKi, R., AND NiCHOLS, J. 2005b. Characteristics of streaming media stored on the Web. *ACM Trans. Internet Technol.* 5, 4, 601–626.
- LI, M., LI, F., CLAYPOOL, M., AND KiNiCKi, R. 2005c. Weather forecasting: Predicting performance for streaming video over wireless LANs. In *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 33–38.
- LI, Q. AND VAN DER SCHAAR, M. 2004. Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation. *IEEE Trans. Multimedia* 6, 2, 278–290.
- LIN, S., COSTELLO, D. J., JR., AND MILLER, M. J. 1984. Automatic-repeat-request error-control schemes. *IEEE Comm. Mag.* 22, 12, 5–17.
- MOON, S. B., KUROSE, J., AND TOWSLey, D. 1998. Packet audio playout delay adjustment: Performance bounds and algorithms. *Multimedia Syst.* 6, 1, 17–28.
- MUNDUR, P., SOOD, A., AND SIMON, R. 1999. Network delay jitter and client buffer requirements in distributed video-on-demand systems. Tech. rep., Department of Computer Science at George Mason University, Fairfax, VA.
- PiNson, M. AND WOLF, S. 2004. A new standardized method for objectively measuring video quality. *IEEE Trans. Broadcast.* 50, 3, 312–322.
- PRASAD, R., MURRAY, M., DOVROLIS, C., AND CLAFFY, K. 2003. Bandwidth estimation: Metrics, measurement techniques, and tools. *IEEE Netw.* 17, 6, 27–35.
- RAMJEE, R., KUROSE, J., TOWSLey, D., AND SCHULZRIiNNE, H. 1994. Adaptive playout mechanism for packetized audio applications in wide-area networks. In *Proceedings of the IEEE Infocom*, 680–688.
- STEiNBACH, E., FAERBER, N., AND GiROD, B. 2001. Adaptive playout for low-latency video streaming. In *Proceedings of the International Conference on Image Processing (ICIP)*, 962–965.
- STOCKHAMMER, T., JENKAC, H., AND KUHN, G. 2004. Streaming video over variable bit-rate wireless channels. *IEEE Trans. Multimedia* 6, 2, 268–277.
- VAN BEEK, P. AND DEMiRCiN, M. U. 2005. Delay-constrained rate adaptation for robust video transmission over home networks. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. Vol. 2, 173–176.
- VAN BEEK, P., DESHPANDE, S., PAN, H., AND SEZAN, I. 2004. Adaptive streaming of high-quality video over wireless LANs. In *Proceedings of the Conference on Visual Communications and Image Processing*, 647–660.
- VAN DEN BRANDEN LAMBRECHT, C. J. AND VERSCHEURE, O. 1996. Perceptual quality measure using a spatio-temporal model of the human visual system. In *Proceedings of SPIE*, 450–461.
- WANG, Z., BANERJEE, S., AND JAMIN, S. 2003. Studying streaming video quality: From an application point of view. In *Proceedings of the 11th ACM International Conference on Multimedia*, 327–330.

- WANG, Z., LU, L., AND BOVIK, A. 2002. Video quality assessment using structural distortion measurement. In *Proceedings of the IEEE International Conference on Image Processing*. Vol. 3, 65–68.
- YANG, F., ZHANG, Q., ZHU, W., AND ZHANG, Y.-Q. 2004. End-to-end TCP-friendly streaming protocol and bit allocation for scalable video over wireless internet. *IEEE J. Select. Areas Comm.* 22, 4, 777–790.
- YANG, G., GERLA, M., AND SANADIDI, M. Y. 2004. Adaptive video streaming in presence of wireless errors. In *Proceedings of IFIP/IEEE Management of Multimedia Networks and Services (MMNS)*, 26–38.
- YUANG, M., LIANG, S., CHEN, Y., AND SHEN, C. 1996. Dynamic video playout smoothing method for multimedia applications. In *Proceedings of the IEEE International Conference on Communications (ICC)*. vol. 3, 1365–1369.
- YUANG, M. C., TIEN, P. L., AND LIANG, S. T. 1997. Intelligent video smoother for multimedia communications. *IEEE J. Select. Areas Comm.* 15, 2, 136–146.
- ZINK, M., KUNZEL, O., SCHMITT, J., AND STEINMETZ, R. 2003. Subjective impression of variations in layer encoded videos. In *Proceedings of the 11th International Workshop on Quality of Service (IWQoS)*, 137–154.

Received November 2006; revised August 2008; accepted August 2008