Providing Adaptive QoS to Layered Video Over Wireless Local Area Networks Through Real-Time Retry Limit Adaptation

Qiong Li, *Member, IEEE,* and Mihaela van der Schaar, *Senior Member, IEEE*

***Abstract*—Robust streaming of video over 802.11 wireless LANs (WLANs) poses many challenges, including coping with packets losses caused by network buffer overflow or link erasures. In this paper, we propose a novel error protection method that can provide adaptive quality-of-service (QoS) to layered coded video by utilizing priority queueing at the network layer and retry-limit adaptation at the link layer. The design of our method is motivated by the observation that the retry limit settings of the MAC layer can be optimized in such a way that the overall packet losses that are caused by either link erasure or buffer overflow are minimized. We developed a real-time retry limit adaptation algorithm to trace the optimal retry limit for both the single-queue (or single-layer) and multiqueue (or multilayer) cases. The video layers are unequally protected over the wireless link by the MAC with different retry limits. In our proposed transmission framework, these retry limits are dynamically adapted depending on the wireless channel conditions and traffic characteristics. Furthermore, the proposed priority queueing discipline is enhanced with packet filtering and purging functionalities that can significantly save bandwidth by discarding obsoleted or un-decodable packets from the buffer. Simulations show that the proposed cross-layer protection mechanism can significantly improve the received video quality.**

***Index Terms*—Adaptive control, cross layer design, home communication system, multimedia communication, queueing analysis, wireless LAN.**

I. INTRODUCTION

REAL-TIME streaming of audiovisual content over the Internet is emerging as an important technology area in multimedia communications. Furthermore, with the advance of Wireless LAN (WLAN) technologies, this multimedia content will be redistributed in the future within the home to a variety of wireless devices [8]. However, as opposed to wired Internet streaming, to achieve a high level of acceptability and proliferation of wireless multimedia, in particular wireless video, several key requirements need to be satisfied in order to provide a reliable and efficient transmission: 1) easy adaptability to wireless bandwidth fluctuations due to co-channel interference, multipath fading, mobility, handoff, competing traffic, etc.; 2) robustness to partial data losses caused by the packetization of video frames and high packet error rate; and 3) support for heterogeneous clients with regard to their access bandwidths, computing capabilities, buffer availabilities, display resolutions and power limitations.

In recent years, many papers proposed various solutions addressing one or several of the previously mentioned requirements. In [12], Girod and Farber give an excellent review of the existing solutions for combating wireless transmission errors. While their focus is on cellular networks, most protection strategies presented can also be applied to the transmission of video over WLANs. The solutions discussed include adapting the modulation schemes in the physical layer (PHY) to obtain different bandwidth-power tradeoffs, combining Forward Error Correction (FEC) with interleaving as a classical solution for combating the effect of bursty errors, and employing closed-loop error control strategies, i.e., Automatic Repeat reQuest (ARQ), against error conditions that vary in a wide range. Moreover, the tradeoffs that can be made between reliability, throughput and delay in providing an optimal solution for wireless video are thoroughly described. Subsequently, various error resilience techniques for low bit-rate video are listed, again with an emphasis on the cellular video transmissions. The focus is on channel-adaptive source coding schemes that are useful when real-time channel feedback is available to the encoder. Importantly, joint consideration of network and application layers is mentioned as an interesting area for further research.

In [13], Shan and Zakhor present a novel integrated application-layer packetization, scheduling and protection strategies for wireless transmission of nonscalable coded video. The novel packetization mechanism proposed is aimed at constructing the application-layer packets in such a way that they are decomposed exactly into an integer number of equal-sized lower-layer packets. Subsequently, each MPEG-4 Simple Profile nonscalable bitstream is divided into four distinct classes having different effects to the overall video quality. The different classes are unequally protected using application-layer FEC or a novel hybrid ARQ scheme, denominated priority-based ARQ.

Cote *et al.* present in [14] a thorough survey of the different video-optimized error resilience techniques that are necessary to accommodate the nature of the compressed video bitstreams, which are very sensitive to bit errors and packet losses.

The contributions discussed until now focus mainly on the robust transmission of video over cellular networks. In [8], Majumdar *et al.* address the problem of resilient real-time video

Manuscript received January 14, 2003; revised August 2, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wenjun Zeng.

Q. Li was with Philips Research USA, Braircliff Manor, NY 10510 USA. He is now with BrainMedia, New York, NY 10010 USA (e-mail: qiongli@ieee.org).

M. van der Schaar is with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA 95616-5294 USA (e-mail: mvanderschaar@ece.ucdavis.edu).

Digital Object Identifier 10.1109/TMM.2003.822792

streaming over IEEE 802.11b WLANs for both unicast and multicast transmission. For the unicast scenario, a hybrid ARQ algorithm that efficiently combines FEC and ARQ is proposed. For the multicast case, progressive video coding based on MPEG-4 Fine Granularity Scalability (FGS) is combined with FEC. Hybrid ARQ schemes, where the rate of the associated FEC is adaptively changed based on the underlying channel conditions, have also been presented by Wang and Zhu in [15] and by Ma and Zarki in [16]. However, it should be pointed out that the protection strategies described in these papers are implemented at the application layer and do not exploit the mechanisms available in the lower layers of the protocol stack.

In summary, the research efforts in the area of robust wireless transmission have mainly focussed on enabling adaptive error-control strategies at the application layer. However, in existing WLAN environments, different protection strategies are available at the various layers of the protocol stack that allow for different tradeoffs among throughput, reliability and delay, such as 1) switching among different modulation and channel coding techniques at the physical layer; 2) applying retransmission and FEC at the MAC layer; 3) applying FEC, ARQ, or hybrid ARQ along with error resilient video coding schemes and error concealment strategies at the application layer; 4) performing packetization optimizations at the various layers.

Currently, these various protection strategies are optimized separately at each network layer. However, a joint consideration is desirable in order to provide an optimal overall performance for the transmission of video. Particularly, in in-home wireless networks, because the transmission is performed in a confined environment and infrastructure, we can pursue a cross-layer[1] optimized protection strategy for providing highly efficient real-time transmission of video, which may not be feasible in large scale legacy networks.

In this paper, we focus on the efficient and robust transmission of video over wireless LANs. Our focus is mainly on one particular type of WLANs, namely, IEEE 802.11 that potentially can be deployed for redistribution of video within the home. Specifically, we introduce a simple and highly effective cross-layer protection mechanism that employs layered coding for the compression of the video combined with adaptive protection strategies, like priority queueing, selective dropping, and unequal protection through real-time MAC-layer retry-limit adaptation. These mechanisms can operate independently, but when combined together, they can considerably improve the robustness and efficiency of video transmission over WLANs.

The adaptive cross-layer protection strategies proposed in this paper can be applied to any prioritized or scalable video coded bitstream, such as nonscalable MPEG-4 or H.264 coded bitstreams, in which different priorities are assigned to the various frame types, data-partitioned bitstreams, MPEG-2 and MPEG-4 hybrid spatio–temporal scalable bitstreams, fully scalable wavelet video bistreams etc. To fulfill the requirements for wireless video identified before, and in particularly to address how adaptive quality-of-service (QoS) can be provided to layered video over WLANs, in this paper we employed MPEG-4 Fine-Granularity-Scalability (FGS) for the compression of the video data (as in [8] and [17]), because it can provide easy adaptation to bandwidth variations and device characteristics [11].

The proposed adaptive cross-layer protection strategy is motivated by the observation that in WLANs, packets are lost whenever a link error occurs and the packet reaches its retry limit, or whenever buffer overflow occurs. Wireless links normally use retry as a means of protection against link errors. The probability that a packet is dropped due to a link error decreases as the retry-limit setting is increased. However, when the retry limit is set too high, the network interface buffer drainage speed slows down, thereby increasing the probability of buffer overflow. Hence, the operation point at which the overall packet loss due to buffer overflow and link errors is mimimal can be determined based on the flow rate, retry-limit setting and channel conditions. Thus, given a certain flow rate and channel condition, the retry-limit setting that minimizes the packet-losses and leads to the highest video quality can be determined. In this paper, we present several heuristic algorithms that determine and adapt in real-time the optimal retry limit corresponding to the mimimum packet loss, referred to as real-time retry-limit adaptation (RTRA) schemes, for both single-queue and multi-queue cases.

The proposed cross-layer protection system can provide not only priority delivery services, but also unequal error protection to the different video streams, by adapting different retry-limit settings in the MAC for the multiple queues containing the different priorities video streams. By adapting the MAC using the proposed RTRA algorithm based on packet rates and channel conditions, we provide adaptive QoS that can cope with time-varying wireless channel and lead to a higher video quality. A systematic method for finding out the optimal retry-limit configurations for different priority queues is also presented.

Experiments show that these adaptive protection mechanisms when synergistically combined can significantly improve the video quality over WLANs. The design rationale of these mechanisms as well as their performance analyzes are detailed in this paper.

The rest of this paper is organized as follows. Section II starts with a brief introduction to the retransmission mechanism available in IEEE Std. 802.11 MAC and then presents a theoretical analysis on how the MAC retry-limit setting may effect buffer overflow and link drops. A heuristic algorithm for single-queue retry-limit adaptation that can minimize overall packet losses is also developed in this section. Section III describes the priority queueing, and related packet filtering and purging. A systematic optimization method for configuring the MAC retry limits for the multiqueue case is also presented in this section. Section IV describes the simulation results that illustrate the effectiveness of the proposed mechanisms. We conclude the paper in Section V.

II. Retry-Limit Adaptation

Due to co-channel interference, multipath fading and mobility, wireless links are generally unreliable, having high packet

[1] In this paper, the word "layer" has been used in two different contexts: network protocol layer and scalable video layer. In the remainder of this paper, for simplicity, we used the term "cross-layer" instead of "cross network protocol layer".

error rates and fluctuating overall bandwidth. For transmission over IEEE 802.11 WLANs [9], there are two different mechanisms that can be deployed by the wireless link to combat the time-varying conditions: switching among different PHY modes, each with a different modulation scheme and a raw data rate, or performing link layer (or the multiple access control (MAC) protocol layer) retry. In this paper, we focus our discussion on the retry mechanism. According to the IEEE Std. 802.11 MAC, when a transmitted packet is not acknowledged properly, retries can be performed and repeated until a certain limit is reached. Packets are dropped when they reach their retry limits. Retry is an efficient means to improve the reliability of the link. In the current standard, this retry limit is normally configured statically, and there is no recommendation or guidance on how the retry limit can be adapted based on the channel conditions or the workload. However, as mentioned in the introduction, the retry limit may affect not only the link packet erasure rate but also the network buffer overflow rate.

In the following, we analyze how MAC retry limit may effect buffer overflow using queueing models and ns-2 [1] simulations. We then develop a retry-limit adaptation algorithm that can balance the buffer overflow and link erasure to minimize the packet losses incurred by the video stream.

A. *Buffer Overflow Versus Link Erasure*

Multimedia packets can be lost in WLANs due to either buffer overflow or link erasure. When User Datagram Protocol (UDP) [20], an open-loop transport protocol without any built-in flow control mechanism, is used for the transmission of video, this will blindly dump as many data packets as requested by the application to the network buffer, independent of the buffer occupancy status. When the buffer is full, arriving packets will simply be dropped, generating an overflow event. When the wireless link experiences high packet error rates or transmission collisions among stations sharing the same wireless medium, the packets require more retransmissions on average in order to be transmitted correctly at the receiver, thereby reducing the drainage speed of the network buffer, and increasing the probability of buffer overflow.

This problem can be modeled using a queueing system with bursty arrivals and a time-varying service rate. To capture the queueing dynamics, a stochastic traffic model that can describe the variable bit-rate characteristics of the video, such as the multistate Markov Chain traffic model [10], and a general distribution that can account the randomness introduced by link retransmissions under varying channel, are necessary. However, performing the precise queueing analysis for the addressed retry limit adaptation problem in such a generic setting is beyond the scope of this paper, and we would like to leave it for future study. In this paper, we present a simplified analysis based on two different queueing models—the fluid model and the M/G/1 model under static channel error conditions.

1) Fluid Model: Recent studies on fluid model for communication networks show that queueing dynamics is affected by the stochastic characteristics of arrival rate [3]–[5]. However, in the following analysis, we assume static channel conditions and constant arrival rate, therefore a simple instance of fluid model as introduced in [7] is adopted. Particularly, we assume packets arrive at a constant rate with uniform (packet) size, and overflow (i.e., packets are dropped before they are put on the link) will happen when the workload exceeds the link capacity. Let L_r be the link retry limit, and P_e the packet error rate (PER) of the link (without retry), then the mean number of transmissions for a single packet until it is either successfully received or it reached its retry limit can be calculated as

$$\begin{aligned} s(L_r, P_e) &= 1\cdot(1-P_e) + 2\cdot P_e(1-P_e) + \cdots \\ &\quad + L_r P_e^{L_r-1}(1-P_e) + (L_r+1)P_e^{L_r} \\ &= \frac{1-P_e^{L_r+1}}{1-P_e}. \end{aligned} \tag{1}$$

Let λ be the arrival rate (packets/s). To simplify the analysis, we assume every transmission takes equal time. In the fluid model, we calculate the overflow rate as

$$p_B(L_r, P_e) = \frac{\lambda s(L_r, P_e) - C}{\lambda s(L_r, P_e)}, \tag{2}$$

where C is the service rate of the link (packets/second)[2] Equation (2) shows that overflow occurs only when $\lambda s > C$. By substituting (1) into (2), we have

$$p_B(L_r, P_e) = 1 - \frac{1}{\rho(P_e)}\frac{1}{1-P_e^{L_r+1}} \tag{3}$$

where $\rho(P_e) = \lambda/C(1-P_e)$ is the effective utilization factor of the link. If $\rho > 1$, then $p_B > 0$.

If we assume that the wireless link is a memoryless packet erasure channel [8], such that the packets are dropped independently, we can calculate the link packet erasure rate (i.e., the packet drop rate after L_r unsuccessful retries) as

$$p_L = P_e^{L_r+1}. \tag{4}$$

The overall loss rate $p_T(L_r, P_e)$, which is defined as the sum of the overflow rate and the link erasure rate[3], becomes

$$\begin{aligned} p_T(L_r, P_e) &= p_B(L_r, P_e) + p_L(L_r, P_e) \\ &= 1 - \frac{1}{\rho(P_e)}\frac{1}{1-P_e^{L_r+1}} + P_e^{L_r+1}. \end{aligned} \tag{5}$$

Equations (3) and (4) show that, when P_e is fixed, $p_B(L_r)$ monotonically increases with L_r, while $p_L(L_r)$ decreases at the same time. Hence, we can find L_r such that $p_T(L_r)$ is mimimized given a fixed value of P_e. We temporarily relax the discrete constraint of L_r, assuming it is a continuous variable. L_r can then be found by solving the equation $dp_T(L_r)/dL_r = 0$, which leads to:

$$L_r = log_{P_e}\left(1 - \frac{1}{\sqrt{\rho}}\right) - 1.$$

Interestingly, we find that $p_B(L_r) = p_L(L_r)$ at this point, implying that the optimal L_r is located at the intersection point of the two functions—$p_B(L_r)$ and $p_L(L_r)$. The optimized retry limit can then be obtained by rounding L_r to the closest integer.

[2] We omit the buffer effect here, assuming the channel is static with a fixed P_e.

[3] We assume both p_B and p_L are relatively small such that they can be added together to approximate the total loss rate.

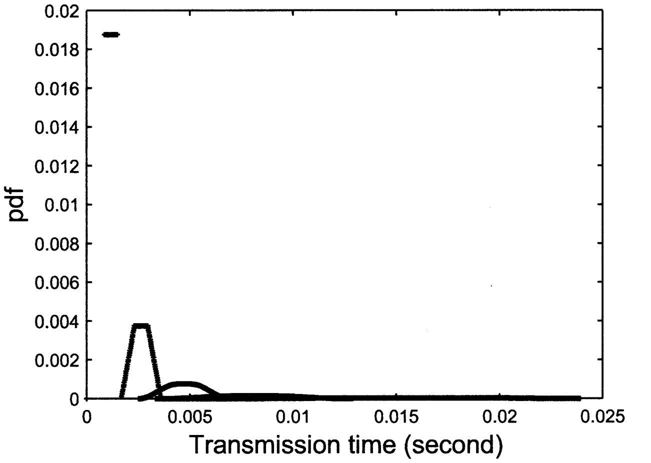


Fig. 1. The pdf of 802.11b frame transmission time when $P_4 = 0.4$ and $L_r = 4$.

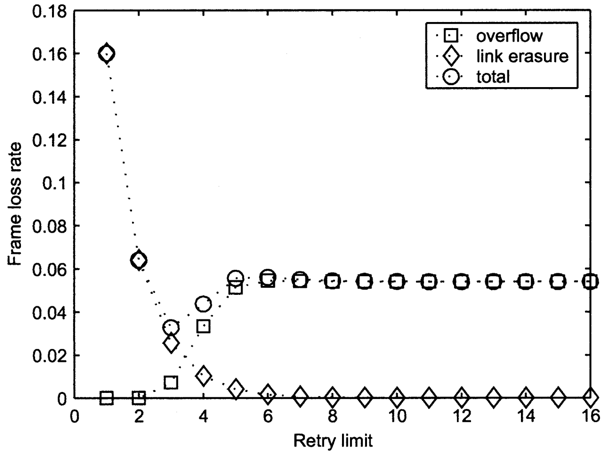


Fig. 2. Packet loss rate versus MAC retry limit obtained through M/G/1 queue analysis.

Therefore, we conjecture that the retry limit optimization can be formulated as

$$\forall L_r \text{ that } \min_{L_r} |p_L(L_r) - p_B(L_r)|. \tag{6}$$

In other words, the optimal retry limit will always be the one that can strike a balance between overflow loss and link loss. This conjecture is also supported by the M/G/1 queueing analysis as we will discuss in Section II-A2.

2) M/G/1 Model: We apply the /M/G/1 queue model [6] to the buffer overflow analysis. The /M/G/1 queue is a single-server system, assuming Poisson arrivals and arbitrary service-time distribution. In particular, we assume the probability distribution function (pdf) of inter-arrival time as

$$a(t) = \lambda e^{-\lambda t} t \leq 0,$$

where λ is the average arrival rate (packets/second), and the service-time pdf as $f(t)$, the same as the frame transmission time over a 802.11 wireless link. $f(t)$ can be calculated as

$$f(t) = \sum_{n_r=0}^{L_r} f(t|n_r)P(n_r) \tag{7}$$

where $P(n_r)$ is the probability that this transmission takes n_r times of retransmission to succeed or reach the retry limit L_r; $f(t|n_r)$ is the pdf of t when n_r retransmissions have been performed. To show the complexity of $f(t)$, we calculated an instance of $f(t)$ as shown in Fig. 1, assuming that the MAC parameters are set as recommended in IEEE Std. 802.11b, and that there are only two mobile nodes in the network: one sender and one receiver. Due to the discrete nature of the random backoff timer deployed by the MAC protocol to perform CDMA/CA operations, t is discrete, thus $f(t)$ is also not continuous.

This M/G/1 queueing system can be modeled by an embedded Markov chain with state transition occurring at the end instant of every frame transmission (including all retries that may be performed). The states of the chain represent the observed queue sizes at these embedded instants (for a general discussion on M/G/1, We refer the reader to [6]). The numerical solution to the queue-length probability distribution of the queueing system can be easily calculated under the assumption of a finite buffer size. The buffer overflow rate equals the probability that a arriving packet observes a full queue which is the probability that the queue length equals the buffer size.

To quantitatively investigate how the different retry limit settings affect the buffer overflow rate under the M/G/1 model, we set the buffer size $B = 50$, and fix P_e as 0.4 and λ as 3.52 Mps. The calculated overflow rate is shown in Fig. 2 together with the calculated link erasure rate and total loss rate based on (4) and (5). We observe that, even though the absolute buffer overflow rate calculated with the M/G/1 model is slightly different from the prediction of (3) (the fluid model), the conjecture as formulated in (6) is still valid with the M/G/1 queue, indicating that queueing dynamics is strongly dominated by the first-order statistics of arrival rate and service rate. Furthermore, we show in Section III that they are also consistent with the simulation results obtained using the ns-2 network simulator.

3) Simulations: Using the ns-2 network simulator, we simulate an ad-hoc WLAN network having two stations, a sender and a receiver. The transmission of a constant bit rate (CBR) data flow of 3.52 Mbps and having packet sizes of 1000 bytes is simulated. The network buffer size is set to 50 packets and the link capacity is set to 11 Mbps having an error rate of $P_e = 0.4$[4] The other MAC parameters, except the retry limit, are set as recommended by the IEEE Std. 802.11b [9]. UDP is used for transport, so that the traffic characteristics of the generators are not changed when packets enter the interface queue. Subsequently, we simulate 16 times the transmission of the CBR data-flow, each time having a different link retry limit. Each simulation run spans over 400 s. The measured overflow rate, link erasure rate and the sum of these two are shown in Fig. 3. From the figure, the following observations can be made.

- The correlation between the retry limit and the packet losses predicted by the fluid model and queueing analyzes (detail omited in this paper) is maintained, namely, there exists a optimal retry limit setting that minimizes the

[4] The buffer size is selected according to the settings that are usually seen in Unix OS kernels; the link speed $(= 11 \text{ Mbps})$ and the $P_e (= 0.4)$ are chosen according to the 11 Mbps CCK PHY mode in the IEEE Std. 802.11b and the field measurement conducted in [2].

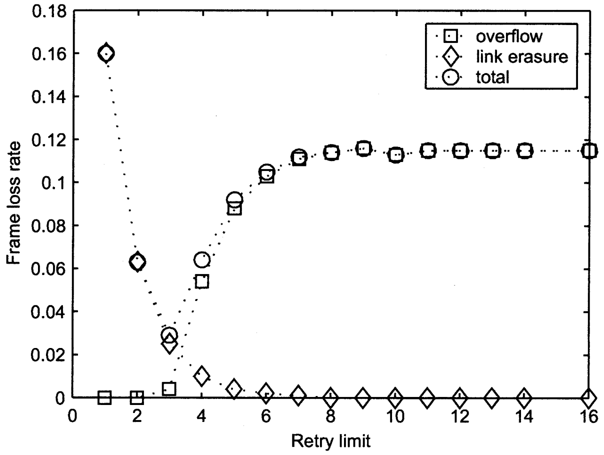


Fig. 3. Simulated packet loss rate versus MAC retry limit under CBR traffic.

overall packet losses due to both overflow and link erasure. The optimal setting takes place where the two types of losses are balanced, as formulated by (6).

- Retry-limit settings can significantly affect the overall packet loss rates. In the simulated case, a small deviation from the optimal setting could cause a substantial increase in the packet loss rate.

Additional simulations reveal that the optimal retry limit is not a fixed value, but rather is determined by the packets arrival rate and the channel condition, as shown in Fig. 4(a) and (b). Fig. 4(a) shows that, when P_e is fixed at 0.4, while the CBR packet generator's output rate changes from 2.71 Mbps to 3.2 Mbps, the optimal retry limit also changes from 6 to 3. Fig. 4(b) shows that, when CBRs rate is kept constantly at 3.01 Mbps, while the P_e changes from 0.35 to 0.45, the optimal retry also changes from 5 to 3.

Additional simulations, omitted here due to space limitations, show that when the sender's traffic source switches from CBR to other traffic (e.g., Exponential, Pareto etc.), different optimal settings are obtained for the retry limit since the queueing process is effected by the traffic characteristics.

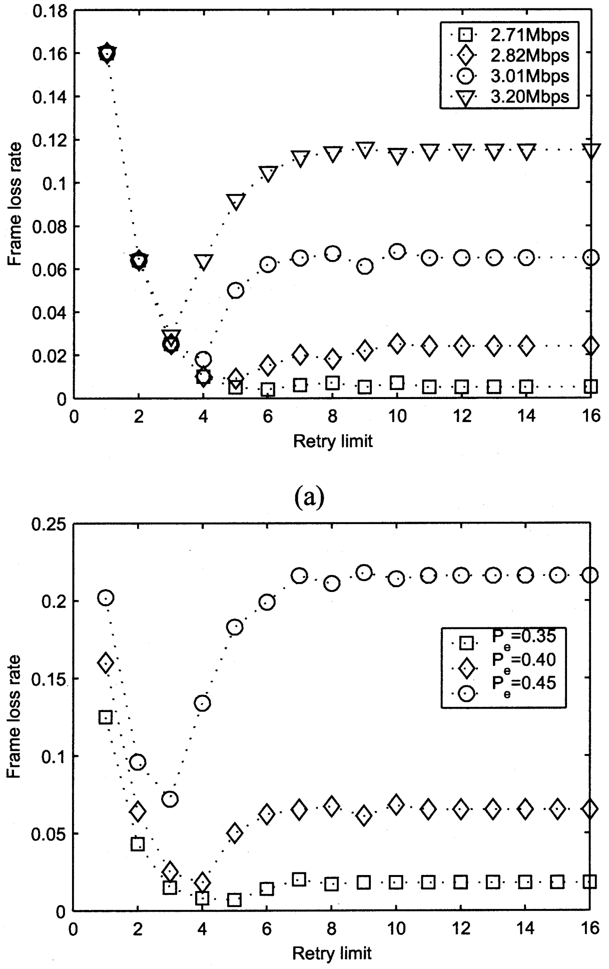
The significant gain in performance that can be obtained if the retry limit is set to an optimal value, motivated us to seek adaptive algorithms that can track the optimal setting of the retry limit when the traffic characteristics and wireless channel conditions change.

B. Adaptation Algorithm

The formulation of the retry-limit optimization problem as described in (6) indicates that the optimal point should be the nearest one to the intersection of $P_L(L_r)$ and $P_B(L_r)$ (see Fig. 3). This observation leads to the following simple algorithm for the retry-limit adaptation.

- Both the network queue and the MAC layer keep monitoring the overflow rate P_B and the packet error rate P_L.
- if $P_B < P_L$, increase the retry limit L_r; if $P_B > P_L$, decrease L_r.

The basic idea behind this simple algorithm is to maintain the retry limit close to the intersection between $P_L(L_r)$ and $P_B(L_r)$. A fine-tuned algorithm that has been tested by extensive simulations is presented in Fig. 5. The reasoning behind the design of this algorithm is also inserted in the algorithm as comments.[5]



(a)

(b)

Fig. 4. Packet loss rate versus MAC retry limit (a) under different incoming traffic rate; (b) under different channel error rate P_e.

The performance of this algorithm is controlled by four parameters: P_{Th_1}, R, P_{Th_2} and θ (see Fig. 5 for the definitions of these parameters). A good instance of their settings obtained through simulations is: $(P_{Th_1}, R, P_{Th_2}, \theta) = (0.0001, 4, 0.01, 10)$. For the convenience of description, this algorithm is referred to as real-time retry-limit adaptation (RTRA) in the rest of this paper.

To test our RTRA algorithm, we assumed a time-varying channel error model that is described by a three-state Markov chain with the state transition matrix $\mathbf{A_e}$ and the state duration T_e.[6] Each state of this model is associated with a PER. The PER vector is defined by $\mathbf{P_e} = [P_{e_1}\ P_{e_2}\ P_{e_3}]$. During each

[5] To calculate P_B and P_L, both the IP queue and the MAC independently deploy an observation window of 50 packets and a low-pass filter of the form $\overline{X}_n = 0.75\overline{X}_{n-1} + 0.25X_n$ to produce smoothed measurements. The adaptation algorithm is triggered whenever either the IP queue or the MAC generates a new measurement. The window size and the filter coefficients were heuristically determined (using numerous simulations) in such a way that the proposed adaptation algorithm can respond to the time-varying channel in a timely manner.

[6] This error model is available in ns-2 simulator, and commonly used to simulate time-varying wireless link behavior when studying higher layer protocol performances [1].

```
while (1) {
        periodically calculate and compare $P_B$ and $P_L$
        //if the sum of $P_B$ and $P_L$ is small ($< P_{Th_1}$),
        //as in the case of good channel conditions or
        //traffic reduction, the MAC decreases $L_r$ until
        //a pre-configured minimum $R$.
        if( $P_B + P_L < P_{Th_1}$ ) {
            if( $L_r > R$ ) {
                $L_r$ decreases by 1;
                continue;
            }
        }
        //if the difference between $P_B$ and $P_L$ is small
        //($< P_{Th_2}$), no action needs to be taken since the two loss rates
        //are already well balanced.
        if ($|P_B - P_L| < P_{Th_2}$)
            continue;
        //normal adaptation
        if ($P_B < P_L$)
            $L_r$ increases by 1;
        else {
            $L_r$ decreases by 1;
            //if $P_B >> P_L$, as in the case when the traffic
            //intensity suddenly increases, the MAC needs to reduce
            //$L_r$ faster in order to releave congestion.
            if( $P_B > \theta \cdot P_L$ ) {
                $L_r$ decreases once more by 1;
            }
        }
}
```

Fig. 5. Retry-limit adaptation algorithm.

state, packets are dropped randomly by the channel with the corresponding error rate.

This RTRA has been tested extensively with ns-2 under different channel conditions and traffic charateristics. Here, we only present one typical simulation result to show the effectiveness of the proposed RTRA scheme.

In the presented case, we assume a CBR source with an output rate of 3.01 Mbps, a packet size of 1000 bytes, and a time-varying channel with

$$\mathbf{A}_e = [0\ 0.5\ 0.5], [0.5\ 0\ 0.5], [0.5\ 0.5\ 0]$$

and

$$\mathbf{P}_e = [0.35\ 0.4\ 0.45]$$

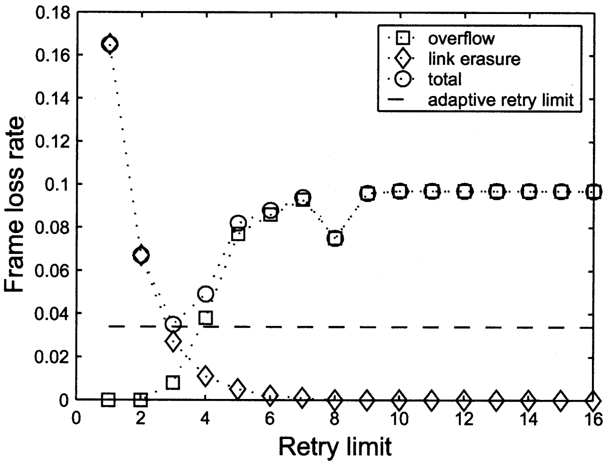


Fig. 6. Packet loss rates under fixed retry limits or the RTRA.

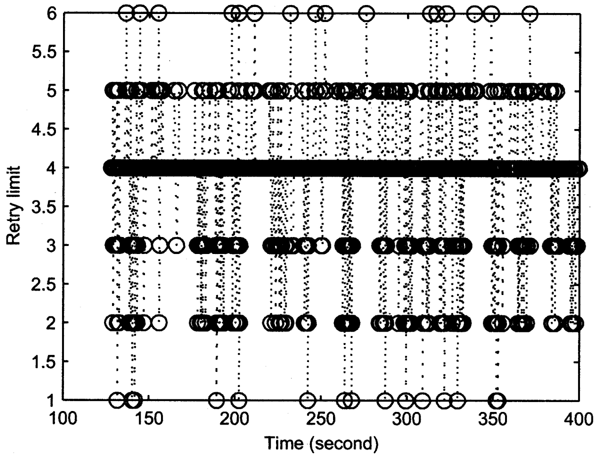


Fig. 7. Trace of retry limit adaptation.

and $T_e = 5$ s. The simulations are then performed either with fixed retry limits (in the range of [1, 16]) or with the RTRA deployed. Fig. 6 depicts the resulting packet loss rates. From this figure, the following observations can be made.

- There exists the best static setting that can minimize the number of packet losses given this channel model. In this particular case, the best static setting is 3.
- When the RTRA is deployed, the simulation achieved a loss rate that is slightly less than that achieved by the best static setting, as shown by the dash line in Fig. 6.

However, as mentioned previously, the best static setting changes with channel conditions and traffic characteristics, and thus the MAC cannot optimize the retry limit *a priori*. Alternatively, the proposed RTRA scheme is able to quickly track the optimal retry limits corresponding to the different states of the selected time-varying channel error model. Fig. 7 shows the tracked retry limit by the RTRA. The average retry limitcalculated from the trace is 3.75.

Additional simulations have been conducted to test the responsiveness of the RTRA to the changes of channel conditions. We find that the proposed RTRA is able to track the optimal retry limit even when the channel state duration T_e is reduced to 0.5 s.

In summary, extensive simulations show that the retry limit adaptation is a good mechanism for minimizing the packet loss

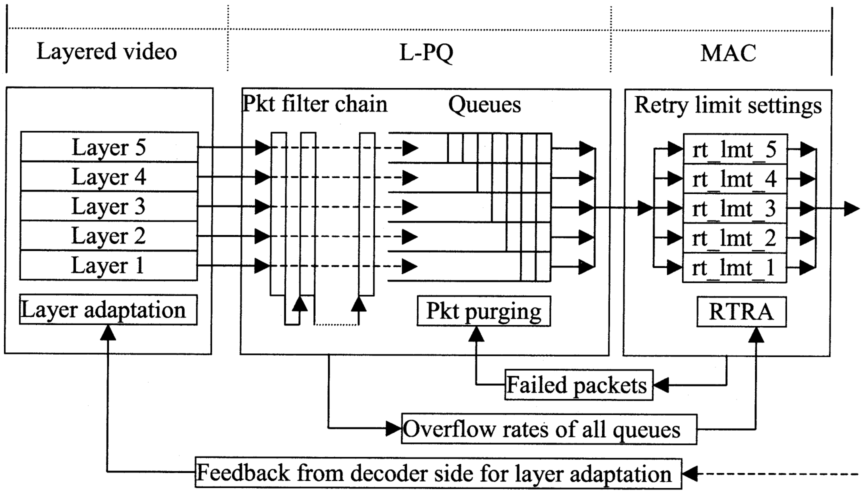


Fig. 8. Combination of the RTRA and the L-PQ.

occurences experienced over IEEE 802.11 WLANs. The proposed RTRA algorithm can accurately track the optimal retry limit setting under a variety of channel conditions, thereby improving the performance of the transmitted video. However, the RTRA algorithm alone cannot provide priority service and optimized unequal protection to video packets, and hence, in order to improve the performance of the transmitted video over WLANs, the RTRA has to be combined with priority queueing, as discussed in Section III.

III. Priority Queueing

Due to the existence of dependencies among video layers and frames, the loss of each video packet has a different impact on the received video quality [18], [19], and thus should be differentially protected. In this section, we apply the priority queueing discipline[6] at the network buffer to the different video layers in order to provide prioritized delivery. Meanwhile, we propose using different retry limits by the MAC to offer unequal protection to video layers.

A. Priority Queueing

In computer networks, when packets arrive at a node where they cannot be immediately forwarded to the next downlink node (receiving service), they have to be stored in the network buffer. The way a node manages these delayed packets in its buffer, such as the order of receiving service or dropping, is called a queueing mechanism (or service discipline). Priority queueing (PQ) is referred to the type of service discipline in which multiple queues are maintained and associated with different priority levels. The packets in these queues will receive differentiated services that are determined by the priority levels. In the so-called absolute PQ discipline, queues with lower priorities can receive service only when all queues with higher priorities are empty. We apply this type of queueing discipline to the different video layers at the network buffer.

Fig. 8 shows the architectural diagram of the design of our proposed PQ for layered video. In the design, the interface (or the network layer in the operating system (OS) kernel of the host machine) maintains multiple queues for each of the video layers. The priority of each queue is correlated with the importance of the associated video layer. All the queues are managed by a common absolute PQ discipline. For the convenience of description, we denominate our design Layered PQ (L-PQ).

In our design, we added two new functions to the conventional PQ: packet filtering and purging. Packet filtering is aimed at discarding the packets that arrive at the queueing system but fail to pass the chain of packet filters that are installed at the entry of the L-PQ . These filters specify the dependency relationship among the various IP video packets (we assume that this dependency information can be easily extracted from the RTP headers). Packet purging is an action that will be taken by the L-PQ whenever a packet transmission fails due to either overflow or link erasure (such as after the retries). Two procedures will be performed in the purging action: a) all packets depending on the lost packet will be purged from the queues, and b) a specially designed filter will be inserted into the chain that will be used by the filtering function to block further dependent packets to enter the queues. The purpose of having these two functions is to save the wireless link capacity and the buffer that would be otherwise wasted with the delivery of unusable packets. Since these functions are performed by the network layer with the assistance of link layer feedback, they are only executed whenever it is necessary. Therefore, they can achieve higher efficiency in terms of bandwidth saving and link utilization, as well as lower delays, when compared with packet scheduling strategies deployed at application layer [18], [19].

The filter chain maintained for the L-PQ to perform packet filtering is managed by a dedicated function. Whenever an overflow or link erasure event happens, a new filter is constructed and inserted into the chain by the management function, while outdated filters are deleted in order to control the growth of the chain.

When the queueing system performs purging over the queues, it also discards packets that are too late to be sent by comparing the RTP timestamps of the buffered packets in the purged queues with that of the failed one that triggered this purging action.

Since the functionality of the filtering and purging are straightforward, we omit here the details of the employed algorithms and filter structure. The bandwidth saving benefit of these functions are studied and the results of the simulations are presented in Section IV.

B. Unequal Protection

In the L-PQ, queues of various priority levels have different perceptions on the available link capacity. Assuming R_i, $i = 1, 2, \ldots N$ are the incoming rates to the queues and C is the total available link capacity, the perceived link capacity of queue j in the worst case[7] can be approximately expressed as

$$C_j = \max\left\{0, C - \sum_{i=1}^{j} R_i\right\}. \tag{8}$$

As long as $R_j < C_j$, queue j will have few overflow losses. However, all queues will still be exposed to the same packet erasure rate. If the MAC layer can differentiate these queues by associating them with different retry limits that are positively correlated with their priorities (see Fig. 8), then the MAC automatically provides unequal protection to packets from different queues. In the following, we develop a principle for the configuration of retry limits for the multiple queues managed by the same L-PQ.

In Section II, it was determined that an optimal retry-limit setting exists that minimizes the overall packet loss. In the fluid model discussed in Section II-A1, we assume that when the arrival rate λ multiplied with the average number of transmissions $s(L_r, P_e)$ for a single packet exceeds the link capacity C, overflow will occur with the rate as defined by (2). Subsequently, we apply the same assumption to the multiqueue system to develop a systematic retry-limit configuration method for the MAC to perform optimized unequal protection.

Similar to the single-queue case, we can show that there exists a set of retry-limit settings that can minimize the overall loss of the multiqueue system under the assumption of the fluid model. Let vector $\mathbf{L_r} = [L_{r1}\ L_{r2}\ \cdots\ L_{rN}]$ be the set of retry limits maintained by the MAC for the queues in the priority queueing system, vector $\mathbf{s} = [s_1(L_{r1}, P_e)\ s_2(L_{r2}, P_e)\ \cdots\ s_N(L_{rN}, P_e)]$ represent the set of average link retransmissions that the packets from different queues will experience under given $\mathbf{L_r}$ and P_e, λ indicate the total arrival rate to the system and vector $\mathbf{\Lambda} = [\Lambda_1\ \Lambda_2\ \cdots\ \Lambda_N]$ represent the departure rates of the queues to the link determined by the queueing discipline. The system-wide average packet retransmissions (SAPR) can be calculated as

$$\overline{s}(\mathbf{L_r}, P_e) = \frac{\mathbf{\Lambda} \cdot \mathbf{s}(\mathbf{L_r}, P_e)}{\mathbf{\Lambda} \cdot \mathbf{1}} \tag{9}$$

where $\mathbf{1} = [1\ 1\ \cdots\ 1]$. Similar to (2), we define the overflow rate of the multiqueue system as

$$p_B(\mathbf{L_r}, P_e) = \frac{\lambda \overline{s}(\mathbf{L_r}, P_e) - C}{\lambda \overline{s}(\mathbf{L_r}, P_e)} \tag{10}$$

where C is the link capacity as before. The link erasure rate can be calculated as

$$p_L(\mathbf{L_r}, P_e) = \frac{\mathbf{\Lambda} \cdot P_e^{\mathbf{L_r}+\mathbf{1}}}{\mathbf{\Lambda} \cdot \mathbf{1}}. \tag{11}$$

It can be easily shown with similar analysis as in Section II-A1 that, when

$$p_B = p_L = 1 - \frac{1}{\rho(P_e)} \tag{12}$$

where $\rho(P_e) = \lambda / C(1 - P_e)$, the overall loss rate $p_T = p_B + p_L$ will be minimum. Equation (12) does not change with $\mathbf{L_r}$, implying that, given $\boldsymbol{\lambda}$ and P_e, there could exist an infinite number of optimal $\mathbf{L_r}$ that can balance the overflow and the erasure such that the system loss rate is minimized.

Our goal is to develop a simple mechanism that can be used to find an optimal retry limit vector that is appropriate for video streaming applications. With the fluid model assumption, we have the following lemmas and propositions:

Lemma 1: Assume $\mathbf{L_{r1}}$ and $\mathbf{L_{r2}}$ are two different retry limit vectors, if they invoke the same SAPR, i.e., $\overline{s}(\mathbf{L_{r1}}, P_e) = \overline{s}(\mathbf{L_{r2}}, P_e)$, then they will generate the same overflow rate, i.e., $p_B(\mathbf{L_{r1}}, P_e) = p_B(\mathbf{L_{r2}}, P_e)$.

Proof: It is true from (10).

Lemma 2: Assume $\mathbf{L_{r1}}$ and $\mathbf{L_{r2}}$ are two different retry limit vectors, if they invoke the same SAPR, i.e., $\overline{s}(\mathbf{L_{r1}}, P_e) = \overline{s}(\mathbf{L_{r2}}, P_e)$, then they will also generate the same link erasure rate, i.e., $p_L(\mathbf{L_{r1}}, P_e) = p_L(\mathbf{L_{r2}}, P_e)$.

Proof: Similar to the calculation of (1), we have

$$\mathbf{s}(\mathbf{L_r}, P_e) = \frac{1 - P_e^{\mathbf{L_r}+1}}{1 - P_e}.$$

From (9), and considering $\overline{s}(\mathbf{L_{r1}}, P_e) = \overline{s}(\mathbf{L_{r2}}, P_e)$, we immediately have

$$\frac{\mathbf{\Lambda}(\mathbf{L_{r1}}) \cdot P_e^{\mathbf{L_{r1}}} + \mathbf{1}}{\mathbf{\Lambda}(\mathbf{L_{r1}}) \cdot \mathbf{1}} = \frac{\mathbf{\Lambda}(\mathbf{L_{r2}}) \cdot P_e^{L_{r2}} + 1}{\mathbf{\Lambda}(\mathbf{L_{r2}}) \cdot \mathbf{1}}$$

which leads to $p_L(\mathbf{L_{r1}}, P_e) = p_L(\mathbf{L_{r2}}, P_e)$.

Proposition 1: Two different retry limit vectors could achieve the same system performance in terms of queue overflow rate and link erasure rate, as long as they invoke the same SAPR.

Proof: It is true due to Lemmas 1 and 2.

Proposition 2: If $\mathbf{L_{r1}}$ is an optimal retry limit vector to the system, and retry limit $\mathbf{L_{r2}}$ will invoke the same SAPR as that of $\mathbf{L_{r1}}$, then $\mathbf{L_{r2}}$ is also optimal to this system.

Proof: It is true because of Proposition 1 .

Proposition 2 provides a method on how to construct a new optimal retry limit vector based on a known solution.

To facilitate the discussion, we introduce the concept of a *shadow retry limit* (SRL) for the MAC, and along with it, a special retry limit vector $\mathbf{L_{srl}}$ with all its elements equal to the SRL. This $\mathbf{L_{srl}}$ will be maintained by the MAC but not enforced on any of the queues. Meanwhile, by assuming the same SAPR as that of $\mathbf{L_{srl}}$, the MAC computes another retry limit vector $\mathbf{L_{re}}$ (with unequal elements) that will actually be applied to the queues. Based on Proposition 1 , these two retry limit vectors will achieve the same performance in terms of queue overflow

[7] The worst case means that every arrival of queue j packets has to wait for the finish of the service of a packet from a queue with lower priority. We assumed that the packets of all queues have similar sizes on average.

and link erasure. If $\mathbf{L_{srl}}$ is optimal to the system, the correspondingly mapped vector will also be optimal according to Proposition 2.

The optimal $\mathbf{L_{srl}}$ of a multiqueue system can be calculated as follows: 1) treat the multiple queues as a single-queue system, such that by lumping all the overflows and link erasures observed from different queues, a single overflow rate and link erasure rate of the system can be calculated; 2) run the RTRA algorithm developed for a single-queue system (see Section II-B for detail) to track the time-varying optimal SRL, and the corresponding optimal $\mathbf{L_{srl}}$.

Subsequently, we discuss how to map the tracked optimal $\mathbf{L_{srl}}$ to a retry limit vector that is appropriate for video streaming applications. Assume vector $\mathbf{P_v} = [P_{v1}\, P_{v2} \cdots P_{vN}]$ specifies the tolerable packet loss rates of all the video layers, which is determined by the QoS requirement of the layered video; $\mathbf{L_{re}}$ is the vector that will be enforced by the MAC. The mapping can be performed with the following algorithm:

1) Calculate vector $\mathbf{L_{rv}} = [\lceil \log_{P_e} P_{v1} - 1 \rceil\ \lceil \log_{P_e} P_{v2} - 1 \rceil \cdots \lceil \log_{P_e} P_{vN} - 1 \rceil]$, which specifies the minimum retry limit for the queues that can satisfy $\mathbf{P_v}$[8].
2) Initiate index $i = 1$.
3) Construct $\mathbf{L_{re}} = [L_{rv1} \cdots L_{rvi}\, 0 \cdots 0]$.
4) If $\overline{s}(\mathbf{L_{re}}, P_e) < \overline{s}(\mathbf{L_{srl}}, P_e)$, go to step 5. Otherwise, continuously reduce the ith components L_{rei} by 1 until $\overline{s}(\mathbf{L}_{re}, P_e) \le \overline{s}(\mathbf{L_{srl}}, P_e)$ again, and stop.
5) if $i = N$, stop; otherwise, $i = i + 1$ and go to step 3.

Remarks:

- $\overline{s}(\mathbf{L_{re}}, P_e)$ is calculated with (9), where $\mathbf{\Lambda}$ and P_e are estimated by the MAC in real-time.
- The mapping algorithm allocates the retry limits to the queues under the SAPR constraint predicted with the SRL assumption. The SRL is tracked by the RTRA algorithm.
- The protection requirements of higher priority video layers are always satisfied first by the mapping algorithm. If the algorithm stops at step 5, it means the whole video protection requirement is fully satisfied.

Clearly, with this mapping algorithm, the system can provide adaptive unequal protection to different video layers.

The queues (normally implemented at the network layer) need to pass down queueing status information to the MAC, such that it can apply the RTRA algorithm appropriately. Meanwhile, the MAC also needs to pass to the upper layer the packet identifications that fail their transmissions after their retries for the queues to perform the purging function. Therefore, this design is a cross-layer solution.

The implementation of L-PQ and RTRA will add computing overhead to the network and link layer operations. However, the incurred complexity is limited, since some form of priority queueing and overflow rate calculation are already available in certain OS kernels (such as Linux, in order to support Differentiated Services and Random Early Discard), and WLAN card drivers generally count and store successful and failed transmissions in their management information bases (MIBs). The only additional steps needed for the implementation of the proposed schemes are the cross-layer exchanges of packet loss information and the run-time reconfiguration of the retry-limit settings.

IV. Results

In this section, the performances of our design is examined by ns-2 simulations, in which a FGS coded video trace is attached to the sender of the simulated two-node WLAN (see Section II-A3 for detail). With this setup, we investigate the bandwidth saving benefits of L-PQs filtering and purging functions, the unequal protection achieved by applying the RTRA to multiqueue system, and the improvement on the transmitted video quality.

A. Traffic Source

In order to create a realistic testing environment, we designed an FGS server and client and connected them to the ns-2 simulator. The server module divides the FGS video into multiple layers that are sent in independent RTP/UDP connections. The server starts by transmitting the base layer, and then, progressively, adds the enhancement layers to the session within the rate-ramp-up period (about 1 s). After that the video layer number will not change. To cope with channel variations, the rate-adaptation is performed automatically by the L-PQ in the network layer.

Besides the video bitstream, an FGS video trace contains the sizes of all base and enhancement layer frames. Using this side information, the server generates RTP packets and sends them down to the simulated WLAN timed by the RTP timestamps. At the client side, a packet trace is also collected, which records all received packets. The information contained in the collected trace can then be used to calculate the distortion of the associated video.

For the experiments we coded a standard definition (SD) video clip at 25 Hz using FGS with a base layer of 1 Mbps and a fine-granular enhancement-layer of 2.4 Mbps. When the video trace is loaded by the server, the server divides the enhancement layer into six smaller layers, each adding approximately 0.5 dB gain in terms of the overall peak-signal-noise-ratio (PSNR). Then, seven RTP/UDP channels are set up to send the base layer and the six enhancement layers.

B. Bandwidth Saving Benefits

This simulation is designed to demonstrate the benefit in bandwidth saving achieved by the packet filtering and purging functions of L-PQ. In the simulation, the MAC uses the same fixed retry limit $(= 4)$ to protect all L-PQ queues.

The percentage of packets that are blocked or purged by L-PQ when the channel PER (P_e) changes in the range of $[0.1, 0.8]$ is shown in Fig. 9. When the channel is bad (such as $P_e > 0.5$), more than 30% of packets out of the total packets sent by the server are either blocked or purged. Note that the exact number of packets that are effected by a single loss (due to either queue overflow or link erasure) depends on the encoding parameters (GOP size, number of B-frames, layering schemes, concealment strategy etc.). In our simulations, we measured that a single loss can cause two to 11 packets to be selectively blocked or purged.

[8] $\mathbf{L_{rv}}$ is derived from $P_e^{\mathbf{L_{rv}}+1} \le \mathbf{P_v}$.

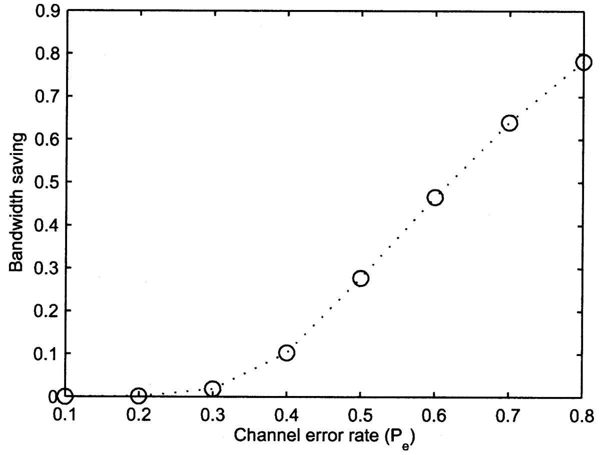


Fig. 9. Percentage of bandwidth saving versus channel error rate (P_e).

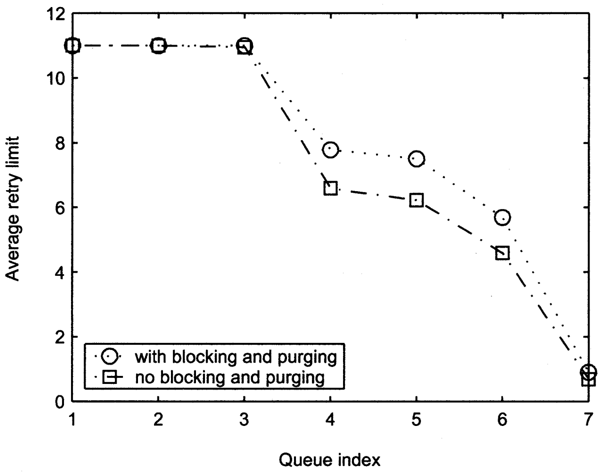


Fig. 10. Average retry limit of each queue.

C. Retry-Limit Adaptation and Unequal Protection

When the RTRA is combined with the L-PQ, the MAC runs the adaptation algorithm in the background to calculate the SRL, and then maps it to a retry limit vector to achieve unequal protection. The garbage collection operation performed by the packet blocking and purging functions of the L-PQ can reduce the overflow probabilities of these queues. As a result, the MAC can calculate a slight higher SRL with the RTRA, and consequently the queues with lower priorities can be assigned with slightly higher retry limits on average.

We simulated two cases, one where the packet blocking and purging functions of the L-PQ are enabled and one where they were not. In both scenarios, the same video trace and number of layers are used as in the previous simulation. The physical link is simulated as a three-state channel as described in Section II-B. We initialize the vector $\mathbf{L_{rv}} = [11\ 11\ \cdots\ 11]$ for mapping the SRL to the vector $\mathbf{L_{re}}$ (see Section III-B for the description of the mapping algorithm). From the simulations, the following observations can be made.

- In both cases, the QoS-specified minimal retry limit of queue 1 to 3 can always be satisfied $(= 11)$, but queue 4 to 7 have decreasing average retry limits that are less than their minimal QoS requirements, as shown in Fig. 10. The calculated SRL trace when packet blocking and purging is enabled is shown in Fig. 11. This SRL is adapted by the RTRA alogrithm under the objective to minimize the overall system-wide packet losses. Clearly, the system can achieve adaptive and unequal protection.
- As expected, the average retry limits of queues 4 to 7 in the case when the packet blocking and purging function is enabled are slightly larger than in the other case.

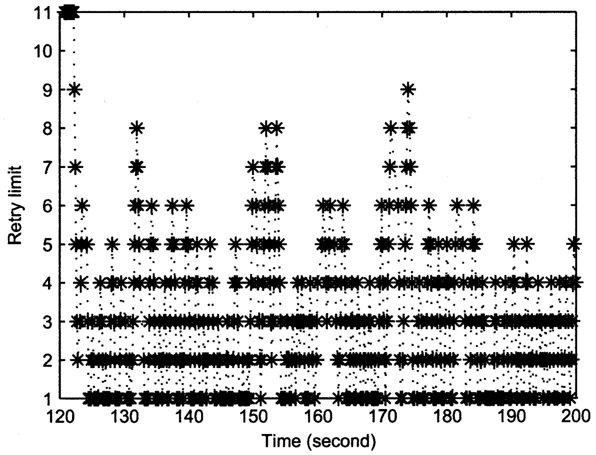
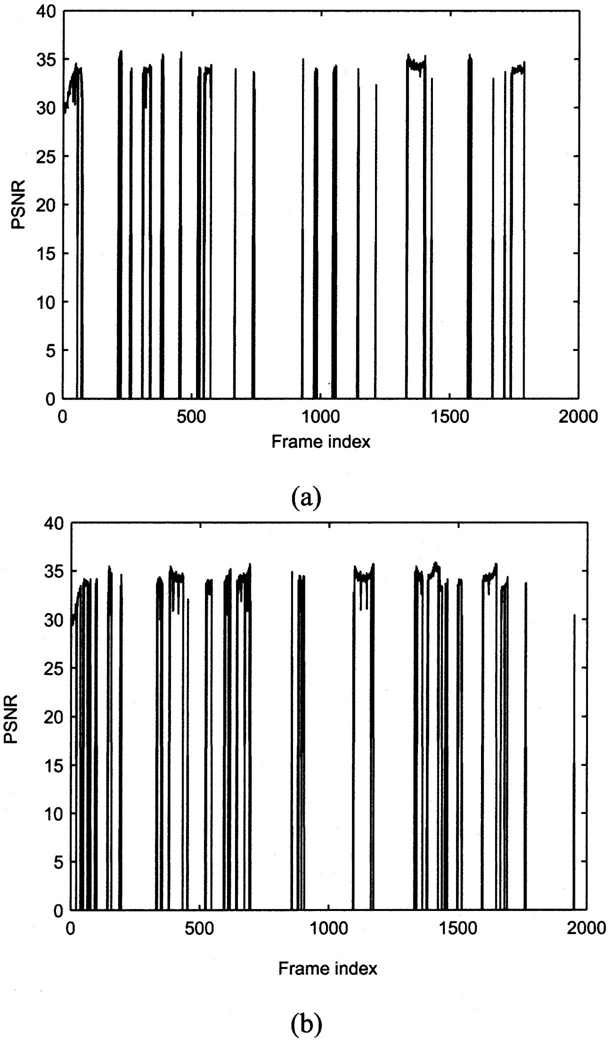


Fig. 11. Calculated SRL by the RTRA.

D. Performance Evaluation

We design two testing scenarios to evaluate the performance of the RTRA and the L-PQ in terms of their impact on the video quality measured by PSNR. In the first scenario, a FIFO (first-in–first-out) queue is used with or without the RTRA being deployed. In the second scenario, the L-PQ is used with or without the RTRA being deployed. In the simulations below, the same video trace and number of layers are used as in the previous experiments. We assume the same three-state channel for the link as before.

Scenario I: FIFO In this scenario, a single FIFO queue is used for the interface. The buffer size is set to 50 packets. We run the simulations with either the RTRA being deployed or a fixed retry limit for the MAC. For fair comparison, the fix retry limit $(= 5)$ is chosen as the best in terms of the overall losses it will produce. The PSNRs of received video frames at the client side are calculated based on the collected packet traces, as shown in Fig. 12. A simple error concealment mechanism is performed by the decoder: if a packet of a base-layer frame is lost, then the entire frame containing the lost packet is set equal to the last correctly received frame. Tables I and II shows the average PSNRs of the received frames and the "freeze frequency" defined as the ratio of the failed frames and the total sent frames. We observe that with the RTRA, the system experiences a lower "freeze frequency" and a better PSNRs for the received frames, thereby leading to better video quality than that obtained when a fixed retry limit is used. However, both these cases lead to intolerably high "freeze frequencies" that considerably affect the viewing experience. This measurement can be dramatically improved by using the L-PQ mechanism.



(a)

(b)

Fig. 12. PSNR of received frames under FIFO, and (a) retry limit is fixed at 5; (b) RTRA is deployed.

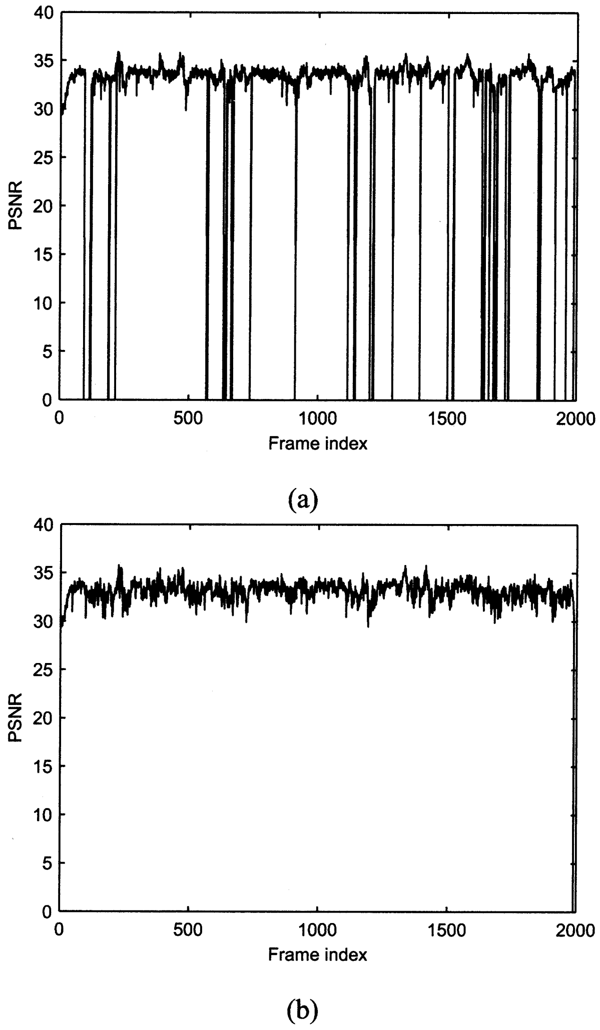
TABLE I
AVERAGED PSNR OF RECEIVED FRAMES AND FREEZE FREQUENCY

Retry limit	Freeze freq.	PSNR (dB)
Fixed at 5	0.84	33.6
RTRA adapted	0.74	33.9

TABLE II
AVERAGED PSNR OF RECEIVED FRAMES AND FREEZE FREQUENCY

Retry limit	Freeze freq.	PSNR (dB)
Fixed at 6	0.0729	33.48
Adapted and unequal	0	33.09

Scenario II: L-PQ In this scenario, the interface maintains seven queues buffering packets from different separate video layers. The base-layer has a buffer size of 15 packets, while the enhancement layer queues have a buffer size of five packets. Hence, the total interface buffer size is 45 packets. These queues are under the management of the L-PQ. Two runs of simulations are performed correspondingly with either equal or unequal protection. In the unequal-protection case, the unequal-protection vector is mapped from the SRL which is in turn adapted by the RTRA. In the equal-protection case, all queues are protected by the same fixed retry limit $(= 6)$, which will generate roughly the same SAPR as the unequal-protection case. The PSNRs of the received frames are shown in Fig. 13, and the average PSNRs and freeze frequencies are shown in Table II. We observe that when adaptive unequal protection is engaged, the system can achieve a 0 freeze frequency, but slightly lower PSNRs of received frames when compared with the equal protection case. Comparing with the simulations in Scenario I, it seems that L-PQ can dramatically reduce the freeze frequency of the received video, thereby leading to a considerably improved system.



(a)

(b)

Fig. 13. PSNR of received frame under L-PQ, and (a) retry limit is fixed at 6; (b) retry limit is adaptive and unequal for different queues.

To determine the differences in subjective video quality between the results presented in Tables I and II, we set up a visual experiment according to the CCIR Recommendation 500-4 [21]. Forty seven viewers participated in the assessment. Among them, thirty participants were not video experts. Before each evaluation, the observers were given a short introduction to the type of video artifacts to expect (i.e., frame freezes, motion-jitter, variations in video quality, blockiness etc.) and

TABLE III
SUBJECTIVE VIDEO QUALITY EXPERIMENT

Retry limit	Visual Score
Fixed at 5	1.4
RTRA adapted	1.9
Fixed at 6 (L-PQ)	3.8
Adapted and unequal (L-PQ)	4.6

how the test scale panel was designed. For each session, five to seven viewers were simultaneously seated in front of the display, at about 4H(H = display height) to 6H viewing distance. A $42''$ 16:9 Plasma monitor was used for the assesment. The ambient illumination was low. The different videos (having different retry limit strategies) were shown sequentially (each particular video was shown sequentially three times) and the observers were asked to evaluate their visual quality using a continuous scale on the test scale panel. Since the experiments are conducted at relatively low bit rates and under adverse channel conditions, and thus, impairments are expected, the selected five scales and their corresponding scores are very annoying (1), annoying (2), slightly annoying (3), perceptible but not annoying (4), and imperceptible (5).

To illustrate the evaluation results, the statistical scores are summarized in Table III. These results clearly show the benefit of using L-PQ for providing an improved subjective quality for wireless video transmission. Also, employing adaptive unequal error protection does further improve the performance of the proposed system.

V. CONCLUSION

In this paper, a cross-layer protection method that can provide adaptive quality of service to layered video when transmitted over wireless networks is proposed. With this method, video layers of different importance will automatically receive appropriate priority delivery and unequal protection depending on channel conditions. The priority delivery is performed by the proposed priority queuing discipline, while unequal protection is achieved through a retry limit vector with unequal elements that are maintained by the MAC for different video layers. The design of this method is based on a key observation, that is, for a given traffic characteristics and a channel condition there exists an optimal retry limit setting for the wireless link under which the total losses due to both link erasure and buffer overflow will be minimum. It is observed that when traffic characteristics or channel conditions change, the optimal setting also changes, but always stay at a value that can balance the link erasure rate and the overflow rate. This observation leads to the design of the real-time retry-limit adaptation (RTRA) algorithm and the systematic method of retry-limit vector configuration for unequal protection. By adopting video-layer dependent packet blocking and purging operations, the designed queueing discipline can significantly save bandwidth that could otherwise be wasted by delivering obsolete or un-decodable packets. Simulations show that the proposed L-PQ method alone can significantly reduce the freeze frequency of streamed video under testing environments. When aided with RTRA, the method can essentially eliminate freeze frequency, achieving zero frame loss, therefore, dramatically improve video quality over wireless local area networks.

ACKNOWLEDGMENT

The authors would like to thank the three anonymous reviewers who provided excellent and thorough feedback that improved the quality of this paper.

REFERENCES

[1] (1996) DARPA Funded VINT Project UCB/LBNL/VINT Network Simulator—ns (Version 2). [Online]. Available: http://www-mash.cs.berkeley.edu/ns

[2] J. Gross and A. Willig, "Measurements of a wireless link in different RF -isolated environments," *Proc. Eur. Wireless 2002*, pp. 762–769, Feb. 2002.

[3] R. W. Brockett, W. Gong, and Y. Guo, "Stochastic analysis for fluid queueing systems," in *Proc. 38th IEEE Conf. Decision and Control*, vol. 3, Dec. 1999, pp. 3077–3082.

[4] A. Yan and W. Gong, "Time-driven fluid simulation for high-speed networks," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1588–1599, July 1999.

[5] W. Gong, D. Towsley, Y. Wu, and D. Ye, "A stochastic differential equation model for queues with traffic smoothing," in *Proc. 42nd Annu. Allerton Conf. Communication, Control and Computing*, 2002.

[6] L. Kleinrock, *Queueing Systems. Volume 1: Theory*. New York: Wiley-Interscience, 1995.

[7] ——, *Queueing Systems. Volume 2: Computer Applications*. New York: Wiley-Interscience, 1996.

[8] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 524–534, June 2002.

[9] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Ref. ISO/IEC 8802-11:1999(E), IEEE Std. 802.11-199, 1999.

[10] S. Li and C.-L. Hwang, "Queue response to input correlation functions: Discrete spectral analysis," *IEEE/ACM Trans. Networking*, vol. 1, pp. 522–533, Oct. 1993.

[11] H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, pp. 53–68, Mar. 2001.

[12] B. Girod and N. Farber, *Wireless Video, a Book Chapter in Compressed Video Over Networks*. New York: Marcel Dekker, 2001.

[13] Y. Shan and A. Zakhor, "Cross layer techniques for adaptive video streaming over wireless networks," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, Aug. 2002, pp. 277–280.

[14] G. Cote, F. Kossentini, and S. Wenger, "Error Resilience Coding," in *Compressed Video Over Networks*. New York: Marcel Dekker, 2001.

[15] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communications: A review," *Proc. IEEE*, vol. 86, pp. 974–997, Oct. 1998.

[16] H. Ma and M. E. Zarki, "MPEG-2 video transmission over wireless access networks using type-I hybrid ARQ schemes for VoD services," in *Proc. Packet Video Workshop99*, Apr. 1999.

[17] M. van der Schaar and H. Radha, "Unequal packet loss resilience for fine-granular-scalability video," *IEEE Trans. Multimedia*, vol. 3, pp. 381–394, Dec. 2001.

[18] P. A. Chou and Z. Miao, "Rate-distortion optimized sender-driven streaming over best-effort networks," in *IEEE Workshop on Multimedia Signal Processing*, Cannes, France, Oct. 2001, pp. 587–592.

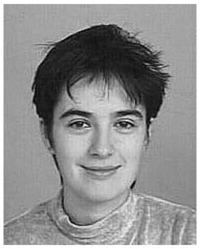
[19] Z. Miao and A. Ortega, "Expected run-time distortion based scheduling for delivery of scalable media," in *12th Int. Packet Video Workshop (PVW 2002)*, Pittsburgh, PA, Apr. 2002.

[20] J. Postel. User Datagram Protocol, RFC768. [Online]. Available: ftp://ftp.rfc-editor.org/in-notes/rfc768.txt

[21] "Method for the Subjective Assessment of the Quality of Television Pictures,", CCIR Rec. 500-4, 1990.

Qiong Li (S'98–M'00) received the B.E. and M.E. degrees from Hefei University of Technology, Hefei, China, in 1986 and 1989 respectively, the Ph.D. degree in electric machines from Tsinghua University, Beijing, in 1995 and the Ph.D. degree in electrical and computer engineering from the University of Delaware, Newark, in 2000.

He is currently Principal Network Architect at BrainMedia, New York. From 2000 to 2003, he was a Senior Member of Research Staff at Philips Research USA, Briarcliff Manor, NY, where he was involved in the development of Internet video streaming technology. His current research interests include computer networks, wireless communications and multimedia communications with emphasis on active queue management, cross-layer optimization, protocol design, and streaming server architecture.



Mihaela van der Schaar (M'98–SM'04) received the M.Sc. and Ph.D. degrees in electrical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands.

In 1996, she joined Philips Research Laboratories, Eindhoven, The Netherlands, where she was a Research Scientist in the TV Systems Department. In 1998, she became a Senior Member of Research Staff in the Wireless Communications and Networking Department, Philips Research, Briarcliff Manor, NY, where she led a team of researchers working on scalable video coding, networking, and streaming algorithms. She is currently an Assistant Professor in the Department of Electrical and Computer Engineering, University of California, Davis, since July 2003. From January to September 2003, she was also an Adjunct Assistant Professor at Columbia University, New York. Since 1999, she is an active participant to the MPEG-4 standard, contributing to the scalable video coding activities. She was coeditor of the MPEG-4 *Fine Granularity Scalability* standard. She is currently Chairing the MPEG Ad-Hoc Group on scalable video coding and is also Co-chair of the Ad-Hoc Group on multimedia test bed. Her research interests include multimedia coding, processing, networking, and architectures. She has coauthored more than 80 conference papers, journal papers, and book chapters in this field and holds ten U.S. patents. She is an Associate Editor for SPIE *Journal of Optical Engineering*, as well as a guest editor for EURASIP *Journal on Applied Signal Processing*, special issue on multimedia over IP and wireless networks.

Dr. van der Schaar is a Member of the Technical Committee on multimedia signal processing of the IEEE Signal Processing Society and is an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA.