Error Protection of Video over Wireless Local Area Networks Through Real-time Retry Limit Adaptation

Qiong Li and Mihaela van der Schaar
Philips Research USA

***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, a novel error protection method for wireless video utilizing a retry-limit adaptation algorithm at the link layer is proposed. The design of this method is motivated by the observation that the retry limit settings of the MAC layer can be optimized to minimize the overall packet losses that are caused by either link erasure or buffer overflow. To develop this method, we conducted theoretical anaylyses on the relation between MAC retry-limit settings and packet error rates, and developed a real-time retry-limit adaptation algorithm to trace the optimal retry limit for the single-queue case under varying wireless channel conditions and traffic loads. Simulations show that the proposed cross-layer protection method can significantly reduce overall packet losses and improve network throughput, resulting better video quality.**

I. Introduction

Real-time streaming of video over 802.11 Wireless LANs (WLANs) poses many challenges, one of them would be in coping with packets losses caused by network buffer overflow or link erasures. The existing solutions for combating wireless transmission errors as reviewed in [5] may include adapting the physical layer modulation schemes, combining Forward Error Correction (FEC) with interleaving, and employing closed-loop error control strategies, i.e., Automatic Repeat reQuest (ARQ). These techniques focus mainly on the robust transmission of video over cellular networks. In [4], Majumdar et al. address the problem of resilient real-time video streaming over IEEE 802.11b WLANs with the proposal of a hybrid ARQ algorithm that efficiently combines FEC and ARQ. Similar schemes have also been presented by Wang and Zhu in [6] and by Ma and Zarki in [7]. However, 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 this paper, we introduce an adaptive cross-layer protection method for IEEE 802.11 type of WLANs. 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 — cross-layer considerations. In this paper, we present a heuristic algorithm that determine and adapt in real-time the optimal retry limit to minimize packet loss, referred to as the real-time retry-limit adaptation (RTRA) scheme.

Experiments show that the proposed cross-layer protection method, due to its adaptive nature, can provide improved network error rate, throughput, and subsequently better video quality over WLANs.

The rest of this paper is organized as follows. Section II presents a theoretical analysis on how link retry-limit setting may effect buffer overflow and link drops. Section III develops and evaluates a heuristic algorithm for single-queue retry-limit adaptation that can minimize overall packet losses. We conclude the paper in Section IV.

II. Impact of Retry-limit on Packet Losses

According to the IEEE 802.11 MAC standard, 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. In the current standard, this retry limit is normally configured statically.

In the following, we analyze how MAC retry limit may effect buffer overflow using queueing models and ns-2 [1] simulations.

A. Buffer Overflow versus Link Erasure

When the network buffer is full, arriving packets will simply be dropped, generating an overflow event. However, buffer overflow is not independent from link erasure. 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 generally modeled using a queueing system with bursty arrivals and a time-varying service rate. However 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: In this analysis, a simple instance of fluid model as introduced in [3] is adopted with the assumption of static channel conditions and constant arrival rate. 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:

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

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

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

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

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

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 [4], 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:

$$p_L = P_e^{L_r+1}. \quad (4)$$

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 [2], becomes:

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

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:

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

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

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

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 the following section.

2) M/G/1 model: We apply the /M/G/1 queue model [2] 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:

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

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

[2] We assume both P_B and P_L are relatively small such that they can be added together to approximate the total loss rate.

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:

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

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.

This M/G/1 queueing system can be modeled by an embedded Markov chain [2]. 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 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 3.52Mps. The calculated overflow rate is shown in Figure 1 together with the calculated link erasure rate and total loss rate based on equations (4) and (5). We observe once again that, the conjecture as formulated in equation (6) is still valid with the M/G/1 queue. Furthermore, we show in the next section that they are also consistent with the simulation results obtained using the ns-2 network simulator.

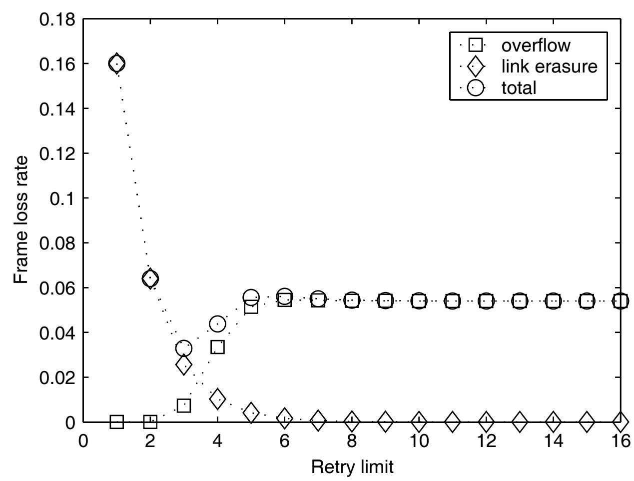

Fig. 1. Packet loss rate vs. MAC retry limit obtained through M/G/1 queue analysis

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 $P_e = 0.4$. The other MAC parameters, except the retry limit, are set as recommended by the IEEE 802.11b standard. 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 400s. The measured overflow rate, link erasure rate and the sum of these two are shown in Figure 2. 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 analyses is maintained.
- 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.

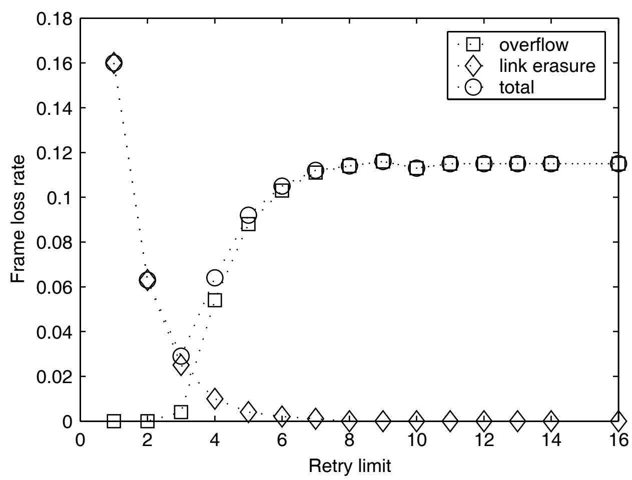

Fig. 2. Simulated packet loss rate vs. MAC retry limit under CBR traffic.

Additional simulations reveal that the optimal retry limit is not a fixed value, but rather is dynamically determined by the packets arrival rate, higher order traffic statistics and the channel condition since the queueing process is effected by these factors.

III. Rea-Time Retry-Limit Adaptation

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 Figure 2). 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.

A fine-tuned algorithm that has been tested by extensive simulations is presented in Figure 3.

The performance of this algorithm is controlled by four parameters: P_{Th_1}, R, P_{Th_2} and θ (see Figure 3 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 [3]. Each state

```
while (1) {
    periodically calculate and compare P_B and P_L
    //if the sum of P_B and P_L is small (< P_Th1),
    //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_Th1 ) {
        if ( L_r > R) {
            L_r decreases by 1;
            continue;
        }
    }
    //if the difference between P_B and P_L is small
    //(< P_Th2), no action needs to be taken since the two loss rates
    //are already well balanced.
    if (|P_B − P_L| < P_Th2)
        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 > θ · P_L) {
            L_r decreases once more by 1;
        }
    }
}
```

Fig. 3. The retry-limit adaptation algorithm

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 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

$$\mathbf{P_e} = [0.35\ 0.4\ 0.45],$$

and $T_e = 5$ seconds. The simulations are then performed either with fixed retry limits (in the range of [1, 16]) or with the RTRA deployed. Figure III 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 Figure III.

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

[3]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].

channel error model. Figure 5 shows the tracked retry limit by the RTRA. The average retry limit calculated from the trace is 3.75.

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

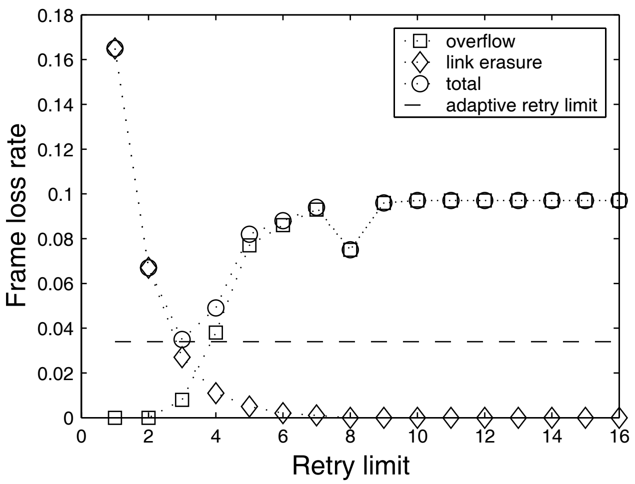

Fig. 4. Packet loss rates under fixed retry limits or the RTRA

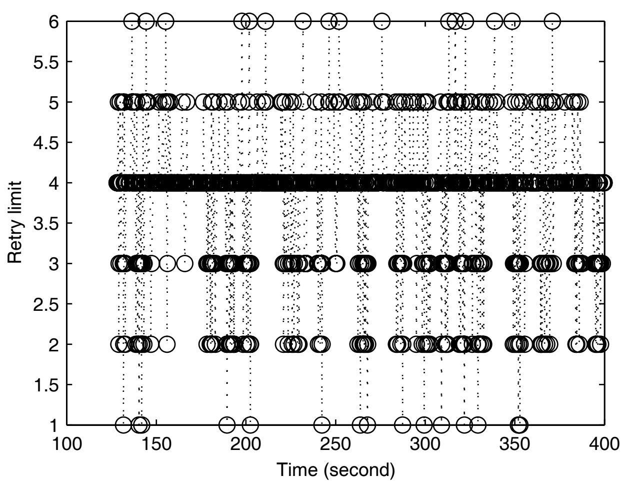

Fig. 5. The trace of retry limit adaptation

A. *Improvement on Video Quality*

To evaluate the effectiveness of RTRA on wirless video protection, a set of ns-2 simulations using the simulated two-node WLAN (see Section II-A.3 for detail) are performed. In order to create a realistic testing environment, we designed an FGS server and client and connected them to the ns-2 simulator.

For the experiments we coded a Standard Definition (SD) video clip at 25 Hz using FGS with a base layer of 1Mbps 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 6 smaller layers, each adding approximately 0.5dB gain in terms of the overall peak-signal-noise-ratio (PSNR). Then, 7 RTP/UDP channels are set up to send the base layer and the 6 enhancement layers.

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. Table III-A 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.

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

IV. CONCLUSION

In this paper, a cross-layer method that can provide adaptive error protection to video when transmitted over wireless networks is proposed. 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 MAC retry limit setting for the wireless link under which the total losses due to both link erasure and buffer overflow will be minimum. Furthermore, 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. Simulations show that the proposed RTRA can reduce packet losses over wireless link and improve link throughput, therefore resulting better video quality over wireless local area networks.

REFERENCES

[1] DARPA funded VINT project. UCB/LBNL/VINT network simulator - ns (version 2). see http://www-mash.cs.berkeley.edu/ns, 1996.
[2] L. Kleinrock. *Queueing Systems. Volume 1: Theory*. Wiley-Interscience, New York, 1995.
[3] L. Kleinrock. *Queueing Systems. Volume 2: Computer Applications*. Wiley-Interscience, New York, 1996.
[4] 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 Transactions on Circuits and Systems for Video Technology*, 12(6):524–534, June 2002.
[5] B. Girod and N. Farber, Wireless video, a book chapter in *Compressed Video over Networks*, Ed. Marcel Dekker 2001.
[6] Y. Wang and Q-F. Zhu, Error Control and Concealment for Video Communications: A Review, *Proceedings of the IEEE*, 86(5):974–997, October, 1998.
[7] H. Ma and M. El Zarki, MPEG-2 Video Transmission over Wireless Access Networks using Type-I Hybrid ARQ Schemes for VoD Services, in *Proc. Packet Video Workshop99*, April 1999.