Adaptive streaming of high-quality video over wireless LANs

Peter van Beek*, Sachin Deshpande, Hao Pan, and Ibrahim Sezan
Sharp Laboratories of America, 5750 NW Pacific Rim Blvd., Camas, WA 98607

ABSTRACT

We address the problem of robust streaming of high-quality video over wireless local area networks in a home environment. By robust streaming, we mean maintaining the highest possible video quality and preventing interruptions to the video under varying bandwidth conditions, which may be due to distance, interference, obstructions, and existence of multiple streams. We propose an application-layer approach where we provide algorithms for dynamic on-line network bandwidth estimation and dynamic on-line adaptation of video rate according to the available network bandwidth. The proposed system employs a packet scheduler, and a video rate control and adaptation mechanism at the sender, and bandwidth measurement and feedback mechanisms at the receiver. Our bandwidth estimation approach uses the actual video data in real time by transmitting it in packet bursts; hence, separate test traffic is not required. Since the proposed method operates at the application layer, it is flexible and applicable to different local area network types and implementations. We propose an extension to multiple streams by providing an algorithm for joint rate allocation to multiple video streams over a network enabling network-adaptive simultaneous streaming of high-quality video.

Keywords: Video streaming, wireless network, LAN, transcoding, transrating, bit allocation, bandwidth estimation.

1. INTRODUCTION

Various types of network technologies are being successfully deployed for use in office and home environments, and are being considered for streaming video data to consumer display devices. Wireless local area network (WLAN) technologies are advantageous due to their relatively high bit rates, relatively good range, ease of deployment (no new wiring required), availability of standards,[1] their rapid adoption in the marketplace, and industry support. Other technologies, such as power-line networks, seek to utilize existing wiring in the home for transport of digital data and likewise benefit from ease of deployment and availability of standards.[2]

The bandwidth available at the application layer for transport of data over wireless links often varies with time and is inherently unpredictable due to various factors. WLANs based on IEEE 802.11 can operate at several data link rates. The 802.11b standard operates at 1, 2, 5.5 or 11 Mbps, while the 802.11a standard operates at 6, 9, 12, 18, 24, 36, 48 or 54 Mbps.[1] The recently finalized 802.11g standard offers the high rates of 802.11a as well as backward compatibility with 802.11b rates. However, actual WLAN data throughput as seen by network layers above the data link layer is significantly lower due to the overhead of the 802.11 protocol, as well as other networking protocols used (e.g. IP and UDP). Furthermore, many other factors contribute to a lowering of actual bandwidth obtained in practice, such as the distance between the transmitter and receiver, the presence of walls and other structures, mobility of the transceivers, and RF interference due to the use of cordless phones and microwave ovens that operate in the same spectrum band. WLANs based on IEEE 802.11 include efficient error detection and correction techniques at the physical layer (PHY) and medium access control layer (MAC). High packet error rates (at the PHY layer) may cause sending stations to switch to lower data link rates, effectively reducing the packet loss rate (at the MAC layer), while decreasing the bandwidth available to higher protocol layers. Also, such systems utilize acknowledgement packets and will retransmit packets that are believed to be lost. Such retransmissions by the sender again effectively reduce the inherent error rate of the medium, at the cost of lowering bandwidth available to the application.

A case study[3] indicates that the actual throughput of an IEEE 802.11a system in an office environment is only about 23 Mpbs at 25 feet, and falls below 20 Mbps (approximately the rate of a single high-definition video signal) at ranges over 70 feet. The maximum throughput of an 802.11b system is barely 6 Mbps and falls below 6 Mbps (approximately the rate of a single standard-definition video signal at DVD quality) at ranges over 25 feet. Other case studies that include

* E-mail: pvanbeek@sharplabs.com

measurements of actual WLAN performance,[4,5] as well as studies based on theoretical models and simulations,[6] show that the bandwidth offered by 802.11 WLANs is in general significantly lower than the data link rates, and depends strongly on product implementation, network setup, network load, interference and other factors, therefore highly unpredictable.

This inherent variability of WLAN networks, the emergence of wireless mobile video clients on such networks, the existence of a variety of interference sources, multiple clients, and the unpredictability of the home environment in general, presents a challenge for the consumer electronics industry for streaming high-quality video over wireless home networks.

In this paper, we address robust streaming of high-quality video over wireless LANs. Although much work has been reported in literature on wireless video,[7] most research to date has focused on the transmission of low bit rate video over mobile cellular networks.[8,9] Much work has also been done on video streaming over lossy packet networks such as the Internet,[10,11,12] again mostly at relatively low bit rates. Our focus is on streaming high bit rate video over wireless LANs such as IEEE 802.11, for example standard-definition MPEG-2 video at approximately 6 Mbps. We also consider simultaneous transmission of multiple high-quality video streams over such networks, for example from a home server or gateway to several display devices in different rooms of a home. Very little work has been reported in this area to date.[13,14]

We propose a feedback-based approach to address the time-varying nature of the channel and the unpredictability of available bandwidth. We provide solutions for dynamic video rate adaptation and dynamic on-line measurement of the channel bandwidth. The bit rate of the video is adapted continuously at the sender based on feedback information from the receiver about the available channel bandwidth. We describe a bandwidth measurement algorithm that uses the intrinsic video traffic by sending it in bursts of packets. Bandwidth is estimated by measuring the arrival times of the burst packets at the receiver, and fed back to the sender. For dynamic on-line rate adaptation, we consider MPEG-2 video and describe a transrater whose output rate is controlled on the basis of the feedback from the receiver.

We also provide an algorithm for joint bit allocation and rate control for multiple video streams for network-adaptive joint streaming over a wireless network. Joint bit allocation and coding of multiple video streams has been addressed for the case of regular constant bit rate channels.[15,16] However, the channel characteristics in the case of 802.11 networks are time-varying and unique in the sense that different bandwidths may be available to different devices all sharing a single wireless channel. Using a simple model of such channels, we extend an existing technique for joint bit allocation to multiple video streams.

The following section presents an overview of our proposed system. Section 3 discusses video rate adaptation for a single stream for a point-to-point application. Section 4 presents a novel technique for dynamic on-line bandwidth estimation. A channel model for multi-stream wireless video transmission, and a joint rate allocation and control algorithm are presented in Section 5. Experimental results are provided in Section 6.

2. SYSTEM OVERVIEW

We address high-quality and robust streaming of video data from a single source device, for example a home server, to one or more receiving display devices, over a local area network, in particular an IEEE 802.11 WLAN. Figure 1 contains a diagram showing the relevant components of a system for transmitting multiple video streams following our approach.

The server part of the system may have either uncompressed video inputs (e.g., from analog) or compressed video inputs. Each video input is encoded or transcoded. Each (re-)encoded video stream is transmitted over a shared channel to a client device, where it is decoded and displayed. For simplicity, the diagram in Figure 1 shows only one client device, although the system actually includes multiple (one client per stream). Both client and server are connected to the network by network interfaces, which may include such components as packetizers, depacketizers and buffering, and implement the various networking protocol layers. The home network, in this case, consists of multiple point-to-point connections, all originating from the server, and carried over a single shared wireless link.

The approach towards robust audio/video transmission proposed in this paper falls into the class of feedback-based joint source-channel coding approaches,[7,17] in which the bit rates of the video streams transmitted are continuously adapted based on information about the channel conditions. To this end, the client part of the system (see Figure 1) includes a bandwidth monitoring module, which estimates the time-varying bandwidth on an ongoing basis. Bandwidth estimates are transmitted back to the sender, which dynamically adapts the bit rates of the various video streams accordingly. The goal is to avoid severe losses of visual quality that would be incurred if video data were transmitted at bit rates that can not be supported by the channel. Lost packets with video data may obviously lead to significant errors in the corresponding video frame, followed by errors in subsequent frames due to interframe error propagation.[7] By decreasing the video bit rate in a controlled manner when necessary, maximum visual quality is retained at all times. Rate adaptation is achieved by bit allocation and rate control at the encoder. We mainly consider the case where the input video is already in compressed format, requiring a transcoder[18] at the sender. Much work on adaptive video in literature has relied on the use of scalable (or layered) video coding as an alternative to transcoding.[10,12,26] However, scalable video coding often is not as bit-efficient as single layer coding, and sometimes lacks sufficient scaling granularity. Also, scalable video coding methods have generally not been adopted by international video coding standards yet.

In Figure 1, V_n denotes input video stream n, where $n = 1, 2, \ldots, N$. R_n denotes the bit rate of input stream n (only meaningful if this stream is in compressed form), while R_n' denotes the bit rate of stream n at the output of its encoder or transcoder. Each video stream is transcoded separately, although their bit rates are jointly controlled by a single dynamic rate adaptation module. This joint rate control module ensures that the aggregate bit rates of transmitted video streams can be supported by the channel, while optimizing the quality across all streams. Such joint optimization depends primarily on the characteristics of the video data, but may also include the use of high-level priorities or weights assigned to different streams. Such priorities may be related to the preferences of the users, or to the capabilities of the client devices. For example, one display device may be assigned a higher priority over other display devices, because it is located in the living room, or because it has a large screen (rendering any distortions more visible).

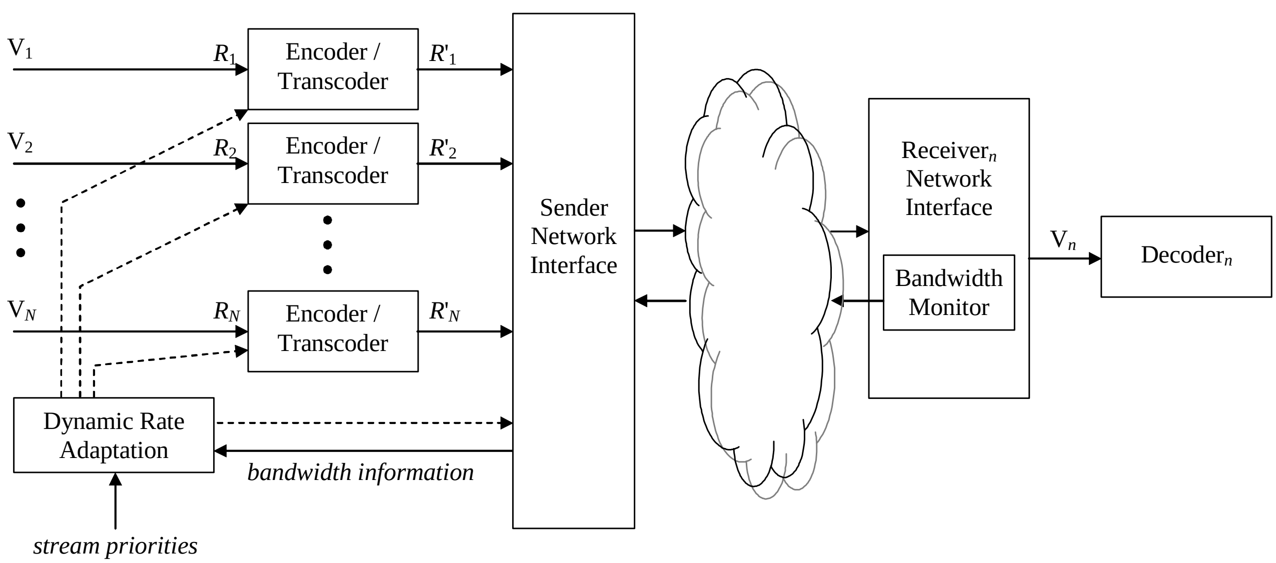


Figure 1. Overview of multi-stream video transmission system.

We discuss the details of video transcoding and dynamic rate adaptation for single streams in section 3. We discuss the details of the bandwidth monitor and estimation module in section 4. We discuss joint rate adaptation for multiple video streams in section 5.

3. VIDEO RATE ADAPTATION

A transcoder converts an existing digitally compressed audio or video bit stream into another bit stream to meet constraints that were not known when the original stream was encoded.[18] In particular, transcoders may be used to reduce the bit rate of a compressed bit stream, in order to match the transmission, storage or processing capabilities of specific networks or display devices. To achieve a reduction in the bit rate, a video transcoder can increase the level of compression applied to the video data, for example by increasing the quantizer step size. A transcoder may also reduce the temporal frame rate of the input video, reduce the spatial resolution, or convert the compression format. We refer to transcoding that involves only bit rate reduction as *transrating*. Various efficient transcoder architectures have been proposed that minimize computational complexity while retaining video quality.[18,19,20,21] In this paper, we have used a so-called re-quantization transrater with a simplified open-loop architecture,[19,21] illustrated in Figure 2. In this transrater, the input bit stream is first variable length decoded (VLD), and quantized transform coefficients are inverse quantized (IQ). Subsequently, these coefficients are directly re-quantized (Q) under the control of a bit allocation / rate control module, and variable length coded again (VLC). Most header data, macroblock motion vectors and other macroblock data is passed unchanged from the input to the output. The architecture is highly efficient in terms of storage and processing complexity, as there is no need for a frame memory, and motion estimation, motion compensation, forward and inverse transforms are not used. The open-loop transrater is subject to loss of visual quality due to drift errors, which can be controlled by the periodic insertion of I-frames. Such I-frames are often present already, for example, given the group-of-pictures (GOP) structure of MPEG-2 bit streams. Our goal in this paper is not to improve upon existing transcoder architectures; rather, we focus on the problems of bit allocation and rate control.

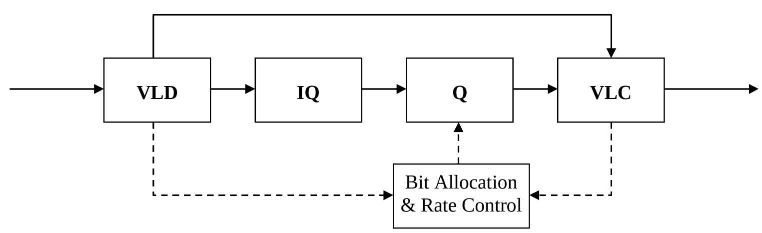


Figure 2. Open-loop re-quantizing transrater architecture.

Our approach to bit allocation and rate control is to simply scale the amount of bits spent on each frame by a given factor.[19] In the case of bit rate reduction from a constant bit rate input to a constant bit rate output, this constant factor is given by: $\beta = R'/R$, where R' is the desired output bit rate and R is the known bit rate of the input bit stream. As illustrated in Figure 2, the bit allocation module analyzes the input bit stream, counting the number of bits spent on each frame, and the number of bits spent on coding the AC transform coefficients for each macroblock. This creates a profile of bits spent on coefficients across the input frame, which is subsequently scaled, providing a target profile of the number of bits that should be spent during re-quantization to produce the output frame.[19] Subsequent rate control is similar to that used in the MPEG-2 Test Model 5, making adjustments to the quantizer step size to ensure the target is achieved.[22] We present constant bit rate MPEG-2 video transrating results using this approach in section 6.

This straightforward approach to bit allocation is especially useful when the bit allocation used in the original encoding was of high quality. Another advantage is that the transrater does not need to know the GOP structure of the input bit stream, which is not directly available as a parameter in the bit stream. The approach can be improved upon by optimizing the distribution of bits across pictures of I, P and B type.[23] The optimal distribution of bits across I, P and B pictures is known to be different for different output bit rates.

We have extended this transrater further, to allow the output bit rate to be adapted dynamically on the basis of information about available channel bandwidth. We allow dynamic rate adaptation on a frame-by-frame basis, and as a result, the scaling factor is allowed to vary from frame to frame. We present results of dynamic rate adaptation of MPEG-2 video using this transrater in section 6.

4. CHANNEL BANDWIDTH ESTIMATION AND FEEDBACK

This section discusses measuring and monitoring available channel bandwidth during transmission of audio/video data. Measurements of the bandwidth are used to dynamically adapt the bit rate of compressed bit streams transmitted. Relevant work in this area has focused mainly on measurements of Internet traffic.[12,24,25,26,27] Streaming of audio and video over the Internet[10,11,12,24,26] is characterized by relatively low bit rates, relatively high packet loss rates and relatively large packet jitter. A common feedback method consists of measuring the packet loss rate and/or packet jitter at the receiver and sending the measured data back to the sender. The sender may apply a so-called *probe-based* scheme[12,24] in which essentially the video rate is increased incrementally until the packet loss rate increases above acceptable levels, after which the sending rate is reduced. However, in our case, the goal is to avoid packet losses as much as possible to retain high quality. Major concerns in Internet video schemes are congestion control, TCP-friendliness and multicast capability;[10,11,12] however, these are not the primary concerns in wireless home networks.

The channel capacity estimation method proposed by Loguinov and Radha[26] is similar to the method proposed here. Both can be seen as extensions to the so-called packet pair algorithms and their variants.[25,27] However, this prior work is intended for use in the Internet, to estimate the so-called *bottleneck bandwidth* of an end-to-end Internet path, where the bottleneck bandwidth is the bandwidth of the slowest link in the end-to-end path. The bottleneck bandwidth of such end-to-end paths gives an upper bound on how fast data can possibly be transmitted, and this quantity does not change on a continuous basis. In contrast, the available bandwidth in a wireless link may change over very small time scales. Other methods for estimating Internet bottleneck bandwidth[25,27] often involve generation of significant amounts of probe (or test) traffic. Such probe traffic presents a significant overhead in that it lowers the bandwidth available for actual audio/video streams. Also, such methods often estimate bandwidth off-line, where the estimate is formed after all test traffic was sent and received. We, on the other hand, seek to utilize the intrinsic audio/video application traffic to estimate available bandwidth on-line, and in real-time.

4.1. Single stream channel and feedback model

Our method is based on an abstract model of the IEEE 802.11 MAC protocol operating in DCF mode.[1] Here, we assume a single audio/video traffic stream is present from sender to receiver. The 802.11 MAC protocol employs two primary mechanisms to control errors in packet transmissions. The first mechanism is retransmission of packets (called *data frames* in 802.11) that were not received successfully/correctly by the receiver. A sender retransmits a data frame when it does not receive an acknowledgement from the receiver. Protocol layers above the 802.11 MAC are not aware of such sequences of (multiple) data frame transmission(s) followed by an acknowledgement. These higher protocol layers only experience variations in the duration of a successful packet exchange, corresponding to variations in the bandwidth seen at those layers. The second mechanism is switching to lower data links rates offered by the 802.11 PHY that are more resistant to interference. WLANs based on IEEE 802.11a/b/g can operate at several data link rates and the standard allows switching between these rates on a packet-by-packet basis. Lower link rates reduce the probability that data frames are lost and need to be retransmitted by the MAC, but at the same time result in lower bandwidth seen at higher protocol layers.

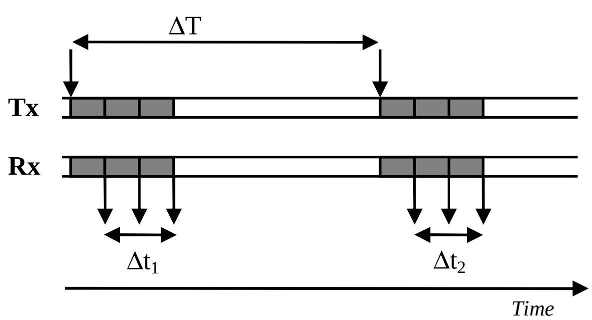
In the following, we define bandwidth as the *maximum throughput* offered by lower network layers to the audio/video application layer at a particular time. We denote the maximum throughput or bandwidth available for transmission of audio/video data by H, which may vary over time for reasons discussed above. We define the *sending rate*, denoted by S, as the rate at which the sender submits data to the network. We define *effective throughput*, denoted by E, as actual throughput of audio/video data realized at any particular time between sender and receiver.

Naturally, the effective throughput can not exceed the maximum throughput or bandwidth at any time: $E \leq H$. So, given the maximum throughput of the channel, say $H = \mathrm{H_A}$ Mbps, the sender must transmit data at a rate, say $S = \mathrm{S_A}$, that is lower than $\mathrm{H_A}$, to ensure that all transmitted data arrives at the receiver (without packet loss). In that case, the effective throughput, say $E = \mathrm{E_A}$ Mbps, is equal to the sending rate: $\mathrm{E_A} = \mathrm{S_A} \leq \mathrm{H_A}$. If the sending rate were higher than the bandwidth, some of the data sent would be lost: $\mathrm{S_A} \geq \mathrm{H_A} = \mathrm{E_A}$. Therefore, the first goal of the sender is to ensure that $E = S \leq H$ at all times. The sender controls its sending rate by scheduling the transmission of audio/video data packets appropriately. The second goal of the sender is to ensure that the bit rate of the video generated, R', is equal to the sending rate: $R' = S$. As discussed in the previous section, this is achieved by appropriate rate control at the encoder or transcoder. The overall goal of the sender, in the single stream case, is to achieve $R' = S \leq H$.

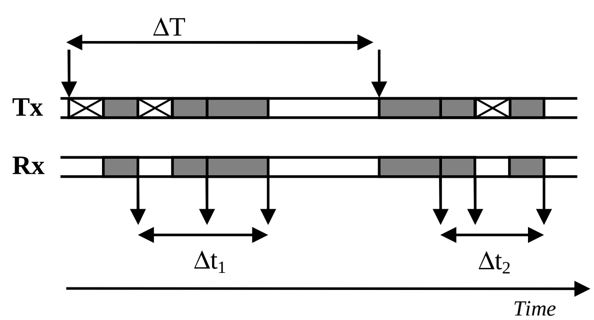
As an example, in the case that the bit rate of the original video stream is already lower than the network bandwidth, the transrater does not have to actively lower the bit rate. The sender must simply match the packet sending rate to the video bit rate in order to achieve the above goals. However, when the given bandwidth starts to decrease and approaches the video bit rate, the sender must react to avoid packet losses, by appropriately lowering its sending rate and the transrater output bit rate dynamically.

4.2. Estimating bandwidth using bursts of packets

We use a method for estimating available bandwidth of a network link that is based on measurement of the time intervals between the arrivals of packets. Such measurements are performed at the receiver application layer, i.e., after the application layer receives a data packet from the lower network layers (e.g. 802.11 MAC, and possibly IP/UDP) and before the data packet is handed to an audio/video decoder (i.e. in the bandwidth monitor module shown in Figure 1). This method is based on the use of *groups* or *bursts* of multiple data packets,[26] and is illustrated in Figure 3. Such bursts of audio/video data packets are formed by the sender application layer. Subsequently, these bursts of packets are submitted to the lower network protocol layers in a back to back manner, and the network transports the packets. At the receiver, the arrival time of individual packets in each burst at the application layer is measured. These bursts of packets are transmitted by the sender in a periodic or semi-periodic fashion. The sender implements a transmission of a burst of packets by temporarily buffering a number of packets that it receives from the encoder or transcoder for transmission. After buffering a number of packets, the sender (at the application layer) then transmits the set of buffered packets as a burst, i.e., without any waiting period between consecutive packet send operations. The sender then goes back to buffering the next group of packets to form the next burst, etc.



a.



b.

Figure 3. Packet burst transmissions over 802.11 WLAN MAC, in case of a) ideal channel conditions, and b) non-ideal channel conditions. The diagrams illustrate submission of bursts of packets by the application layer to the transmitter (Tx) and the arrival of such packets at the application layer from the receiver (Rx) with arrows. The duration of each burst (Δt) is measured at the receiver. The time interval between bursts is scheduled to be ΔT. Time slots in the diagram in b) with an X inside indicate packets that were not acknowledged by the receiver MAC and required re-transmission. Time slots in the diagram in b) that are longer than the others indicate packets transmitted at a lower link rate, effectively having a greater duration.

As illustrated in Figure 3, the duration of such bursts as seen at the receiver is directly related to available bandwidth of the channel. In both cases shown in Figure 3, an effective throughput E is achieved, of say $E = E_S$ Mbps. In the first (ideal) case, the channel actually would support a higher throughput, up to a maximum throughput or bandwidth H, say $H = H_A$ Mbps. Therefore, $E_S < H_A$, and there is room for additional traffic. In this case, the sender may decide to increase the video bit rate using the transrater rate control (as long as the rate does not exceed that of the input video), and to increase its packet sending rate. In the second (non-ideal) case, the maximum throughput drops because the underlying MAC uses more of the capacity of the channel to transmit the packets in the data stream and there is less room for additional traffic. The maximum throughput in this case, say $H = H_B$ is lower than in the first case: $H_B < H_A$. The effective throughput can still be supported: $E_S < H_B$ holds as well. Depending on the margin between E_S and H_B, the sender may decide to decrease the video bit rate using transrating, and to decrease its packet sending rate.

In the case that all packets in each burst arrive successfully at the receiver, the effective data throughput E equals

$$E = \frac{P \cdot M}{\Delta T} \tag{1}$$

where P denotes the packet size or the size of the data payload of each packet, M denotes the number of packets in each packet burst, and ΔT denotes the time interval between subsequent packet bursts. Here, we assumed that each packet in a single burst has the same payload size. The sender can control its sending rate, and in turn the effective throughput, by appropriately scheduling packets and packet bursts, i.e., by controlling P, M and ΔT. The packet scheduler may be part of the sender network interface shown in Figure 1.

The maximum throughput H is achieved, albeit temporarily, during transmission of a packet burst. Therefore, the maximum throughput can be estimated by:

$$H = \frac{P \cdot (M - 1)}{\Delta t} \tag{2}$$

where Δt is the time interval between the arrival of the first and the last packet of a burst (see Figure 3). In this manner, bursts of data packets during the streaming of audio/video data can be used to estimate bandwidth on a periodic basis, in real time.

4.3. Robustness of bandwidth estimates

Measurements of bandwidth are subject to errors that may be considered as noise, as will be illustrated in section 6. For example, the (limited) resolution of the clock used to measure packet arrival times may cause such errors.[26,27] Robustness of the bandwidth estimate can be improved by increasing the number of packets per burst M. For example, using bursts with more than two packets reduces the effects of limited resolution of the measurement clock.[26,27] However, increasing M means the buffer size at the sender side must be increased, resulting in a higher cost of implementation, and higher transmission delays for audio/video data. A common solution in prior work on measuring Internet bottleneck bandwidth is to collect a frequency distribution of a set of bandwidth estimates and taking either the mean, median or mode of that distribution as the final bandwidth estimate.[25,26,27] However, such techniques may require a relatively long time period before a meaningful distribution can be collected, and do not exploit the sequential nature of the measurement samples.

Here, we propose to consider this time-sequential nature of the data, and to apply well-known signal processing techniques for smoothing, estimation and prediction. Examples are FIR or IIR filtering, statistical processing (including mean square error estimates, maximum a posteriori estimates, Wiener filtering, Kalman filtering, etc.), and curve fitting. Statistical processing that includes both filtering and prediction may be particularly useful, since the results of past measurements are used to control the rate of audio/video data transmitted at some later time. To compute a final estimate of the bandwidth $\hat{H}[i]$ at burst i, our current implementation employs simple first-order IIR filtering of initial bandwidth samples $H[j]$ for bursts $j \le i$, as follows:

$$\hat{H}[i] = (1 - w) \cdot \hat{H}[i-1] + w \cdot H[i] \tag{3}$$

where w is a smoothing parameter between 0 and 1. Alternatively, one may filter the measured time intervals (Δt) before computing a bandwidth estimate. The final estimate of the bandwidth at a burst i, $\hat{H}[i]$, is transmitted back to the sender immediately after receiving all packets in the burst. The feedback from receiver to sender may also include an indication of any packet loss detected by the receiver. The sender responds to such information about detected packet loss in a similar fashion as it responds to decreases in measured bandwidth: by lowering the video bit rate and the sending rate.

5. MULTI-STREAM CHANNEL MODEL AND BIT ALLOCATION

This section discusses the case of multiple video streams being transmitted simultaneously from a single server to multiple clients over a wireless LAN. In particular, we present a simple channel model and bit allocation technique aimed at jointly optimizing the quality of multiple video streams.

5.1. Multi-stream channel model

We have already mentioned several characteristics of wireless LANs. The physical and data link protocol layers of such networks are designed to mitigate the adverse conditions of the channel medium. One of the characteristics of these networks specifically affects bit allocation among multiple streams as in a multi-stream transmission system. In particular, in the case of IEEE 802.11, an access point at the server may be communicating at *different* data link rates with different client devices.[1,4] WLANs based on IEEE 802.11 can operate at several data link rates, and may switch or select data link rates adaptively to reduce the effects of interference or distance on traffic between the access point and a particular client device. Different traffic streams to different stations in the network may be transmitted at different data link rates. We refer to this capability of the network as *multi-rate support*. The fact that the server may be communicating with different client devices at different data rates, in a single wireless channel, affects the model of the channel as used in bit allocation for joint coding of multiple streams.

Prior work[15,16] in joint rate control and bit allocation uses a conventional channel model, where there is a single channel rate that can simply be divided among video streams in direct proportion to the requested rates for individual streams. This conventional channel model does not hold in 802.11 WLANs due to their multi-rate capability. Due to the possible use of different data link rates for transmissions to different client devices, the bandwidth available to one device may be different from the bandwidth available to another device, while traffic to each device still contributes to the overall utilization of a single, shared, channel. In this section, we denote the maximum throughput or bandwidth available to device n by H_n. This bandwidth may vary over time due to variations in the channel conditions. A mechanism to estimate H_n has been proposed in section 4, and in this section we assume the bandwidth values H_n for all devices ($n = 1, 2, \ldots, N$) are known.

Our model of the shared channel is such that the server transmits data to each client device n for a fraction f_n of the time. Therefore, an effective throughput is obtained from the server to client n equal to $f_n H_n$. Furthermore, the multi-stream channel constraint is given by:

$$\sum_{n=1}^{N} f_n \leq 1.0 \tag{4}$$

If the fractions f_n add up to 1.0, the channel is utilized to its maximum capacity.

Let R'_n denote the bit rate of stream n at the output of the encoder or transcoder. To be able to transmit video streams to all devices concurrently, there must exist a set of f_n, $n = 1, 2, \ldots, N$, such that for all n:

$$f_n H_n = R'_n \tag{5}$$

under the constaint of Eq. 4. In general, the problem of determining a set of fractions f_n is an under-constrained problem. Naturally, the goal is to find a solution to this problem that maximizes some measure of the overall quality of all video streams combined. In the following, we present a solution that is based on a joint coding principle, where the bit rates of different streams are allowed to vary based on their relative coding complexity, in order to achieve a uniform picture quality across streams. This approach implicitly maximizes the minimum quality of any video stream that is jointly coded.

5.2. Joint multi-stream bit allocation

A practical approach to the problem of bit allocation for joint coding of multiple video programs was proposed by Wang and Vincent.[16] This approach extends the approach used in the MPEG-2 Test Model 5,[22] using the notions of *super GOP* and *super frame*. A super GOP is formed over multiple MPEG-2 streams and consists of L super frames, where a super frame is a set of N frames containing one frame from each stream and all frames in a super frame coincide in time.

Wang and Vincent[16] propose a bit allocation method in which a target number of bits is assigned to a super GOP based on a given (constant) channel bit rate. Subsequently, a target number of bits is assigned to each super frame within a super GOP. Finally, a target number of bits is assigned to each frame within a super frame. In particular, streams are allocated bits proportional to the estimated coding complexity of the frames in each stream. That is, streams with frames that are more complex are allocated more bits compared to streams that are less complex, resulting in an equal amount of distortion in each stream.

In this paper, we extend the bit allocation method of Wang and Vincent for the case of channels with multi-rate capability, such as those based on IEEE 802.11 WLANs. We consider N video streams, where each stream is encoded with GOPs of equal length L. We consider a set of N GOPs, one from each stream, concurrent in time, forming a super GOP containing $N \times L$ frames. The first step in the proposed bit allocation is to assign a target number of bits to each GOP in a super GOP, where each GOP belongs to a different stream. The allocation is performed in proportion to the relative complexity of each GOP in a super GOP. The second step in the bit allocation procedure is to assign a target number of bits to each frame of the GOP of each video stream.

Let $r_{n,t}$ denote the number of bits generated by the encoder/transcoder for frame t of video stream n. Controlling the rate on a GOP-by-GOP basis, we can reformulate Eq. 5 as follows:

$$f_n H_n = \frac{frame\_rate}{L} \sum_{t=1}^{L} r_{n,t} \tag{6}$$

Following the MPEG-2 Test Model 5 and Wang and Vincent,[16] we assume $r_{n,t}$ is related to a coding complexity measure $C_{n,t}$ and the quantizer value $Q_{n,t}$ for each frame, as follows:

$$C_{n,t} = Q_{n,t} \; r_{n,t} \tag{7}$$

As stated, we adopt a constant quality approach, meaning that the quantizer values over different streams must be equal, up to a constant factor $K_{n,t}$ that accounts for the differences in picture types (I, P and B):

$$Q_{n,t} = K_{n,t} \; Q \tag{8}$$

where $K_{n,t}$ is simply either K_I, K_P or K_B, depending only on the frame type. We substitute Eq. 7 and Eq. 8 in Eq. 6, and utilize Eq. 4 (factoring out Q), to obtain the following unique solution for the set of unknowns f_n :

$$f_n = \frac{\frac{1}{H_n} \sum_{t=1}^{L} \frac{C_{n,t}}{K_{n,t}}}{\sum_{n=1}^{N} \frac{1}{H_n} \sum_{t=1}^{L} \frac{C_{n,t}}{K_{n,t}}} \tag{9}$$

The coding complexity measures $C_{n,t}$ are typically estimated from prior encoded frames, or, in the case of transcoding, utilizing information from the input bit stream.[16,23] Note that we assumed, without loss of generality, that the channel is utilized to its maximum capacity, i.e., the sum of channel utilization fractions adds up to 1.0. If, in practice, some of the channel capacity has to be set aside for other types of traffic, this can be taken into account by scaling the fractions f_n such that their sum is kept below 1.0.

Given f_n, the actual target rate for a GOP of stream n can be computed using Eq. 6. As mentioned above, the second step in the bit allocation procedure is to assign a target number of bits to each frame of the GOP of each video stream. This can be achieved using existing bit allocation methods, such as the one provided in TM5. Subsequent coding or transcoding can be performed as usual, with any standard method.

When channel conditions change over time, the throughput rates H_n will vary. In this case, the above bit allocation technique can be executed at every frame time, where summations are performed over sliding windows with the length of a GOP. This results in dynamic rate adaptation for multiple video streams as shown in Figure 1. Note further, that the above bit allocation method can be easily modified for the case where GOPs are not used.

6. EXPERIMENTAL RESULTS

In this section, we present results of single stream video transrating, as well as channel bandwidth estimation.

6.1. Video rate adaptation results

We first present results for single stream transrating at constant bit rates, followed by results for transrating with a time-varying target rate. The results reported here are based on the public software implementation of the MPEG-2 Test Model encoder and decoder.[22] Our transrater implementation is based on the MPEG-2 Test Model as well.

The first set of results compares transrating with direct encoding, as well as re-encoding (i.e., full decoding followed by full encoding). These results are for the Mobile & Calendar sequence, with CIF resolution (352x288 pixels), at 30 frames per second (progressive). The original sequence was encoded once at 4.0 Mbps, resulting in a compressed video stream with average PSNR of the luminance frames of 34.2 dB. Subsequently, this encoded bit stream was transrated to 3.0 Mbps, 2.5 Mbps and 2.0 Mbps, using our re-quantization transrater. Furthermore, the 4.0 Mbps bit stream was re-encoded at the same bit rates for comparison. We also compare against the results of direct encoding at 3.0, 2.5 and 2.0 Mbps.

The first set of results is summarized in Figure 4. As expected, transrating results in a decrease of the PSNR compared to direct encoding. Interestingly, the PSNR performance of our open-loop re-quantization transrater is very close to that of full re-encoding, except at 2.0 Mbps, in this case. Therefore, this transrater achieved a huge savings in computational complexity over re-encoding, at a modest decrease of visual quality, over the range of bit rates tested here.

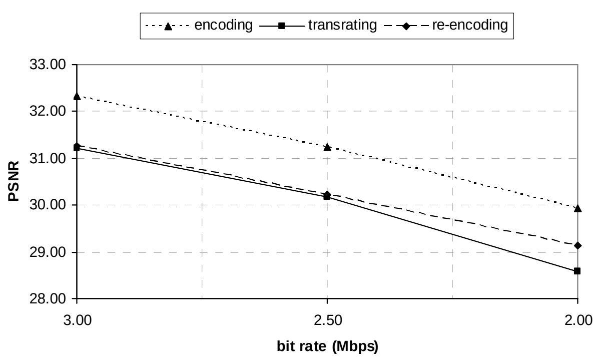


Figure 4. Average PSNR of luminance frames at 3.0, 2.5 and 2.0 Mbps, for transrating, re-encoding, and direct encoding (Mobile & Calendar, 352x288, 30 frames per second).

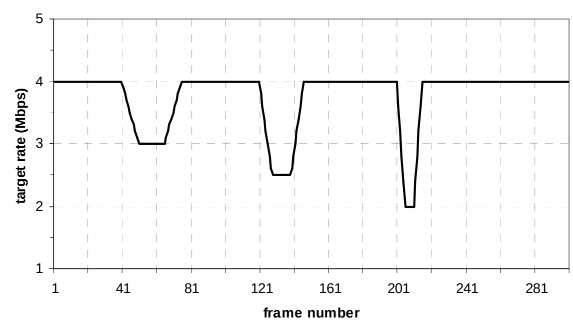


Figure 5. Time-varying target bit rate, simulating drops in available channel bandwidth.

The second set of results illustrates the ability of the transrater to respond to changes in the target bit rate over time. In this experiment, we provided a time-varying target bit rate to the transrater bit allocation and rate control mechanism, as shown in Figure 5. This (manually generated) target corresponds to a situation where the input video bit rate is at 4 Mbps, and this bit rate needs to be reduced at a few instances in time due to repeated drops in available channel bandwidth. The drops in channel bandwidth simulated here are of decreasing duration and increasing magnitude (down to 3.0, 2.5 and 2.0 Mbps respectively). We compare the resulting variations in the PSNR, as well as in the number of bits generated per frame, against the case of simply transrating the input to a constant 4 Mbps target output. The original sequence is the same as above (Mobile & Calendar).

The results are shown in Figure 6 and Figure 7. Note that the number of bits generated by the transrater varies significantly from frame to frame, depending strongly on the frame type (I, P or B). Therefore, in Figure 6, instead we show the ratio of the actual number of bits generated for a frame over the target number of bits for that frame. Ideally, this ratio should remain at or close to 1.0 for every frame, independent of its type. As shown in the diagram, this ratio stays within 5% of its ideal value for the reference case (transrating at 4.0 Mbps without a change in the target rate). The ratio stays within 10% of its ideal value most of the time in the case where the target rate changed dynamically as in Figure 5. The main exception in both cases is near the start of the sequence, which may be expected. The diagram also shows that the number of bits generated by the transrater with a dynamic target rate is nearly the same as that of the reference at most times other than the times where drops in the target occurred. The change of PSNR over the frames of this sequence is shown in Figure 7, for both cases. Again, the PSNR is seen to diverge from the reference only near areas where drops in the target rate occurred.

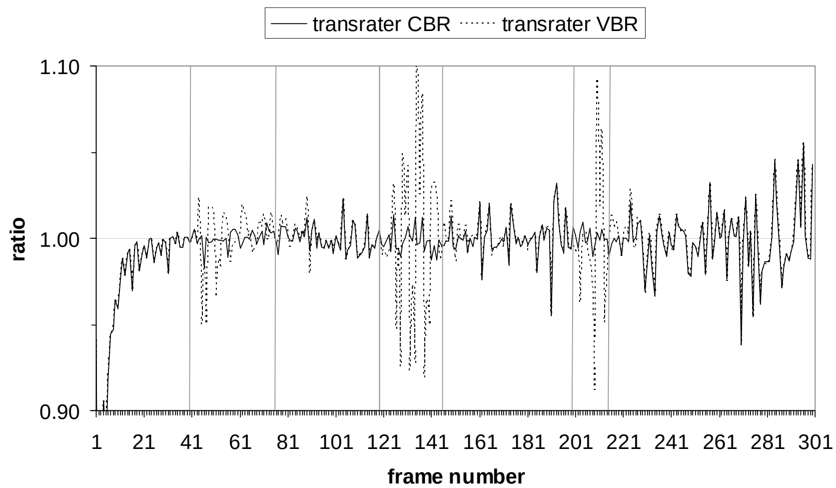


Figure 6. Ratio of actual number of bits generated over target number of bits, for each frame, for constant bit rate transrating vs. transrating with a time-varying target bit rate.

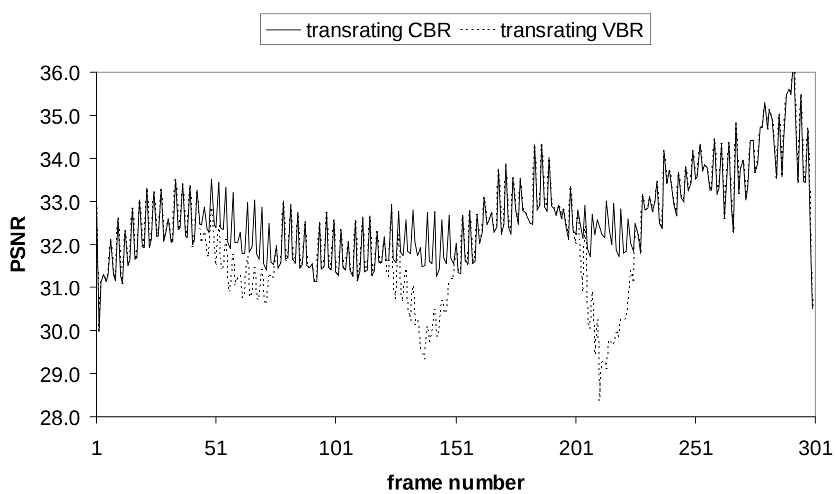


Figure 7. PSNR of luminance frames for transrating at a constant bit rate vs. a time-varying target bit rate (Mobile & Calendar, 352x288, 30 frames per second).

6.2. Channel bandwidth estimation results

We have implemented a client/server streaming test bed consisting of two laptop PCs with 850-900 MHz Pentium III processors running Windows 2000, with 802.11b WLAN cards, configured in ad-hoc networking mode. The laptops transport data packets using UDP/IP on top of 802.11b. Software running on one PC acted as a server, sending data packets to the receiver using the UDP, IP and 802.11b protocols. Measurements of time can be provided by clocks internal to the hardware/software platform. Different platforms offer different APIs and may support clocks with different performance in terms of clock resolution. We have used a Windows API offering a high performance counter with sub-millisecond resolution, the exact resolution depending on the hardware. For example, the resolution of this timer was found to be better than microsecond resolution on the two Windows 2000 laptop PCs used for testing, and was found to be better than nanosecond resolution on a newer Windows XP desktop PC with a Pentium 4 processor. Such resolution is sufficient for our purposes. All measurements reported here were carried out indoors in our laboratory building, which contains few interior walls but many cubicle areas.

An initial set of experiments was carried out to validate our test bed, consisting of measurements of maximum throughput using different 802.11b link rates, different packet sizes and different distances between the two laptops. The best throughput in our set up was obtained when using a link rate of 11Mbps, and an IP packet size of 1500 bytes (i.e. a packet payload of 1472 bytes): approximately 6.2 Mbps. IP packets with sizes larger than 1500 were fragmented by our equipment, resulting in lower throughputs (due to the duplication of packet headers), even though the 802.11 standard allows larger packet sizes. The maximum throughput dropped to 3.9 Mbps when using a 5.5 Mbps 802.11 link rate, and dropped further to 1.7 Mbps when using a 2 Mbps link rate. In our measurements, we were able to increase the distance between the two laptops to about 30 m without experiencing a drop of the bandwidth or packet loss. Measurements at a distance of about 43 m revealed significant drops in bandwidth, down to 3.4 Mbps when using a payload of 1000 bytes, and down to 3.0 Mbps when using a payload of 1472 bytes, both at link rate of 11 Mbps. The above bandwidth values were computed on the basis of repeated measurements of the transmission duration of sequences of 1000 back-to-back packets.

The next set of results illustrate our on-line bandwidth estimation technique in more realistic scenarios. The first example illustrates bandwidth performance in an ideal case, where the laptop PCs were located close to each other, and no interference from external sources was present. The 802.11b link rate was 11Mbps, while the packet payload (P) was constant at 1472 bytes. Each experiment consisted of transmission of 100 short bursts. In this example, each burst consisted of 10 packets ($M = 10$) and the time between subsequent bursts (ΔT) was scheduled to be 40ms. Therefore, effective throughput in this case is approximately 2.9 Mbps. The raw bandwidth samples, as well as smoothed bandwidth values ($w = 0.1$) are shown in Figure 8. From earlier measurements, and from literature, we know that the bandwidth in this case is 6.2 Mbps on average. Note that the bandwidth samples vary somewhat around the 6.2 Mbps value; their average value over 100 bursts is 6.24 Mbps and the standard deviation is 0.22 Mbps. Smoothing by IIR filtering brings the standard deviation down to 0.08 Mbps.

The second example illustrates bandwidth performance in non-ideal conditions, where the laptop PCs were located at a distance of 43 m from each other, separated by many cubicles and a few walls. All other parameters were the same as in the first example. Raw and smoothed bandwidth samples for the non-ideal case are shown in Figure 9. The average bandwidth over 100 bursts is much lower in this case: 3.3 Mbps. The standard deviation of the raw samples is much higher: 1.20 Mbps. Smoothing by IIR filtering brings the standard deviation down to 0.23 Mbps. In these experiments, no packet loss was observed. These results indicate that the 802.11 MAC was able to transmit all packets successfully, using the re-transmission mechanism, although resulting in varying bandwidth.

7. CONCLUSION

In this paper, we have proposed a bandwidth-adaptive system for robust streaming of high-quality video over wireless LANs in a home environment. We have proposed a feedback-based approach to address the time-varying nature of the channel and the unpredictability of available bandwidth in such wireless LANs. We provide techniques for dynamic video rate adaptation and dynamic measurement of the channel bandwidth. We have described a bandwidth estimation algorithm that uses the intrinsic video traffic by sending it in bursts of packets. A video transrater at the sender adapts the video bit rate continuously, based on the channel bandwidth estimates obtained at the receiver. We also provide an algorithm for joint bit allocation and rate control for transmitting multiple video streams over a wireless network, based on a simple model of the channel. We have reported initial results on dynamic video rate adaptation obtained with a low-complexity transrater, and results on bandwidth estimation measured using an actual 802.11b wireless LAN set up.

Further work will include experiments with a closed loop bandwidth monitoring system, where bandwidth estimates from the receiver are fed back to the sender, and where a packet scheduler at the sender adapts data transmissions accordingly. Furthermore, we plan to integrate the video transrater itself in this closed loop system, and perform further experiments with the transrater bit allocation and rate control in real-life scenarios. Finally, we plan to test our novel multi-stream channel model and joint bit allocation and video coding technique.

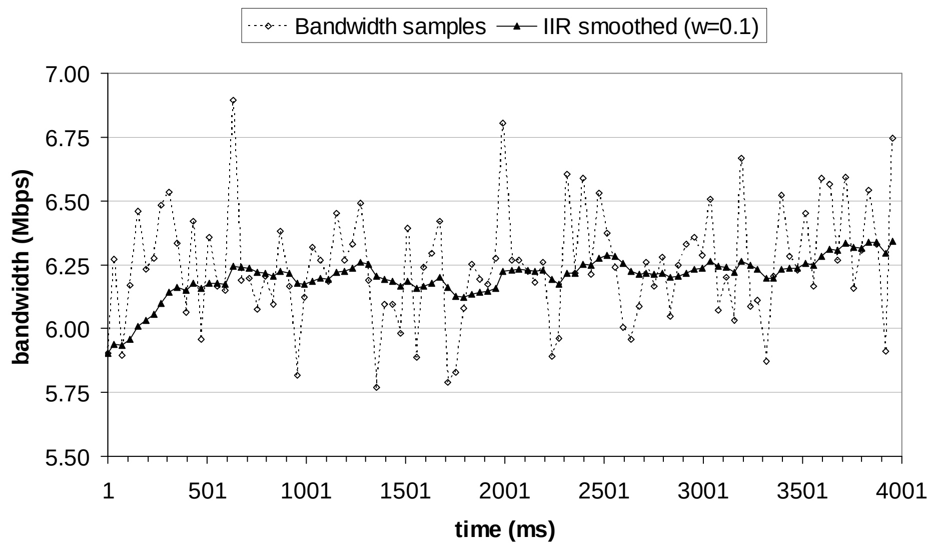


Figure 8. Measurements of maximum throughput/bandwidth using the packet burst method in ideal conditions.

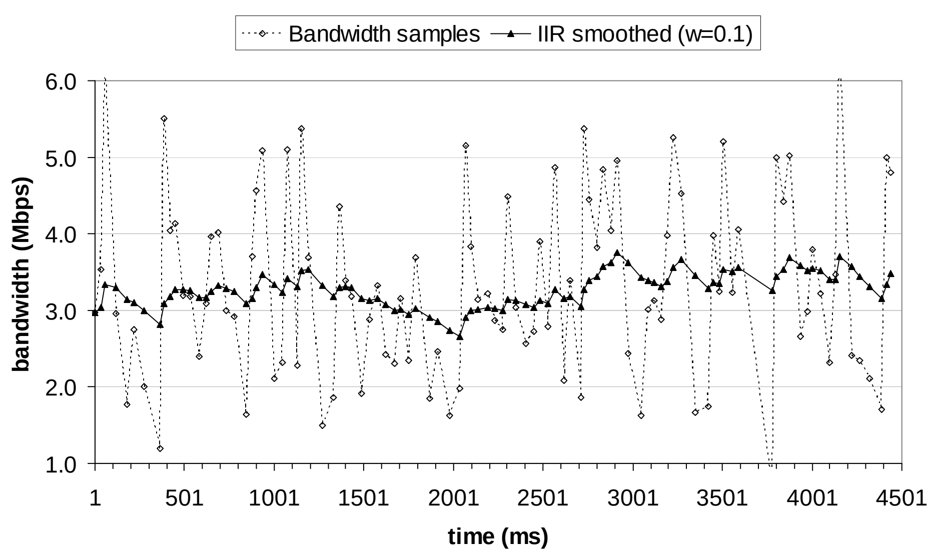


Figure 9. Measurements of maximum throughput/bandwidth using the packet burst method in non-ideal conditions (43 m distance between sender and receiver, including several walls).

REFERENCES

1. B. O'Hara and A. Petrick, *The IEEE 802.11 Handbook - A Designer's Companion*, IEEE Press, 1999.
2. S. Gardner, B. Markwalter and L. Yonge, “HomePlug standard brings networking to the home,” *Communication Systems Design*, vol. 6, no. 12, 2000.
3. J.C. Chen and J.M. Gilbert, “Measured performance of 5-GHz 802.11a wireless LAN systems,” *Atheros Communications*, white paper, 27 August 2001.
4. W. Smith, “Wireless video - Obtaining persistent video quality from a inherently variable wireless medium,” *ViXS Systems Inc.*, white paper, 9 January 2002.
5. A. Vasan and A.U. Shankar, “An empirical characterization of instantaneous throughput in 802.11b WLANs,” Technical Report CS-TR-4389, University of Maryland, College Park, MD, November 2002.
6. A. Kamerman and A.R. Prasad, “Performance analysis,” in: *WLAN Systems and Wireless IP for Next Generation Communications*, Editors: N. Prasad and A. Prasad, pp. 213-239, Artech House, Boston, 2000.
7. B. Girod and N. Färber, “Wireless Video,” in: *Compressed Video Over Networks*, Editors: M.-T. Sun and A.R. Reibman, pp. 465-511, Marcel Dekker, 1999.
8. C.-Y. Hsu, A. Ortega and M. Khansari, “Rate control for robust video transmission over wireless channels,” *Visual Communications and Image Processing ’97*, Proc. SPIE Vol. 3024, pp. 1200-1211, 1997.
9. S. Aramvith, I. Pao and M.-T. Sun, “A rate-control scheme for video transport over wireless channels,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 5, pp. 569-580, 2001.
10. D. Wu et al., “Streaming video over the Internet: Approaches and directions,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 282-300, 2001.
11. J.F. Kurose and K.W. Ross, *Computer Networking: a top-down approach featuring the Internet*, Addison-Wesley, 2001.
12. J. Bolot and T. Turletti, “Experience with control mechanisms for packet video in the internet,” *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 1, pp. 4-15, 1998.
13. A. Majumdar et al., “Multicast and unicast real-time video streaming over wireless LANs,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 524-534, 2002.
14. G. Berbecel et al., “High quality video over powerline networks,” *SID Int. Symposium, Seminar & Exhibition (SID 2003)*, pp. 340-343, Baltimore, Maryland, 2003.
15. G. Keesman and D. Elias, “Analysis of joint bit-rate control in multi-program image coding,” *Visual Communications and Image Processing ‘94*, Proc. SPIE Vol. 2308, pp. 1906-1917, 1994.
16. L. Wang and A. Vincent, “Bit allocation and constraints for joint coding of multiple video programs,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 6, pp. 949-959, 1999.
17. N.G. Duffield and K.K. Ramakrishnan, “Feedback of rate and loss information for networked video,” in: *Compressed Video Over Networks*, Editors: M.-T. Sun and A.R. Reibman, pp. 383-430, Marcel Dekker, 1999.
18. A. Vetro, C. Christopoulos and H. Sun, “Video transcoding architectures and techniques: An overview,” *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18-29, 2003.
19. H. Sun, W. Kwok and J.W. Zdepski, “Architectures for MPEG compressed bitstream scaling,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 191-199, 1996.
20. P.A.A. Assunção and M. Ghanbari, “A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 8, pp. 953-967, 1998.
21. Y. Nakajima, H. Hori and T. Kanoh, “Rate conversion of MPEG coded video by re-quantization process,” *Proc. IEEE Int. Conf. on Image Processing (ICIP 95)*, vol. III, pp. 408-411, Washington, DC, 1995.
22. ISO/IEC JTC1/SC29/WG11 N0400, “MPEG-2 Test Model 5,” Sydney, Australia, April 1993.
23. L. Lu, S. Xiao, J. L. Kouloheris and C. A. Gonzales, “Efficient and low cost video transcoding,” *Visual Communications and Image Processing 2003*, Proc. SPIE Vol. 4671, pp. 154-163, 2002.
24. J. Bolot and T. Turletti, “A rate control mechanism for packet video in the Internet,” *Proc. IEEE INFOCOM ’94*, pp. 1216-1223, Toronto, Canada, 1994.
25. K. Lai and M. Baker, “Measuring bandwidth,” *Proc. IEEE INFOCOM ’99*, pp. 235-245, New York, NY, 1999.
26. D. Loguinov and H. Radha, “Video receiver based real-time estimation of channel capacity,” *Proc. IEEE Int. Conf. on Image Processing (ICIP 2002)*, vol. III, pp. 213-216, Rochester, NY, 2002.
27. V. Paxson, “End-to-end Internet Packet Dynamics,” *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 4, pp. 139-152, 1997.