EURASIP Journal on Applied Signal Processing 2005:2, 220–228

Cross-Layer Techniques for Adaptive Video Streaming over Wireless Networks

Yufeng Shan

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA
Email: yfshan@eecs.berkeley.edu

Received 16 July 2003; Revised 9 June 2004

Real-time streaming media over wireless networks is a challenging proposition due to the characteristics of video data and wireless channels. In this paper, we propose a set of cross-layer techniques for adaptive real-time video streaming over wireless networks. The adaptation is done with respect to both channel and data. The proposed novel packetization scheme constructs the application layer packet in such a way that it is decomposed exactly into an integer number of equal-sized radio link protocol (RLP) packets. FEC codes are applied within an application packet at the RLP packet level rather than across different application packets and thus reduce delay at the receiver. A priority-based ARQ, together with a scheduling algorithm, is applied at the application layer to retransmit only the corrupted RLP packets within an application layer packet. Our approach combines the flexibility and programmability of application layer adaptations, with low delay and bandwidth efficiency of link layer techniques. Socket-level simulations are presented to verify the effectiveness of our approach.

Keywords and phrases: video streaming, wireless, cross layer, multimedia.

1. INTRODUCTION

Real-time streaming media over wireless networks, such as live video streaming of a football match to a wireless terminal, is becoming an increasingly important application. In order to be real time, the end-to-end delay of the delivery system should be minimized, given a certain quality requirement. Traditional streaming systems (http://www.microsoft.com/windows/windowsmedia, http://www.real.com) use a large receiver buffer to absorb channel variation and to facilitate the utility of error-recovery schemes. The receiver prebuffering time usually varies from several seconds to tens of seconds, which is not suitable for real-time streaming. In this paper, we propose a set of building blocks for channel and application adaptive real-time wireless streaming applications. Our approach combines the flexibility and programmability of application layer adaptations, with low delay and bandwidth efficiency of link layer techniques. We propose a novel packetization scheme so that forward error correction (FEC) codes can be applied within an application packet at radio link protocol (RLP) packet level rather than across different application packets and thus reduce delay at the receiver. Furthermore, a priority-based automatic repeat request (ARQ), together with a scheduling algorithm, is applied at the application layer to retransmit only the corrupted RLP packets to improve the wireless bandwidth efficiency.

Unlike wired packet switched networks that suffer from congestion-related loss and delay, the wireless networks have to deal with a time-varying, error-prone, physical channel that in many instances is also severely bandwidth constrained. As such, the solutions needed for wireless video streaming applications are fundamentally different from wired streaming. For instance, In [1], Qiao and Shin propose a two-step adaptive hybrid ARQ scheme for transmitting H.263 video sequences over wireless LANs, which, (1) based on both the wireless channel conditions and the deadline constraint, adaptively selects the best error correction code by looking to an optimal code table which is predetermined before starting the video service, and, (2) based on the actual frame loss events, adaptively uses the prebuilt optimal code table to guarantee certain quality of service (QoS) in terms of frame loss rate. An unequal error protection-(UEP-) based error-control scheme for transmission of low bit rate MPEG-4 video over wireless channels is proposed by Cai et al. in [2], where MPEG-4 bit stream is divided into two classes according to the importance of video data in a coded bit stream. In their scheme, an MPEG-4 bit stream is divided into many segments, and each segment is reorganized into a fixed-length structure. Then, UEP is applied to each class of data with respect to video frame reconstruction at the receiver. In [3], Budagavi et al. propose a solution for video transmission over wireless networks by combining the efficiency of DSP chips with the error resilience of MPEG-4 bit stream. From system design point of view, the authors analyzed all aspects of designing a wireless multimedia system on DSP chips, from MPEG-4 video algorithms, error-resilience tools, channel coding to processor capabilities. A theoretical analysis of the overall mean squared error (MSE) in hybrid video coding is presented in [4] by Stuhlmüller et al. for the case of error-prone transmission. Their proposed model covers a complete transmission system, including the rate-distortion performance of the video encoder, FEC, interleaving, and the effect of error concealment and interframe error propagation at the video decoder. Using the proposed model, the optimal tradeoff between intra- and intercoding as well as the optimal channel code rate can be determined for given channel parameters by minimizing the expected MSE at the decoder.

Zheng and Boyce in [5] propose a packet coding scheme and a protocol called "complete UDP" which uses information from the physical layer and RLP [6] layer to assist the packet level error recovery. In [5], the authors assumed that the error-burst length is small enough compared to the physical layer frame length and the error events are independent from frame to frame. Zorzi in [7] considers the problem of designing an error-control scheme under delay constraints. Two types of error-control schemes for a Gilbert-Elliott burst-error channel, namely, interleaved FEC block codes and Go-Back-N ARQ are investigated. The author presents an approximate characterization of the residual bit error rate at the output of the error-control scheme, and examines the effect of the channel burstiness, type of codes used, and channel data rate on the tradeoffs between residual bit error rate and delay, and between delay and achievable information rate. Li and van der Schaar in [8] focus on the efficient and robust transmission of video over one particular type of wireless LANs, namely IEEE 802.11, and propose a novel error-protection method that can provide adaptive QoS to layered coded video by utilizing priority queuing at the network layer and retry-limit adaptation at the link layer. The proposed cross-layer protection system can provide not only priority delivery services, but also UEP to the different video streams, by adapting different retry-limit settings in the media access control (MAC) for the multiple queues containing the different video streams priorities. Krishnamachari et al. in [9] evaluate different error-control and adaptation mechanisms available in the different layers for robust transmission of video, namely, MAC retransmission strategy, application layer FEC, bandwidth-adaptive compression using scalable coding, and adaptive packetization strategies. The authors propose a novel vertical system integration that enables the joint optimization of the various protection strategies existing in the protocol stack by performing tradeoffs between throughput, reliability, and delay depending on the channel conditions and application requirements. Based on their model, a strategy for the adaptive selection of application layer FEC, maximum MAC retransmission limit, and packet sizes depending on the channel condition to maximize the video quality under different multipath channel conditions is developed.

In wireless environments, the channel conditions change rapidly over time due to noise, interference, multipath, and the movement of the mobile host. In such a context, transmission control schemes have to dynamically adapt both to the application requirements and to the channel conditions. In this paper, we propose a set of building blocks for channel and application adaptive wireless streaming applications. In doing so, we exploit two well-known principles in wireless video communication: the first has to do with the fact that different parts of a video bit stream are of different importance, and hence need to be protected via FEC and ARQ to different degrees; the second has to do with adaptivity to channel conditions by dropping unimportant packets. While UEP for video bit stream has been around for a long time, most such techniques are applied at the bit level, rather than packet level. Since most of the today's wired and wireless networks are packet oriented, our approach in this paper is unequal treatment of packets and adaptivity at the packet level. Given this, the main issue that arises is: which network protocol layer this unequal treatment of video, and adaptivity needs to be done at? Is it better to do it at the application layer or at the link layer? Clearly, there are pros and cons for either case. At lower layers, such as link layer, implementation complexity to adapt to packet importance becomes an issue. In addition, many link layer protocols in today's networks do not support a mechanism for different FEC protection or retransmission policies for differently marked RLP packets. This would motivate one to apply unequal FEC and retransmission techniques at the application layer. However, applying FEC at the application layer results in excessive delays, if redundancy is applied across different packets. Also, adaptive retransmissions from the application layer have the inherent disadvantage of additional delay. Specifically, the packets at the application layer tend to be larger and hence, if existing transport protocols such as TCP are used, unless all RLP packets within the TCP packet are received successfully with fewer than the maximum number of retransmissions allowed at the link layer, the entire application layer packet needs to be retransmitted, resulting in excessive delay and waste of bandwidth. In the case of UDP, unless all link layer packets are received successfully, the entire UDP packet is lost.

Based on the above considerations, our proposed building blocks for channel and application adaptive schemes are as follows.

(1) Priority-based ARQ, together with a scheduling algorithm, is applied at the application layer; however, in order to avoid excessive delays, UDP-Lite [10] is employed at the transport layer, so that corrupted radio link layer packets can be passed along to the application layer at the receiver. This way, the receiver can ask the transmitter to resend only the corrupted RLP packets within an application layer packet. To facilitate the process of specifying the exact RLP packet to be resent, the application layer packet is constructed in such a way that an exact integer number of RLP packets, say M, is generated after decomposition of any application layer into RLP packets. This has two advantages: (a) the receiver only sends an index to the sender as to which of the M

packets were corrupted; (b) no modifications are needed at the link layer.

(2) Furthermore, to avoid excessive delay introduced due to FEC coding across application layer packets, FEC is implemented within an application packet, but not at bit level, rather at RLP packet level. Specifically, video data is first divided into four classes based on their importance; a UEP algorithm described in Section 3 is applied to distribute FEC bits among the four classes; finally packetization at the application layer is done in such a way that (a) a given application packet contains data and FEC for only one class, (b) after the application layer packet is decomposed into RLP packets, each RLP packet corresponds to either data or FEC. At the receiver, once the UDP-Lite passes on the RLP layer packets to the application layer, the application layer can decide whether or not retransmissions are needed on any portion of the application layer based on its awareness of the particular FEC used for that class.

The above proposed scheme has the advantage that all the processing is done end to end at the application layer; yet, FEC and ARQ granularity are at the RLP packet size, thus reducing delay, overhead, and waste of bandwidth.

This paper is organized as follows. A packetization scheme for the application layer is discussed in Section 2; in Section 3, class-based UEP is discussed; priority-base ARQ scheme is discussed in Section 4; simulations are presented in Section 5, followed by conclusions in Section 6.

2. APPLICATION LAYER PACKETIZATION SCHEME

The basic idea of our packetization scheme is to construct the application layer packet in such a way that it is decomposed exactly into an integer number of equal-sized RLP packets.[1] At the receiver side, the UDP-Lite protocol passes the corrupted data to the application. The application layer at the receiver side can easily identify which RLP packet is lost by analyzing the received application packet, and hence request for retransmission. At the RLP layer at the sender, the application packet with all the header information is divided into M data blocks shown as "B" in Figure 1, with their size equal to the payload of RLP packet. Our proposed packetization algorithm at the sender calculates the number of bits that should be included in the payload of the application packet using the following:

$$\text{payload} + \text{sizeof}(\text{header}) = M^{*}\ \text{sizeof}(\text{RLP payload}), \quad (1)$$

where M is an integer, and *payload* is the payload of the application packet as shown in Figure 2.

Every data block of payload has a 4-bit sequence number to indicate the position of the block in the application packet payload. The receiver may receive a packet with corrupted or lost blocks in it, as shown in Figure 2, where the

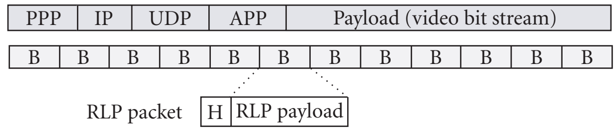


Figure 1: Packetization scheme in the application layer.

Figure 2: Payload structure of the application packet.

fourth block and the sixth block are lost. The application analyzes the received application packet, identifies lost blocks and, based on its knowledge of FEC level for that class of data, decides whether any retransmissions are needed. If so, the 4-bit sequence number in Figure 2 is used to indicate the RLP packet payload.

Due to UDP-Lite properties, one RLP layer packet loss corresponds to one data block loss in the application packet. The receiver can thus estimate the link layer packet loss rate by analyzing the lost blocks in the application packets. This information can then be fed back to the sender to determine optimal FEC levels for a given channel condition. We argue that this is a more accurate method to estimate the channel condition and loss rate at the RLP layer than the traditional application layer estimation techniques. This is because traditional schemes observe only one application packet loss regardless whether one or more RLP packets are lost. Suppose decomposition of header information in the application layer onto RLP packet results in p RLP packets, and decomposition of the video data results in q RLP packets. After N application packets, using our proposed scheme, the RLP packet loss rate P_{BLER} can be estimated at the receiver by

$$P_{\text{BLER}} = \frac{\sum_{i=0}^{i=N-1} L_i}{N \times (p+q)}, \quad (2)$$

where L_i is the number of lost RLP packets within the ith application packet. Since the loss of any header information in the RLP layer may cause the entire application packet to be lost, in this case, we use the following equation to estimate the L_i:

$$L_i = \frac{L_{i-1} + L_{i+1}}{2}. \quad (3)$$

Finally, successive application packets loss means a long burst loss in the channel; so for this situation, we choose to estimate the L_i as follows:

$$L_i = p + q. \quad (4)$$

Whenever an application packet arrives at the receiver side, the receiver estimates L_i and P_{BLER} based on (2)–(4) and feeds this information back to the sender. The sender uses this information to determine how much FEC should be added to each class of data.

[1] In doing so, we assume that the mobile host can obtain RLP packet size via negotiations with the base station at the beginning of the streaming session.

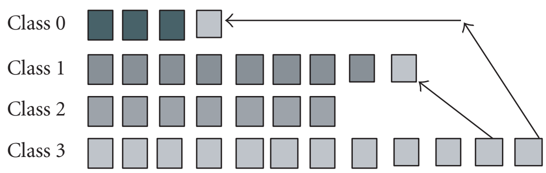


FIGURE 3: The FEC bandwidth allocation scheme.

TABLE 1: PSNR using EEP versus UEP.

BLER	3%	6%	9%	12%
EEP	27.985	22.356	19.816	17.320
UEP	33.956	30.865	29.137	27.540

3. CLASS-BASED UEP

The encoded MPEG-4 data in a bit stream includes interleaved header, shape, motion, and texture information. We classify the MPEG-4 bit stream according to the importance of data and dependency of frames as follows:

Class 0: header information;
Class 1: I and P frames with scene change;
Class 2: shape and motion information of P frames;
Class 3: texture information of P frames.

UEP has been proved to be an efficient way to protect different classes of data with different amounts of redundancy [2, 4]. In our scheme, a block-based Reed-Solomon (RS) code is applied as FEC over different classes of data. The RS encoder chooses k video data blocks and generates $(n - k)$ parity blocks. The video and parity blocks are packetized into one application packet as payload in such a way that once an application packet is decomposed into RLP packets, each RLP packet either corresponds to data or parity. Also each application packet corresponds to only one class of data. Every data block has its own block sequence number as shown in Figure 2. The block sequence number is useful at the receiver side in that it provides the RS decoder with the position of the lost block. The RS decoder can then recover up to $(n - k)$ lost blocks with this position information instead of recovering $(n - k)/2$ lost blocks without the position information. Given a target application packet loss rate P_{loss}, the estimated P_{BLER} from the receiver, and fixed n, the upper bound on k can be computed using

$$P_{\text{loss}} = \sum_{i=n-k+1}^{n} \binom{n}{i} P_{\text{BLER}}^{i} \left(1 - P_{\text{BLER}}\right)^{n-i}. \tag{5}$$

The UEP can be achieved by selecting different k's for different classes. The bandwidth allocated to FEC, B_{FEC}, is shared among all the data classes as shown below:

$$B_{\text{FEC}} = \sum_{i=0}^{i=C-1} B_{\text{FEC}\,i}, \tag{6}$$

where C is the total number of classes, $B_{\text{FEC}\,i}$ is the bandwidth allocated to class i, and B_{FEC} is the total bandwidth allocated to FEC. We use a simple method to allocate FEC bandwidth to different classes at the sender side as shown in Figure 3. For simplicity, we assume that all data classes have a known, predefined, required loss rate, specified by the user. We start by allocating enough bandwidth to the most important class until the desired packet loss rate is achieved. We then successively move on to the less important classes until FEC budget is exhausted.

Table 1 shows the PSNR of a video sequence described in detail in Section 6, using a UEP scheme as described above, and an equal error protection (EEP) scheme. Block loss rate P_{BLER} varies from 3% to 12%. Of the total bandwidth, 10% is allocated to FEC. As expected, UEP performs considerably better than EEP.

4. PRIORITY-BASED ARQ

Priority-based ARQ (P-ARQ) is a hybrid-ARQ scheme taking into account different priorities of different data classes. Based on our packetization scheme, several blocks in the received application packet shown in Figure 2 may be lost. If the RS decoder cannot recover the errors, a P-ARQ algorithm is run to decide whether or not to send an ARQ. There are two steps for the P-ARQ algorithm to decide whether to retransmit a lost block. Both steps ensure that the retransmitted packets are not "late" by the time they arrive at the receiver. In the first step, the receiver checks

$$\text{RTT} < T_{\text{buff}} \tag{7}$$

to decide whether to send an ARQ packet. In the above equation, RTT is the round trip time, and T_{buff} is the display deadline of the frame with the lost block. If condition (7) is satisfied, an ARQ packet is sent back with the information specifying the location of the lost blocks and T_{buff}. In the second step, the sender analyzes the ARQ packet and checks the following conditions in order to decide whether a packet is to be resent:

$$\begin{gathered} \eta > 0, \\ \text{RTT} + T_{\text{sender}} < T_{\text{buff}}, \end{gathered} \tag{8}$$

where η is the remaining number of retransmissions for the lost block, and T_{sender} is the time difference between the time the sender receives the ARQ and the time it resends out the lost block. The parameter η is initially set based on the class of data block and is decremented whenever the data block is retransmitted. Specifically, initially η is set to $0, 1, 2, 3$ for classes $3, 2, 1, 0$, respectively. For an ARQ, if conditions (8) are satisfied, the lost block is eligible for retransmission.

Suppose the number of lost blocks in an application layer packet is R. Since there is a level of redundancy among the RLP packets in an application layer packet, in many situations, the sender does not need to resend all these R blocks. Specifically, it suffices for the sender to send $[R - (n - k)]$

TABLE 2: The retransmission bandwidth of our scheme versus the traditional scheme.

PLR	3%	6%	9%	12%
RS(10, 9)	4.37	3.65	3.19	2.74
RS(10, 8)	4.62	3.90	3.68	3.11

packets with an additional level of n/k FEC redundancy, that is,

$$T = (R - (n - k)) \times \frac{n}{k}. \tag{9}$$

After selecting blocks to be retransmitted, the sender packetizes them together with other video data to make an application packet.

The receiver sends a P-ARQ packet only after it receives an entire application packet. The delay induced by the application layer feedback is larger than that induced by RLP layer feedback. Specifically, if the ith RLP packet among the total of $(p + q)$ packets is lost, the induced delay is given by

$$T_{\text{delay}} = \frac{(p + q - i) \times \text{sizeof}(\text{RLP})}{B}, \tag{10}$$

where sizeof(RLP) is the RLP packet size in bits, and B is the channel bandwidth in bps.

5. SIMULATIONS AND DISCUSSION

We evaluate our proposed scheme using simulations on a socket-based test bed. We have encoded the video sequence "foreman" in CIF format, at 7.5 frames per second, with MPEG-4 encoder using one I frame followed by 14 P frames in one group of pictures (GOP). We have chosen a quantization step size of 8 and a video packet length of 960 bits. Packet-based RS code is used as FEC. Every application packet is mapped onto 12 RLP packets at the link layer. The network bandwidth is 256 kbps. We assume that the end-to-end RTT is 300 milliseconds, the wireless link layer RTT is 150 milliseconds, and the reverse channel bandwidth is always available for the negative acknowledgements (NAKs) and ARQs.

5.1. Comparisons against traditional schemes

In this section, we compare our scheme against traditional schemes, where the network does not pass the corrupted data along to the application layer. In this case, one RLP packet loss may cause the loss of an entire application packet, resulting in the retransmission of the entire application packet. We show the effectiveness of our scheme in bandwidth utilization assuming (a) no retransmission at RLP layer, and (b) maximum of one retransmission for every application packet. Every application packet consists of 10 blocks of data in its payload. Table 2 shows the retransmission bandwidth utilization of our scheme versus the traditional scheme under different channel conditions, with RLP layer packet loss rate ranging from 3% to 12%. Each figure in the table indicates the ratio between the number of retransmitted blocks in the traditional scheme and the number of retransmitted blocks in our scheme. As seen, our scheme reduces the retransmission bandwidth by a factor of 3 to 5.

5.2. Comparisons against RLP layer retransmission scheme

Our proposed scheme in this paper can adaptively transmit different classes of data according to the RLP packet loss rate, receiver buffer size, and priority of classes. Specifically, the sender selectively drops the lowest priority of data in poor channel conditions when the display deadline cannot be met. Alternatively, consider an RLP layer retransmission (RLR) scheme in which the RLP layer does not adapt itself to the network conditions and keeps retransmitting up to a maximum number of allowed retransmissions. The following simulations show that our scheme outperforms RLR schemes. In our scheme, we assume application layer FEC adaptation to channel conditions, application layer retransmissions based on data classes, and no RLP level retransmissions. In RLR, we assume a maximum of three RLP layer retransmissions, RS(10, 9) for FEC at the application layer, and no adaptation in the application layer.

Figure 4 shows the video quality using our scheme versus RLR under different packet loss conditions. In our scheme, since lowest-priority packets are dropped in poor channel conditions, only a few frames are affected. However with the RLR, high-priority data have the same loss rate as low-priority data, resulting in severe performance degradations. Figure 4a shows the PSNR as a function of RLP packet loss rate, and Figure 4b shows the PSNR as a function of frame number with loss rate of 6%. The dips in RLR scheme in Figure 4b are due to errors in I frames propagating across the GOP. The dips in our scheme are due to data corresponding to last frames in a GOP being dropped.

Figure 5 compares two scheduling schemes at the application layer for 4% RLP packet loss rate. In Figure 5a, the number of dropped packets within a GOP equals the number of retransmitted packets. As seen, this results in the data corresponding to later P frames in a GOP being dropped. This is due to our simplistic scheduling algorithm in which the data for successive frames are sent sequentially. An alternative would be to send higher-priority classes of a later P frame before lower-priority classes of earlier P frames. In Figure 5b, if the number of retransmitted RLP packets for GOP i is N_i, we drop $N_i/3$ packets in GOP i, and $N_i/3$ packets in each of GOP $i + 1$ and $i + 2$. Since the wireless channel is bursty, this distributes the loss among different GOPs, resulting in smaller PSNR variations across frames.

5.3. Our in-packet FEC scheme versus traditional cross-packet FEC scheme

In our scheme, FEC is done within one application layer packet, but at the granularity of RLP packets. Traditional cross-packet FEC encodes k application layer packets and generates $n - k$ parity packets. In this section, we compare our in-packet FEC scheme with the traditional cross-packet

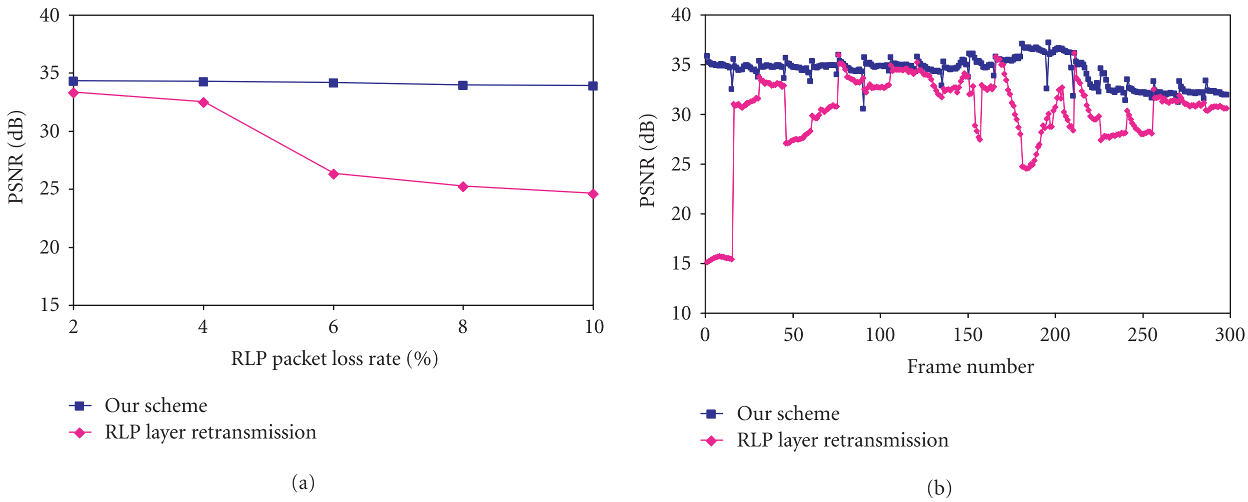


FIGURE 4: The video quality of our scheme versus RLR. (a) Average PSNR of the test video sequence at different RLP loss rates. (b) PSNR as a function of frame number at RLP loss rate 6%.

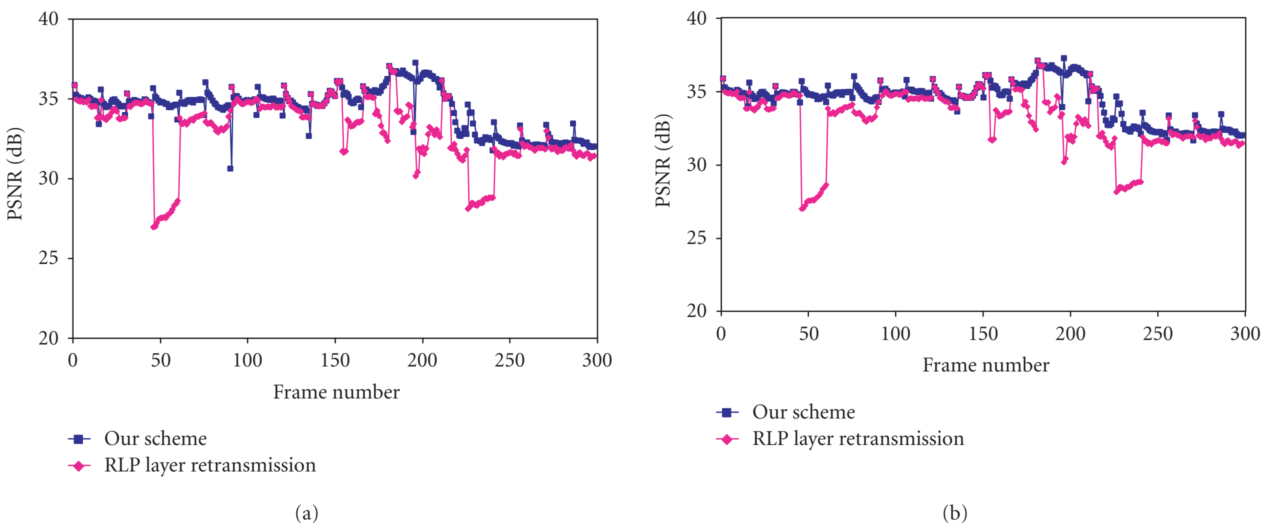
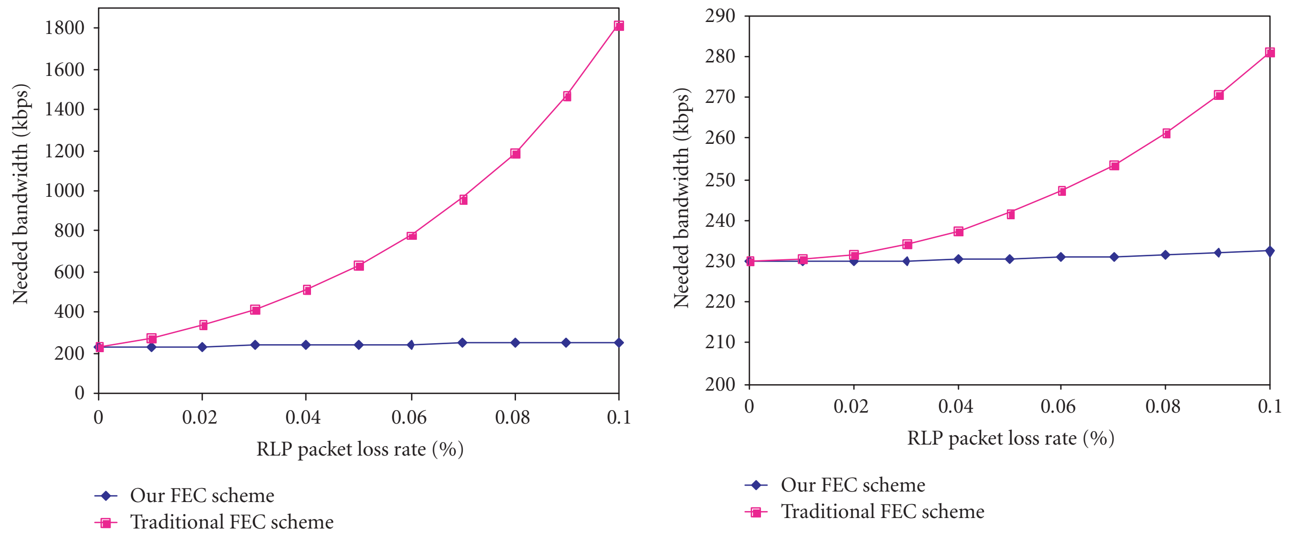


FIGURE 5: The video quality of our scheme versus RLR with active drop packets.

FEC scheme in terms of bandwidth utility at different RLP layer loss rate. The video bit stream is MPEG-4 encoded at 230 kbps. RS(N, 20) is used for both schemes, no application layer retransmission is used, N is determined based on channel conditions, and RLP layer retransmission is set to no and one time, respectively. Figure 6 shows the needed bandwidth of the two FEC schemes to fully recover the lost packets at the application layer. Since our scheme is granular at RLP layer, an RS(23, 20) can recover up to 13% RLP layer loss rate. The needed bandwidth utility of our scheme slightly moves from 230 kbps at 0% RLP loss rate to 255.6 kbps at 10% RLP loss rate. On the other hand, for the traditional scheme, one RLP packet loss can cause the whole application packet loss. Thus, the application layer packet loss rate is much higher than RLP layer, the needed bandwidth exponentially grows from 230 Kbps at 0% RLP loss rate to 1817 kbps at 10% RLP loss rate as shown in Figure 6a. In Figure 6b, the application layer packet loss rate is greatly reduced with one RLP layer retransmission. For our scheme, the needed bandwidth increases from 230 kbps at 0% RLP loss rate to 232.3 kbps at 10% RLP loss rate. For the traditional scheme, the needed bandwidth is up from 230 kbps at 0% RLP loss rate to 281 kbps at 10% RLP loss rate. Based on Figure 6, as the RLP loss rate increases, our scheme outperforms much more than the traditional scheme, which is what we expect in an error-prone wireless system.

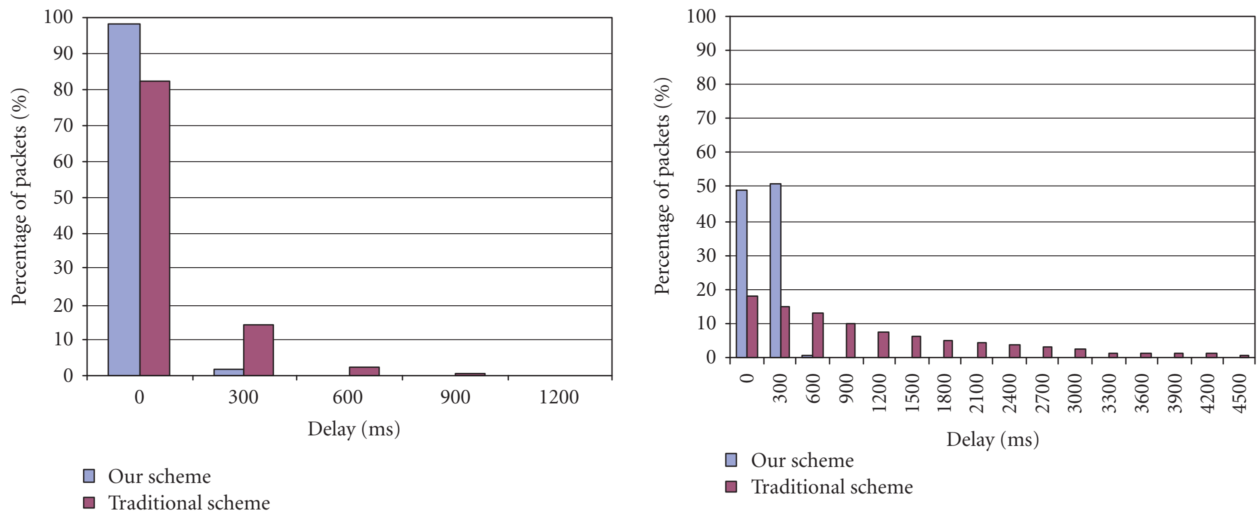
5.4. Delay analysis

In this section, we analyze the delay of our scheme versus the traditional scheme in different channel conditions.



(a) (b)

Figure 6: Needed bandwidth of our scheme versus traditional scheme to fully recover the lost packet in the application layer: (a) no RLP layer retransmission; (b) one RLP layer retransmission.



(a) (b)

Figure 7: Analysis of the delay of our scheme versus the traditional scheme, both with no RLP layer retransmission: (a) 1% RLP layer loss rate; (b) 8% RLP layer loss rate.

The delay is defined as the time difference between the packet arrival time and the expected time that a packet should arrive at. MPEG-4 encoded video sequence bit rate is at 230 kbps. A fixed RS(21, 20) FEC is applied to both our scheme at the block level and the traditional scheme in the application packet level. A block size is the same as an RLP packet payload size which is 24 bits. Thus, the total number of application layer packets of this encoded video sequence is around 2800. Whenever the FEC codes cannot recover the lost blocks or packets, an application layer ARQ is triggered. In Figure 7, our scheme and the traditional scheme are compared at 1% and 8% RLP layer loss rate, respectively. No RLP layer retransmissions are set at both schemes. For our scheme, FEC is applied within an application layer packet, but at the granularity of the RLP layer. It can recover up to 4.76% RLP layer losses. Figure 7a shows that most of the packets of our scheme arrive at their expected arrival time except several delayed packets due to channel burst loss. Since any RLP layer packet loss can result in a whole application layer packet loss in the traditional scheme, the traditional scheme needs more

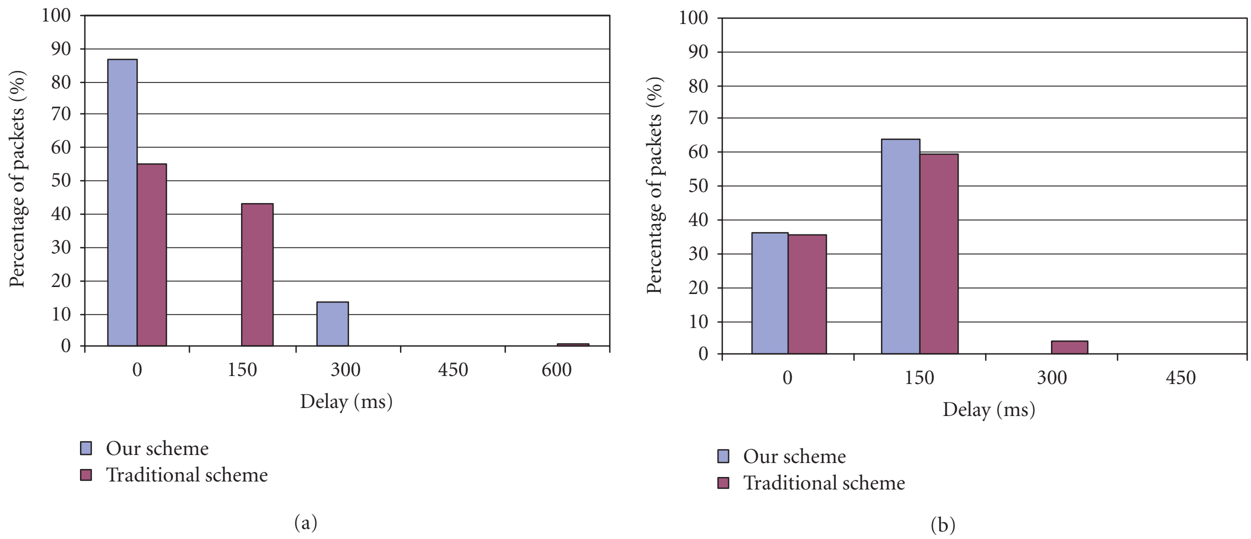


Figure 8: Analysis of the delay of our scheme versus the traditional scheme at (a) 3% RLP layer loss rate, with no RLP retransmission for our scheme and one RLP layer retransmission for the traditional scheme; and at (b) 5% RLP layer loss rate, with one RLP layer retransmission for our scheme and up to three times RLP layer retransmissions for the traditional scheme.

retransmissions to recover the losses as shown in Figure 7a, more than 400 packets need to be retransmitted once, and 2 packets even need to be retransmitted four times. At 8% RLP layer loss rate, both our and the traditional FEC schemes cannot recover the lost packets by using only the FEC codes, thus, application layer retransmissions are triggered at both schemes as shown in Figure 7b. Our scheme can recover most of the lost packets with one retransmission due to the cross-layer characteristics. At 8% RLP loss rate can cause severe application packet losses in the traditional scheme, therefore, more retransmissions are triggered and result in a long delay as shown in Figure 7b.

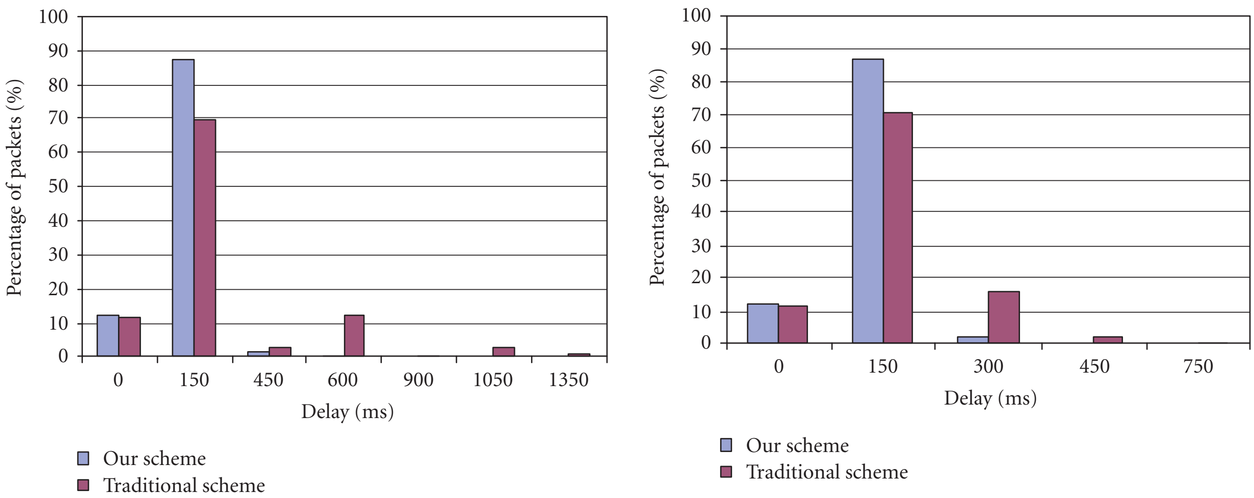
Figure 8a compares the delay of our scheme versus traditional scheme at 3% RLP layer loss rate. Since there are no RLP layer retransmissions at our scheme, around 300 application packets need to be retransmitted once at the application layer due to channel burst losses beyond the recovery capability of the FEC codes. At 3% RLP layer loss rate, most of the lost RLP layer packets can be recovered with one RLP layer retransmission, so only 27 application layer packets need to be retransmitted twice at the traditional scheme. We can theoretically say that a 3% RLP layer loss rate with one retransmission can be reduced to 0.09%. In Figure 8b, we further set the RLP layer loss rate to 5%. RLP layer retransmission is set to one time for our scheme and up to three times for the traditional scheme. Our scheme can recover all the lost packets with the support of one RLP layer retransmission and in-packet block-based FEC scheme. In the traditional scheme, nearly 150 packets need to be retransmitted twice at the RLP layer, even with the same amount of FEC support at the application layer.

In Figure 9, we compare our scheme and the traditional scheme at a higher RLP layer loss rate, 10%, with RLP layer retransmissions. If only one RLP retransmission is given, our scheme can recover most of the packets with the support of FEC codes, except that very few packets need to be retransmitted at the application layer as shown in Figure 9a. One RLP layer retransmission cannot recover the loss at the application layer, given a higher RLP layer loss rate. The traditional scheme needs several application layer retransmissions in Figure 9a. In Figure 9b, up to three RLP layer retransmissions can recover almost all the lost RLP layer packets. Both our and the traditional schemes are with short delays, but our scheme is better than the traditional one, as almost no application layer retransmission is needed.

Based on the analysis of the delay, our scheme outperforms the traditional scheme if the same condition is given or even a tighter condition is given to our scheme. The reason is that our adaptation scheme is done at the application layer, but at the granularity of the RLP layer. Our approach efficiently combines the flexibility and programmability of the application layer adaptations, with low delay and bandwidth efficiency of link layer techniques. User studies indicate that users consider delays larger than 300 milliseconds not suitable for real-time video. From the simulations, we can conclude that our scheme is suitable for real-time video streaming. In some cases, our scheme can do even better with the support of one RLP layer retransmission.

6. CONCLUSIONS

In this paper, we have proposed application and channel adaptive scheme for video transmission over wireless networks. We have demonstrated the effectiveness of the application layer adaptivity combined with the RLP layer granularity. Future work involves use of fine grain scalable (FGS) video in conjunction with some of the schemes proposed in the paper.



(a) (b)

FIGURE 9: Analysis of the delay of our scheme versus the traditional scheme at 10% RLP layer loss rate and with RLP layer retransmissions. (a) RLP layer retransmission is set to one time; (b) RLP layer retransmission is set to up to three times.

ACKNOWLEDGMENTS

The author would like to thank the anonymous reviewers who provided excellent and thorough feedback that improved the quality of this paper. Thanks go to Professor Avideh Zakhor's supervision on the original work [11] and helpful comments on this paper. Also, thanks go to Professor Shivkumar Kalyanaraman's help in the final revise. This work is sponsored by AFOSR Award FDF49620-99-1-0062 and partly supported by ARO Award DAAD19-00-1-0559.

REFERENCES

[1] D. Qiao and K. G. Shin, "A two-step adaptive error recovery scheme for video transmission over wireless networks," in *Proc. 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 3, pp. 1698–1704, Tel Aviv, Israel, March 2000.

[2] J. Cai, Q. Zhang, W. Zhu, and C. W. Chen, "An FEC-based error control scheme for wireless MPEG-4 video transmission," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC '00)*, vol. 3, pp. 1243–1247, Chicago, Ill, USA, September 2000.

[3] M. Budagavi, W. R. Heinzelman, J. Webb, and R. Talluri, "Wireless MPEG-4 video communication on DSP chips," *IEEE Signal Processing Magazine*, vol. 17, no. 1, pp. 36–53, 2000.

[4] K. Stühlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, 2000.

[5] H. Zheng and J. Boyce, "Packet coding schemes for MPEG video over Internet and wireless networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC '00)*, vol. 1, pp. 191–195, Chicago, Ill, USA, September 2000.

[6] ETSI, "Digital cellular telecommunications system (Phase 2+); General packet radio service (GPRS); Mobile station (MS)-base station system (BSS) interface; Radio link control/medium access control (RLC/MAC) protocol," EN 301 349 v 8.3.1 (GSM 04.60 v 8.3.1 Release 1999), 2000.

[7] M. Zorzi, "Some results on error control for burst-error channels under delay constraints," *IEEE Trans. Vehicular Technology*, vol. 50, no. 1, pp. 12–24, 2001.

[8] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 278–290, 2004.

[9] S. Krishnamachari, M. VanderSchaar, S. Choi, and X. Xu, "Video streaming over wireless LANs: A cross-layer approach," in *Proc. IEEE Packet Video*, Nantes, France, April 2003.

[10] L. Larzon, M. Degermark, and S. Pink, "UDP lite for real time multimedia applications," Tech. Rep. HPL-IRI-1999-001, Extended Enterprise Laboratory, HP Laboratories Bristol, Bristol, UK, April 1999.

[11] Y. Shan and A. Zakhor, "Cross layer techniques for adaptive video streaming over wireless networks," in *Proc. IEEE International Conference on Multimedia and Expo (ICME '02)*, vol. 1, pp. 277–280, Lausanne, Switzerland, August 2002.

Yufeng Shan received the B.E. and M.E. degrees from Shandong University, China, in 1991 and 1994, respectively. From 1995 to 2000, he was a research faculty member at the Department of Computer Science, Shandong University. In 2000, he held a Visiting Researcher position at Planète Group, INRIA Sophia-Antipolis, France. From 2001 to 2002, he was a Research Engineer at the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. Since 2003, he has been with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, first as a Research Engineer and now he is working towards his Ph.D. His research interests include multimedia communications and networking.