# Time-based Adaptive Retry for Wireless Video Streaming[1]

Mei-Hsuan Lu, Peter Steenkiste, and Tsuhan Chen

Electrical and Computer Engineering Department
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213-3890, USA
Tel. (412) 268-7114, Fax (412) 268-3890
meihsual@ece.cmu.edu, prs@cs.cmu.edu, tsuhan@ece.cmu.edu

## Abstract

Video streaming over time-varying, error-prone wireless LANs (WLANs) poses many challenges. One problem is that WLANs are designed without awareness of the characteristics of application data, which causes performance degradation in a noisy or congested environment. In this paper, we propose a Time-based Adaptive Retry (TAR) mechanism for MPEG-like video streaming over 802.11 wireless networks. TAR dynamically determines whether to send or discard a packet based on its *retransmission deadline* instead of adopting a static retry limit uniformly over all the packets. Our approach can adapt the retry limit for each individual packet, thus providing indirect unequal error protection over different types of video frames. Analytical and simulation results show that TAR significantly improves video quality and saves channel bandwidth. We also describe a preliminary software-based implementation of TAR and use it to demonstrate the practicality of the proposed approach.

***Key Words*:** adaptive retransmission, 802.11 MAC, video streaming, cross-layer optimization

[1] This work is supported in part by the Institute for Information Industry (III), Taiwan.

# 1 Introduction

With the wide adoption of 802.11 WLANs, wireless video streaming has gained a lot of attention. However, due to the time varying, error prone nature of WLANs, wireless video streaming poses many challenges. Many solutions have been proposed to improve the quality of wireless video streaming. One approach is to apply additional controls in the application layer at the video source. Examples of this type of solution include selective dropping [1], FEC, ARQ, or Hybrid ARQ [2], and application layer scheduling [3]. These end-to-end techniques perform well in the wireline environment, but may not react promptly in a more dynamic wireless network. Another method is introducing an additional server on the boundary of the wired and wireless networks. Representative examples include video transcoding [4], adaptive video caching [5], and dynamic rate shaping [6]. Since the bottleneck of a connection usually happens on the "last-mile" wireless link, those solutions react to channel variations more quickly. However, introducing an intermediate entity also causes extra delay and requires additional investments.

Recently, researchers have proposed the idea of cross-layer design to combat the challenges of wireless video streaming [7]. While the conventional layered architecture reduces the network design complexity, it is believed that the layered strategy does not always result in optimal overall performance for wireless video streaming. Moreover, several duplicate protection strategies are implemented at various layers, causing unnecessary overhead. For video streaming applications with high bandwidth requirements coupled with tight delay constraints, a cross-layer design that optimally adapts to both underlying channel conditions and specific application requirements can result in substantial performance gains. Our work falls in this category of solutions.

In this paper, we present a cross-layer mechanism for video streaming over 802.11-like[2] wireless networks. The 802.11 specification [8] recommends a simple retransmission mechanism: a failed packet is retransmitted several times until a static retry limit is reached. We argue that this count-based retransmission scheme should be replaced with a time-based one to provide better quality of service for delay sensitive video streaming applications. With time-based logic for media access control (MAC) layer retransmission, the

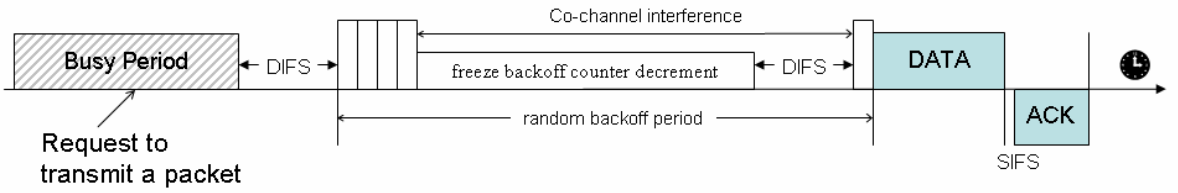
[2] The proposed scheme can be applied to any network with local retry.

media access control (MAC) layer dynamically determines whether to send or discard a packet based on a *retransmission deadline* attached by the video server. This can significantly reduce the number of late packets. In addition, the proposed mechanism can leverage the inter-coded property of today's video coding standards [9] [10] to provide unequal error protection over different types of video packets. Both simulation and experimental results show that our approach outperforms the count-based retransmission mechanism in terms of video quality.

The rest of this paper is organized as follows. We review the conventional 802.11 retransmission mechanism in the next section. Section 3 presents the proposed time-based retransmission algorithm where an initial delay of one GOP (Group Of Pictures) period is sufficient. Section 4 gives a brief introduction of video coding properties, which leads to a method for selecting *retransmission deadlines*. Section 5 analytically compares the proposed scheme with a widely known cross-layer protection strategy presented in [8]. Section 6 presents our simulation and experimental evaluation results. Finally, we discuss related work in Section 7 and summarize in Section 8.

# 2 802.11 Retransmission


(a) The channel is idle for a DIFS interval before the transmission




(b) The channel is busy during a DIFS interval before the transmission

**Figure 1.** Data transmission in the 802.11 standard

In this section, we review the conventional retransmission scheme adopted in the 802.11 standard [8]. Today's most popular 802.11 protocols include 802.11a, 802.11b, and 802.11g. 802.11b was released first and can operate at speeds of 1, 2, 5.5 and 11 Mbps, referring to four different physical (PHY) modes. Each PHY mode represents a particular modulation scheme and code rate. While higher speeds offer a higher data rate, they are also more vulnerable to noise and interference. 802.11a and 802.11g use an Orthogonal Frequency Division Multiplexing (OFDM) to provide eight different PHY modes ranging from 6 to 54 Mbps. 802.11a operates at a different frequency band from 802.11b/g.

The 802.11 MAC defines two forms of medium access, Distributed Coordination Function (DCF) and Point Coordination Function (PCF). PCF is a centralized access scheme in which the access point grants access to the channel to stations that request it. PCF was never deployed commercially. We therefore restrict our study to the DCF access scheme. The DCF mode is a contention-based medium access scheme. As shown in Figure 1, before transmission, a station first senses the wireless channel to detect whether the channel is busy or idle. If the channel is idle for a distributed inter frame space (DIFS) interval, it transmits immediately (Figure 1(a)); otherwise it backs off for a randomly selected time interval based on the current contention window size (Figure 1(b)). The backoff interval is a certain number of backoff slots, where the length of a backoff slot is medium-dependent. If the medium becomes busy due to other co-channel interference during the backoff period, the station holds the backoff counter until the channel is detected idle again for a DIFS interval. After sending a packet, the station can infer frame loss by the lack of a positive acknowledgement from the receiver. To deal with transmission error, the sender increases the contention window size to the next power of two until the maximal window size is reached. Based on the current contention window, a new backoff time is chosen and the same backoff process as described earlier in this paragraph is repeated. This retransmission process continues until a static retry limit is reached. At that point, the MAC layer abandons the current packet and starts to serve the next one.

Allowing a larger retry limit when several competing stations are attempting to gain access to the medium increases the chance of successful transmission of a particular packet. However, a larger retry limit can also result in a longer delay in the retransmission process, which can be harmful to delay-sensitive data such as video. Recall that the backoff time counter is decremented as long as the channel is sensed idle, but is

"frozen" when a transmission is detected on the channel and reactivated when the channel is sensed idle again for more than a DIFS interval. Therefore, the actual time taken in the retransmission process is heavily dependent on the activities of competing traffic. It is crucial to select an appropriate retry limit so that the timing constraint of video data is not violated.

The exponential backoff mechanism is borrowed from the 802.3 standard (Ethernet) to resolve collisions in a shared network. While this technique works well in a wired medium, several issues are raised when it is used in transmitting video data over wireless networks. In the rest of this section, we present a simple analysis to demonstrate the problems of such count-based retransmission scheme, operating in a noisy and a congested channel, respectively.

## 2.1 Noisy channel

To gain some insight into the impact of 802.11 retransmissions in a noisy channel, let us consider the time taken in the retransmission process for a single packet (i.e. from the time point when a packet is ready for its initial transmission until the point when the retry limit is reached). Suppose $R$ denotes the retry limit; $CW_{min}$ and $CW_{max}$ denote the minimal and maximal contention window sizes, respectively. For an $L$-byte long packet transmitted in PHY mode $m$, the expected transmission time without the intervention of competing traffic for reaching $R$ retries is:

$$\mathrm{T}_R^m(L) = \sum_{i=0}^{R} (T^m(L) + \frac{\min(2^i \cdot (CW_{\min} + 1) - 1, CW_{\max})}{2} aSlotTime) \qquad (1)$$

where $T^m(L) = T_{DATA}^m(L) + aSIFSTime + T_{ACK}^m + aDIFSTime$ is the transaction time for sending an $L$-byte long packet in PHY mode $m$. Here $T_{DATA}^m(L)$ and $T_{ACK}^m$ are the transmission time for an $L$-byte long data packet and the corresponding acknowledgement; *aSIFSTime* and *aDIFSTime* are the SIFS and DIFS intervals as described in the 802.11 specification. If we consider an 802.11a network with $R$ = 4, $L$ = 1024, and PHY mode = 5 (24 Mbps), with the 802.11a PHY characteristics given in Table 1 we obtain $\mathrm{T}_R^m(L) = \mathrm{T}_4^5(1024) = 4.21\,\mathrm{msec}$ , which is relatively small compared to the fading duration in an indoor

environment [11]. As a result, it is relatively easy for a packet (or several consecutive packets) to reache the retry limit during deep fading. If this packet happens to belong to a reference video frame, the resulted video distortion will propagate all the way to the end of the GOP, causing serious quality degradation at the receiver. In this case, the retry limit is too conservative and a larger retry limit is needed.

**Table 1**. IEEE 802.11a PHY characteristics

| *Characteristics* | *Value* | *Comments* |
|---|---|---|
| *aSlotTime* | 9 $\mu$s | Slot time |
| *aSIFSTime* | 16 $\mu$s | SIFS time |
| *aDIFSTime* | 34 $\mu$s | DIFS time |
| $CW_{min}$ | 15 | Minimal contention window size |
| $CW_{max}$ | 1023 | Maximal contention window size |

## 2.2 Congested channel

Now let us consider another situation where the channel is heavily loaded with traffic from multiple contending stations. In a congested network, most packet loss results from collisions. As described in the beginning of this section, upon a packet loss, the wireless station randomly chooses a backoff time and tries to retransmit the packet after the backoff time expires. However, during the backoff period, other contending stations may "seize" the channel and force the backoff counter to "freeze" until the channel is sensed idle again for more than a DIFS interval. Therefore, the actual length of the backoff period can be much longer than expected. To understand the possible impact on video data under such channel condition, we use an analysis similar to that of Section 2.1. We borrow the concept of *conditional collision probability*[3] as described in [12]. Assume the channel is "seized" by other contending stations with a constant and independent probability *p* at each time slot. Then (1) can be rewritten as:

$$\mathrm{T}_R^m(L) = \sum_{i=0}^{R} \left(T^m(L) + \frac{\min(2^i \cdot (CW_{\min}+1)-1, CW_{\max})}{2}(p \cdot T^m(L) + aSlotTime)\right) \quad (2)$$

[3] The *conditional collision probability* is the probability of a collision seen by a packet being transmitted on the channel. Interesting readers may refer to [12] for the details.

where we assume all the contending stations transmit packets with the same length *L* and physical mode *m* as the backoff station. If we apply the same parameters as in Section 2.1 and assume the *conditional collision probability* $p$ = 0.8, we then obtain $T_R^m(L) = T_4^5(1024) = 83$ msec . This value is much larger than the expected backoff time with no or low competing traffic (4.21 msec).

Consider a video sequence with a frame rate or 15 frames/sec (an inter-frame interval is 67 msec). If a packet takes 83 msec to transmit, then the transmit time for a video frame (which usually includes multiple packets) can be hundreds of milliseconds, which is much longer than the inter-frame interval, so the frame is likely to be late for playback. Even worse, transmitting late packets delays subsequent eligible packets, causing more and more packets to be late, that is, the delay is cumulative. Introducing some level of initial delay at the receiver may postpone the occurrence of this effect, but it cannot completely solve the problem. When the retransmission process takes too long to send a packet, the retry limit is too aggressive and a smaller retry limit is preferred.

Congestion and noise are typical features of the volatile wireless medium. The time-varying nature of wireless networks makes it difficult to select a static retry limit that is suitable for video data under all conditions. A dynamic retry mechanism that can adapt to channel conditions is preferable. In the next section, we propose a Time-based Adaptive Retransmission mechanism, abbreviated as TAR, to improve the performance of wireless video streaming. TAR is devised to support adaptability in the retransmission process while remaining compatible with the 802.11 protocol.

# 3 Time-based Adaptive Retransmission (TAR)

Figure 2 shows the main idea of TAR. Unlike the conventional 802.11 MAC which discards a packet when a predefined, static retry limit *N* is reached, TAR adaptively decides whether to discard or (re)send a packet based on a *retransmission deadline D* attached to that packet. The retransmission deadline is assigned by the application layer according to the application's specific requirements for the transmitted media data, and it is used by the MAC layer to decide when to stop the retransmission process for each packet. A TAR-aided MAC keeps retransmitting a packet as long as its *retransmission deadline* is larger than the current time.

Therefore, the exact retry limit is not determined *a priori* but dynamically adapted to network conditions. TAR is a cross-layer protection mechanism since it involves the collaboration between the application and MAC layers.

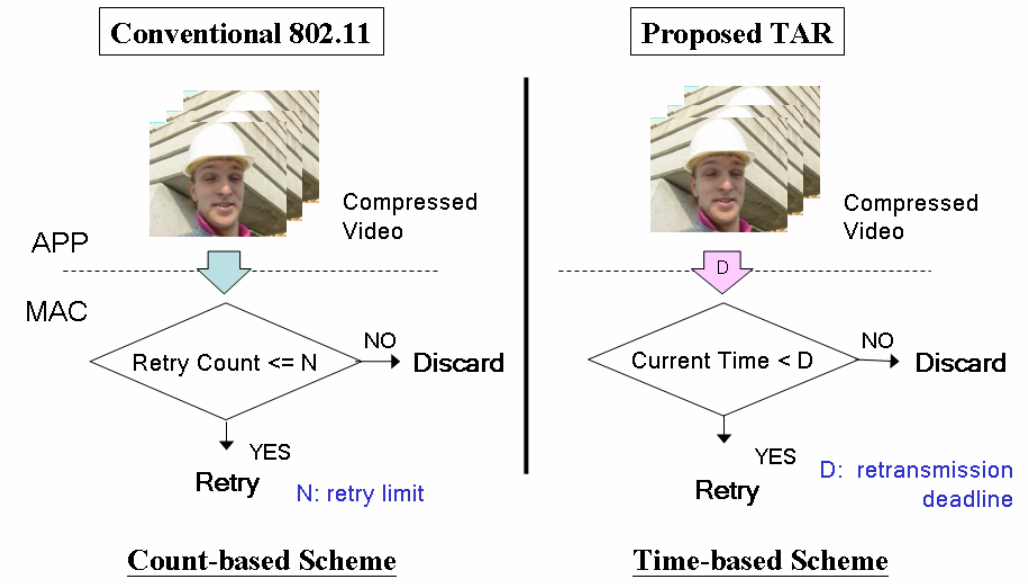



**Figure 2.** Main idea of TAR

TAR has the following attractive properties for wireless video streaming:

- *Adaptation to traffic activities and channel conditions:* As discussed in Section 2, the random backoff period in a noisy channel or a heavily loaded channel can be quite different. TAR adaptively determines the retry decision based on the retransmission deadline so that late packets are directly dropped at the sender without wasting network bandwidth.
- *Automatic indirect Unequal Error Protection:* In general, video data are inter-coded, resulting in differences in the importance of the various types of frames. Therefore, stronger protection should be applied to more important video packets than to the other ones. Through careful assignment of retransmission deadlines, the system can provide unequal error protection for different types of packets. Specifically, a far off deadline will result in more retries (and hence, stronger protection against transmission errors) is assigned to more important packets while a closer deadline, and thus fewer retries, is assigned to less important ones. This property is similar to the unequal error

protection (UEP) applied in forward error control.

- *Equal channel access opportunity*: TAR maintains the same 802.11 channel access opportunity by mimicking the transmission behavior of the conventional 802.11 protocol. Specifically, the contention window size is adjusted based on the conventional 802.11 protocol so that the time-based retransmission behavior is transparent to other stations. As is shown in the example in Figure 3(a), if the retry limit is set to four, a TAR-aided station will be configured to reset its contention window size at its fifth retry, mimicking the case as if the old packet is discarded and a new packet is being transmitted, even though the same packet is being transmitted. As for another example shown in Figure 3(b), if a TAR-aided station decides to discard a packet after the second retry (due to retransmission deadline expiration), it will not reset the contention window size for the transmission of the next packet. The contention window size is reset until after the second retry of the new packet, mimicking the conventional channel access behavior defined in the 802.11 specification. This property assures equal channel access opportunities for TAR and non-TAR stations.

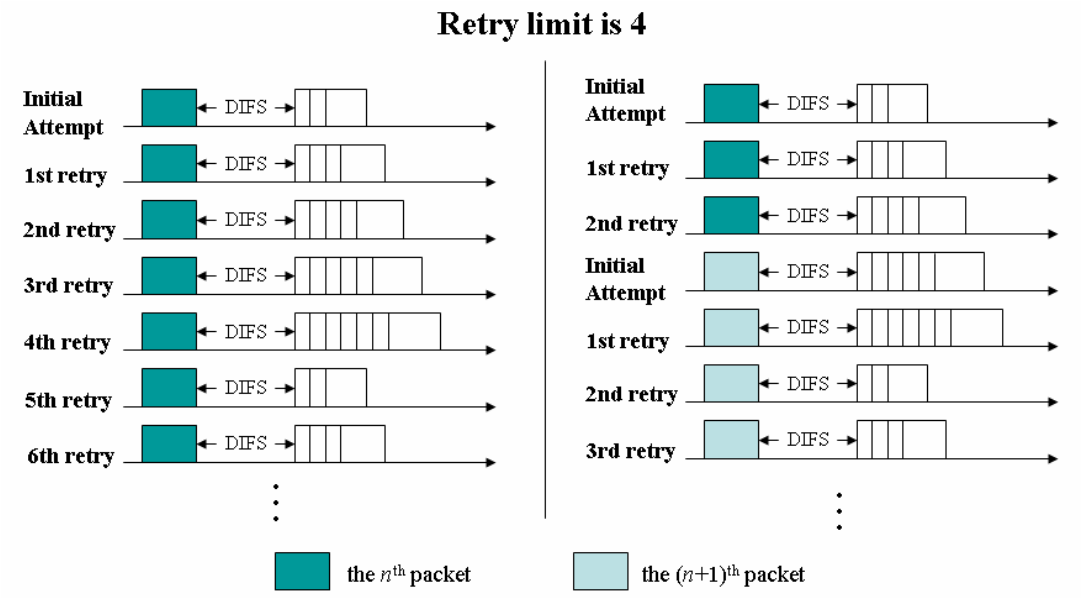



**Figure 3.** Examples of TAR's transmission behavior to mimic the conventional 802.11 protocol with retry limit four when the adapted number of retransmissions is (a) larger than the retry limit and (b) smaller than the retry limit.

Simply put, TAR replaces the conventional “count-based” retransmission scheme with a “time-based” retransmission strategy, which is preferable for serving delay sensitive video data. With the proper assignment of retransmission deadlines, such time-based retransmission method provides better user perceived quality. In the next section, we present a simple method for determining the retransmission deadlines for MPEG-like inter-coded video.

# 4 Retransmission Deadline

We present a simple method to determine the retransmission deadlines for pre-encoded video data. This solution takes advantage of the inter-coded property of today’s video coding techniques (e.g. [9] [10]). It is well known that successive frames have “temporal redundancy”, so a good method of compression is to transmit only the changes in information from one frame to the next. In general, this technique defines three types of frames: I frame, P frame, and B frame. I (Intra-coded) frames are constructed without reference to any other frames. I frames are larger than the other types of frames because they contain all the information needed to reconstruct a frame. P (predicted) frames are predicted from past frames. Only the difference between the previous frame and the P frame is transmitted. Because of the temporal redundancy, P frames contain much less data than I frames. Finally, B (bidirectional predicted) frames contain only the information that cannot be derived from previous and future frames. B frames are typically the smallest frames. The typical video encoding structure is shown in Figure 4. The repeated pattern of a leading I frame followed by optional P and B frames is known as a Group of Pictures (GOP).

While reducing temporal redundancy significantly increases the compression ratio, it also causes the problem of “error propagation”. That is, packet loss of a reference frame will propagate across multiple frames which are inter-coded with respect to the reference frame. In other words, receiving an inter-coded frame is useless if its corresponding reference frame is lost. The goal of our solution is to prioritize those important reference frames by giving them a later retransmission deadline.

Consider a video sequence with GOP size $\alpha$ and inter-frame interval $\lambda$. Without losing generality, we assume $\alpha$ and $\lambda$ are fixed and known *a priori*. As shown in Figure 5, the video sequence is expressed by

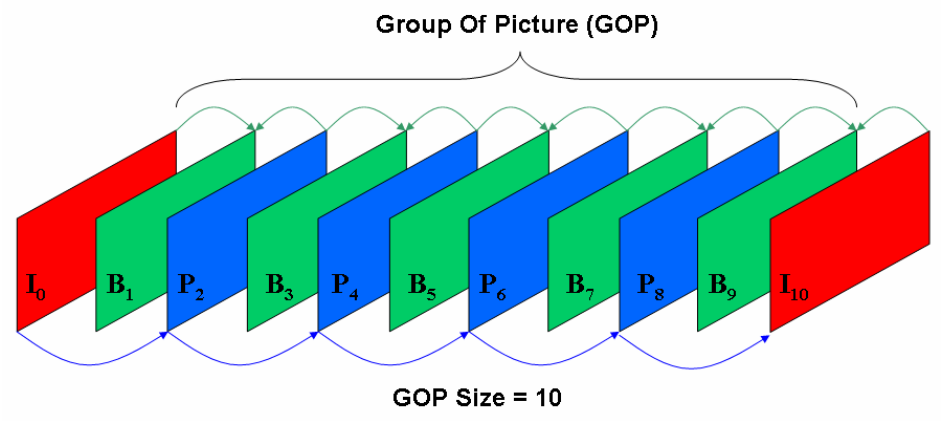



**Figure 4.** Typical inter-coded video frame structure

$$S = \{F_{1,1} F_{1,2} F_{1,3} \ldots F_{1,\alpha} F_{2,1} \ldots F_{i,j} \ldots\}$$

where $F_{i,j}$ denotes the $j$-th frame within the $i$-th GOP. Frame $F_{i,j}$ is composed of video packets with $P_{i,j}^{(k)}$ denoting the $k$-th video packet in $F_{i,j}$. For each $P_{i,j}^{(k)}$, We define the following *retransmission extension period* in seconds:

$$R(P_{i,j}^{(k)}) = \lambda (M(F_{i,j}) + 1) \qquad (4)$$

where $M(F_{i,j})$ is the number of frames inter-coded with respect to $F_{i,j}$. For simplicity, all video packets corresponding to the same frame are assigned the same $R(\cdot)$, which is shown as the dashed lines in Figure 5. For example, for the first I-frame, we assign $7\lambda$ as its *retransmission extension period* because 6 frames are directly or indirectly inter-coded with respect to it and we add one for the frame itself. Similarly, for the first P-frame, we assign $6\lambda$ as its *retransmission extension period* and for the first B frame, we assign $\lambda$ as its *retransmission extension period* because no frames are inter-coded with respect to it. By associating a reference frame (and all its composing video packets) with a larger $R(\cdot)$, Eq. (4) represents the relative importance of each packet in terms of its error propagation properties.

Given the *retransmission extension period* defined in (4), the wireless sender can compute the *retransmission deadline*, that is, the time instant when the retransmission process stops and the packet is discarded. We assume that video is played strictly following the original temporal relationship at the receiver

(no stretching or shrinking of the display time). We formulate the *retransmission deadline* $D(P_{i,j}^{(k)})$ for video packet $P_{i,j}^{(k)}$ as:

$$D(P_{i.j}^{(k)}) = ((i-1)\alpha + (j-1))\lambda + R(P_{i.k}^{(k)}) \qquad (5).$$

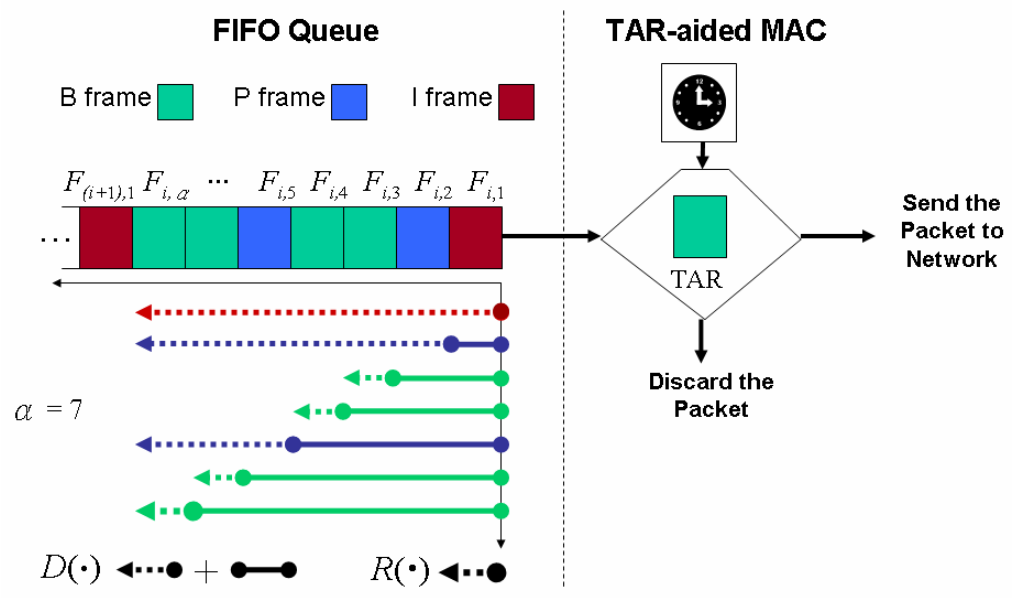



**Figure 5.** System diagram of a TAR-aided MAC

Our adaptive retry technique works as follows: Just before packet $P_{i,j}^{(k)}$ is to be (re)sent, the MAC compares the current time with $D(P_{i,j}^{(k)})$. If the current time is less than $D(P_{i,j}^{(k)})$, an initial transmission or a retry is issued. Otherwise this packet is dropped and the process is repeated for the next packet in the queue.

Note that (5) is derived by assuming a maximal initial delay of one GOP period ($\alpha\lambda$), which means one GOP-size buffer should be sufficient at the receiver. We count on an out-of-band signaling protocol to synchronize this information between the video sender and the video receiver. This initial delay is well within the range of acceptable delays (1-10s) for noninteractive video streaming [14]. One can extend the retransmission deadline in (5) to create more retransmission opportunities in a fading channel. Larger retransmission delay implies a larger smoothing window and allows the wireless network to recover from longer error periods. The tradeoff is the requirement of a larger buffer and a longer initial delay at the receiver.

# 5. Comparison with the FEC_ARQ strategy

As mentioned in Section 1, many cross-layer solutions have been proposed for wireless video streaming. In general, the methodology is to select the values of the protocol parameters to maximize or minimize an objective function. In this section, we compare TAR with the "Adaptive Application-Layer FEC and MAC-Layer ARQ" or FEC_ARQ strategy proposed in [14]. We select the cross-layer method proposed in [14] for two reasons. First, the authors presented a very comprehensive evaluation of different error control and adaptation mechanisms available in different layers for wireless video streaming. Secondly, the method proposed in [14] is an end-to-end approach which requires no support from the underlying network. We are interested in comparing this end-to-end method with our local time-based mechanism.

The FEC_ARQ strategy is an end-to-end approach that considers the MAC retry limit, application-layer forward error correction (FEC), and adaptive packet size selection in an integrated manner. The best set of retry limit, Reed-Solomon (RS) FEC parameters, and packet size are chosen in terms of maximal throughput efficiency, given certain channel conditions (bit error rate). Because the current 802.11 WLANs do not employ FEC, the application-layer FEC is used.

The FEC_ARQ strategy adopts a cross-packet RS coding for video packets at the application layer. It is necessary to apply the RS coding across video packets because 802.11MAC implementations discard the entire MAC frame whenever there is an error. As shown in Figure 6, $K$ video packets each of length $L_a$ bytes are buffered at the interleaver. The first symbol (byte) from each of the $K$ packets is sent through a ($N$,$K$) RS coder resulting in ($N$-$K$) parity symbols, which form the first byte of the ($N$-$K$) parity packets. This is repeated for the $L_a$ bytes resulting in ($N$-$K$) parity packets each of length $L_a$. Each video or parity packet is transmitted via an IEEE 802.11 MAC frame. The loss of a frame results in a symbol erasure at the RS decoder in the application layer. The application-level RS decoder can correct up to ($N$-$K$) packet losses out of $N$ packets.

We choose the optimal packet size given current link layer parameters such as SINR, bit error rate (BER), symbol rate, and constellation size for maximal throughput. In general, the packet length $L_a^*$ that maximized the throughput is given by [7]

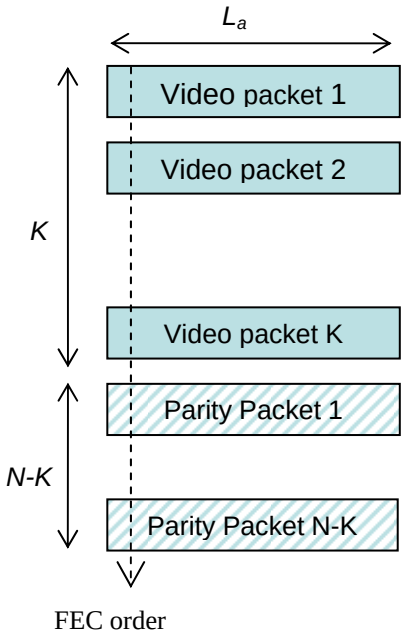



**Figure 6.** (*N*,*K*) APP-Layer RS Code [14]

$$L_a^* = \frac{H}{2} + \frac{1}{2}\sqrt{H^2 - \frac{4b_m H}{\ln(1-p_e)}} \qquad (6)$$

where $H$ is the size of the packet header, $b$ is the number of bits per symbol, and $p_e$ is the probability of a symbol error, which depends on the modulation type and link SINR. The optimal packet size will be used for both approaches in the following analysis. For simplicity, we only consider the base layer video stream encoded with I and P frames. Therefore, all the frames within a GOP have the same retransmission deadline, which is at the end of the GOP period. For performance comparison, we study the expected number of correctly-received video packets.

Let us first consider FEC_ARQ: Assuming the symbol size is 8 bits ($b$ = 8) and the symbol error rate is fixed, the probability of a packet error with length $L$ bytes is

$$p = 1-(1-p_e)^L \qquad (7).$$

Eq. (7) can be generalized with any bit error model (e.g. bursty errors). With a fixed retry limit $R$, the packet error rate $p_R$ after $R$ retransmissions is given by:

$$p_R = p^{R+1} \quad (8).$$

The expected number of retries can be computed as

$$r = \sum_{i=0}^{R} (i+1) \cdot (1-p) \cdot p^i + (R+1) \cdot p^{R+1} \quad (9).$$

An (*N*,*K*) RS decoder can correct up to (*N*-*K*) packet erasures - more than (*N*-*K*) packet erasures results in a decoding failure. Therefore, if we assume no packet fragmentation and concatenation in the lower layers ($L_a$ = *L*), the probability of packet error after RS decoding is given by

$$p_{RS} = 1 - \sum_{i=0}^{N-K} \binom{N}{i} p_R{}^i (1-p_R)^{N-i} \quad (10).$$

When a decoding failure happens, the receiver will have *N* - *i* (*i* < *K*) correctly-received packets, including both video and parity packets. On average, (*K*/*N*)(*N* – *i*) packets out of *N* – *i* correctly-received packets should be video packets. Hence, the expected number of correct video packets can be obtained as

$$n_{fec\_arq} = K(1-p_{RS}) + \sum_{i=N-K+1}^{N} (N-i)\frac{K}{N} \binom{N}{i} p_R{}^i (1-p_R)^{N-i} \quad (11).$$

For cross-layer optimization, $n_{fec\_arq}$ is maximized by choosing optimal *N*, *K*, and *R* in [14], given a packet error rate *p*.

Now let us consider TAR under the same channel condition. That is, given the same number of transmission opportunities as FEC_ARQ, we are interested in how many video packets TAR can send successfully. Since TAR utilizes the full channel capacity for video packets, the expected number of correct video packets in TAR can be written as

$$n_{tar} = N \cdot r \cdot (1-p) \quad (12).$$

Eq. (12) can be explained as follows: Given the average number of retries *r* in FEC_ARQ, the expected number of transmissions, including both failed and successful ones, in serving *N* packets is $N \cdot r$. With a packet error rate *p*, the expected number of successful transmissions is $N \cdot r \cdot (1-p)$. Since TAR only sends video packets (FEC is not adopted), this corresponds to $n_{tar}$ in (12).

Figure 7 compares TAR and FEC_ARQ for *N*=63, *R*=8. The parameter values are chosen to be consistent with [14]. In Figure 7 (a), FEC_ARQ shows a linearly increasing number of correct video packets until *K* >

47; this corresponds to full FEC protection of the video data against channel errors. When $K$ increases further, the number of parity packets ($N$-$K$) is no longer sufficient to cover channel errors so the expected number of correct video packets starts to drop. As $K > 57$, the FEC provides almost no protection and the packet error rate is dominated by $R$, so the number of correct video packets increases proportionally to $K$. It is clear that TAR outperforms FEC with different values of $K$.

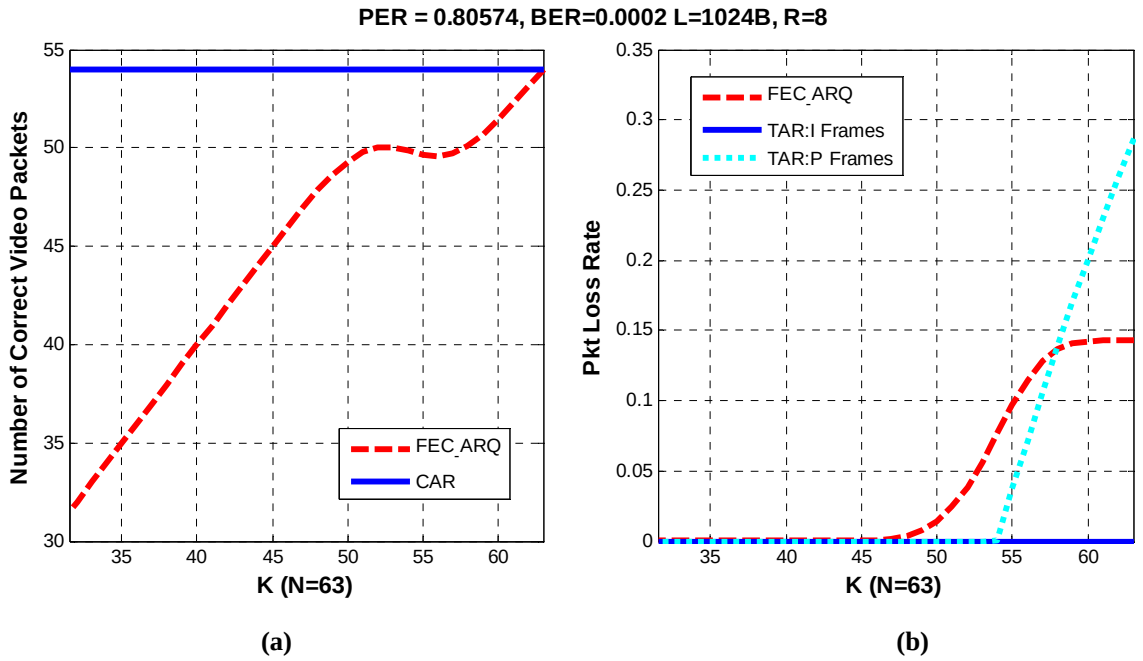



**Figure 7.** Comparison of TAR and FEC_ARQ with equal error protection (EEP) in (a) number of correct video packets and (b) packet loss rate.

In Figure 7 (b), the packet loss rate is given, under the assumption that we have an equal number of I-frame and P-frame packets (=$K$/2). We see that with the FEC_ARQ strategy, FEC cannot completely correct packet errors when $K > 47$. In contrast, TAR achieves error free transmission for I frames for any K thanks to its adaptive retransmission strategy. However TAR does suffer packet losses for P frames when $K > 54$ because at that point, the video bandwidth requirements exceed the channel capacity.

In general, the retransmission strategy should keep the packet loss rate of the reference frames small to avoid the error propagation effect. Figure 7 shows that FEC_ARQ's strategy of maximizing the number of correctly received video packets prevent it from operating at the optimal point. Selecting a large $R$ can reduce the gap between TAR and FEC_ARQ, but it also increases the probability of late arrivals in FEC_ARQ. In

contrast, TAR prioritizes I frame packets by adopting time-based adaptive retransmissions. TAR explicitly considers the delivery deadline in its retransmission strategy, which eliminates the need of FEC. In fact, one can prove that $n_{tar} > n_{fec\_arq}$ for any combination of ($N$,$K$) and $R$. A formal proof for this statement is given in the appendix. This performance gain is obtained from not sending parity packets and using the time-based adaptive retry within the GOP period.

Figure 7 presents a comparison between for TAR and FEC_ARQ using equal error protection (EEP) for FEC-ARQ, that is. FEC-ARQ does not consider the differences in packet types during optimization. To provide a fair comparison, we now consider FEC_ARQ with unequal error protection (UEP). We again assume that we have the same number of I-frame and P-frame packets (=$K$/2) and we use all the parity packets to protect I frames while there is no FEC protection for P frames. The comparison results are shown in Figure 8. In Figure 8 (a), we observe that the pattern of the number of correct video packets is similar to that in Figure 7(a). However, since all the parity packets are applied to I frames, the number of correct video packets in FEC_ARQ with UEP is smaller than that in FEC_ARQ with EEP. Nevertheless, this does result in a smaller packet error rate for I frames, as is shown in Figure 8 (b). With unequal error protection, the I-frame loss rate Is lower than the P-frame loss rate. Since there is no FEC for P frames, the P-frame loss rate is dominated by $R$ and is independent of $K$. We see that TAR again outperforms FEC_ARQ with UEP because it does not need to send parity packets but uses time-based retransmission to protect reference frames.

Besides using different error recovery strategies, TAR and FEC-ARQ also have different deployment requirements. FEC_ARQ falls into the category of end-to-end cross-layer approaches, which means that optimization decision is made by the source, so FEC_ARQ requires no support from the network. TAR does have to be integrated into the network, which makes deployment more difficult. However, because it resides in the network, TAR is more agile. For example, TAR can achieve good performance even under highly bursty error conditions because retransmissions are issued locally “on demand”. FEC_ARQ will need to enlarge its interleaving depth to recover from bursty errors. This may have two undesirable side effects. First, it may increase buffer space requirements and delay at the receiver to accommodate FEC decoding. Second, it may increase the amount of bandwidth that is wasted on parity packets when error rates are low, because it is difficult to do perfect application-level adaptation to wireless channel conditions.

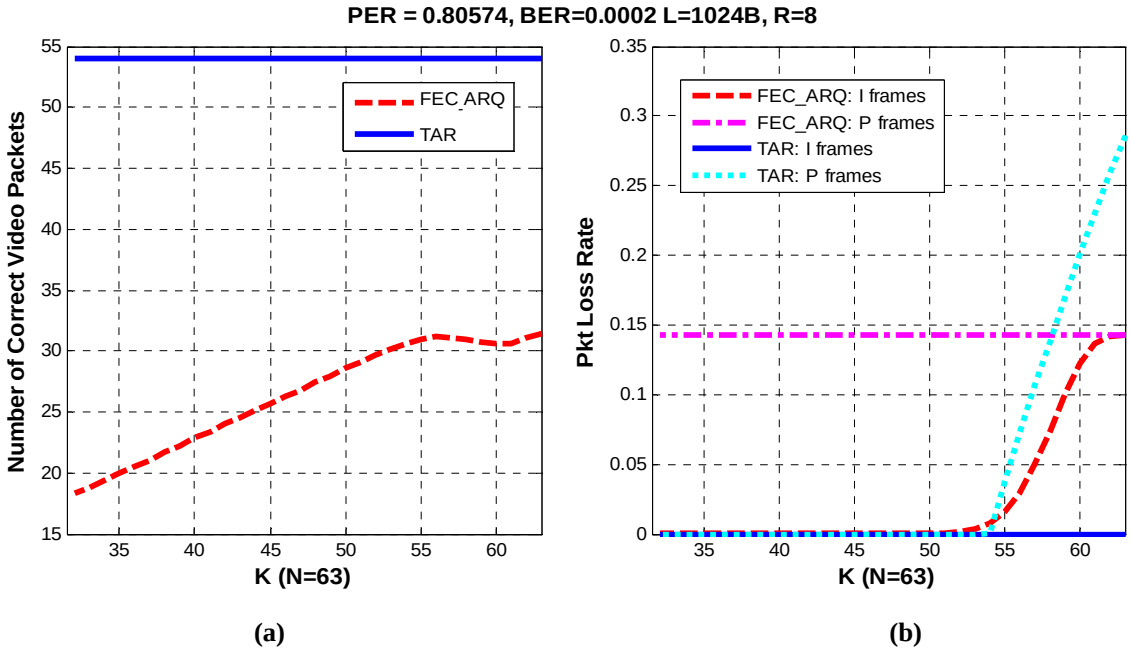



**Figure 8.** Comparison of TAR and FEC_ARQ with unequal error protection (UEP) in (a) number of correct video packets and (b) packet loss rate.

# 6. Evaluation Results

We use both simulation and experimental measurements to evaluate the performance of TAR. The simulation results are based on a congested noisy environment where multiple moving stations transmit video data concurrently. The experimental results reflect a more realistic comparison. We also describe our TAR implementation to demonstrate the practicability of TAR.

## 6.1 Simulation-based Evaluation

We use OPNET [15] 10.5.A to simulate an 802.11a independent basic service set (IBSS) with six stations. Two stations serve as the video sender/receiver pair and the others are contending stations with back-to-back traffic to transmit. The 802.11 MAC layer was modified to employ TAR. The test video sequence is "foreman" encoded in MPEG-4 CIF format with quantization step size 4, 30 frames per second, and 15 frames per GOP. The encoding pattern is IBBPBBPBBPBBPBB. Six copies of the sequence are concatenated to form a 60-second video stream. The video sequence is packetized into 1024-octet MAC frames with

average inter arrival time 5 msec (equal to the encoding rate). The initial delay is set to 500 ms (one GOP period) to accommodate the latency introduced by retransmissions as described in Section 3.

To simulate the wireless LAN environment, RF propagation is modeled using a Friis path loss large-scale radio propagation model and a Rayleign fading envelope [16]. The latter models the fading phenomenon on short time-scales, which arises due to moving transmitters, receivers, or objects along the transmission path. We configured each station with a walking speed of 0.5 m/s. The receive power thresholds for the different data rates are based on the Atheros 802.11a wireless LAN card datasheet [17].

We ran simulations for three retransmission mechanisms: (a) the proposed TAR mechanism, (b) a fixed retry limit (= 4), and (c) an infinite retry limit. For fair comparison, the fixed retry limit (=4) is chosen so that it is optimal in terms of the overall losses it will produce. Table 2 compares the packet loss rates for the three mechanisms. We distinguish between packet losses at the network (when the retry limit is reached or when the retransmission deadline expires after several failed trials), at the video sender (when the delivery deadline for a packet is earlier than the initial transmission time and the packet is discarded without any transmission attempts), and at the video receiver (when a packet arrives too late for playback). We see that TAR suffers the lowest total packet loss rate among the three mechanisms. Most packet losses in TAR occur at the sender, which suggests that TAR saves channel bandwidth by not sending useless packets that will inevitably be discarded at the receiver. In contrast, most packet losses with the other two mechanisms occur at the receiver, that is, after the packets have consumed channel bandwidth.

Figure 9 breaks down the packet loss rate by video frame type. The graph shows that TAR achieves indirect unequal error protection: the I frame packet loss rate is less than the P-frame packet loss rate, which in turns is less than the B-frame packet loss rate. Figure 10 uses the “ahead of schedule time” metric to compare the delays incurred by the different mechanisms. The “ahead of schedule time” metric represents the difference between the packet arrival time and the corresponding frame playback time. Hence, a positive value of the ahead of schedule time means an eligible packet while a negative value indicates a late packet. With a 500 ms initial delay, TAR does not suffer from late arrivals while the other two mechanisms do suffer delays that accumulate over time.

**Table 2.** Comparison of the packet loss rate

| Drop Location | Retry Limit | I-frame Packets | P-frame Packets | B-frame Packets |
|---|---|---|---|---|
| Sender | TAR aided | 0 | 0.30% | 13.87% |
| | Fixed at 4 | - | - | - |
| | Infinite | - | - | - |
| Network | TAR aided | 0 | 0 | 0.71% |
| | Fixed at 4 | 0.56% | 0.47% | 0.53% |
| | Infinite | 0 | 0 | 0 |
| Receiver | TAR aided | 0 | 0 | 0 |
| | Fixed at 4 | 48.81% | 48.81% | 50.00% |
| | Infinite | 89.32% | 88.78% | 89.76% |

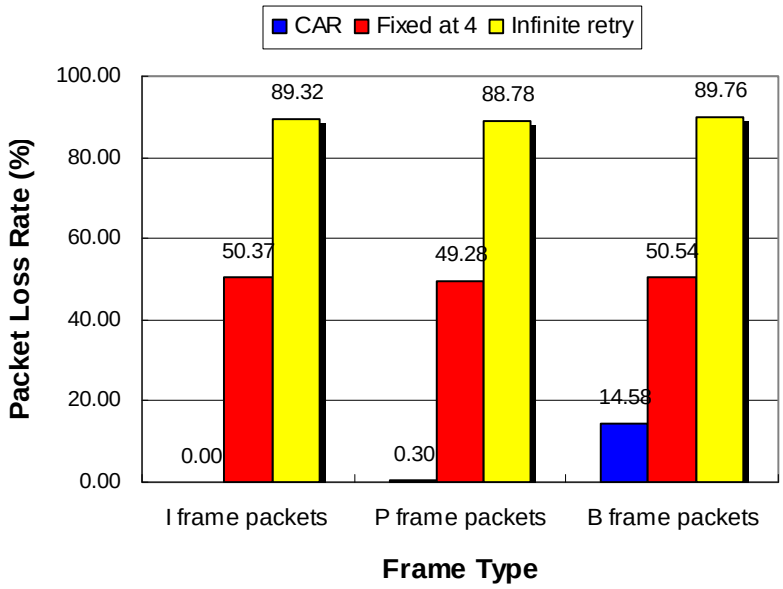



**Figure 9.** Packet loss rate for different types of video frames

The video quality can be assessed using the Peak Signal to Noise Ratio (PSNR) metric; a high PSNR value corresponds to a high visual quality. Figure 11 compares the PSNR of the Y component for the three mechanisms. Although several distortions occur, TAR consistently maintains a higher PSNR throughout the simulation period. The average PSNR of TAR, fixed retry limit (=4), and infinite retry are 37.44 dB, 24.1 dB, and 19.09 dB, respectively. The fluctuating pattern for the other two mechanisms is caused by error resilience behavior and the repeated video content (every 10 seconds).

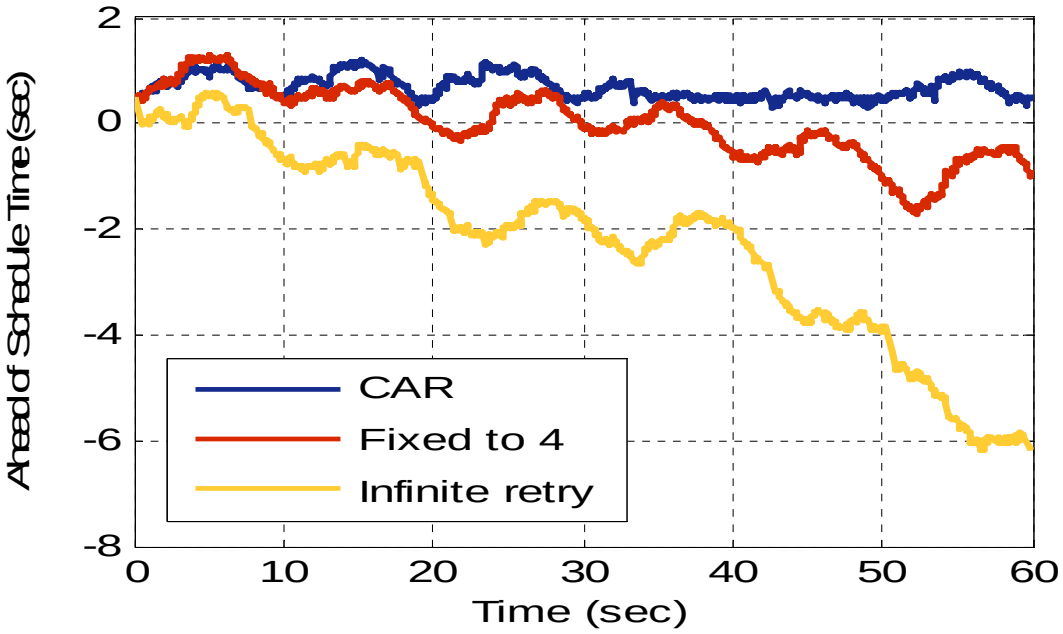



**Figure 10.** Comparison of the Ahead of Schedule Time

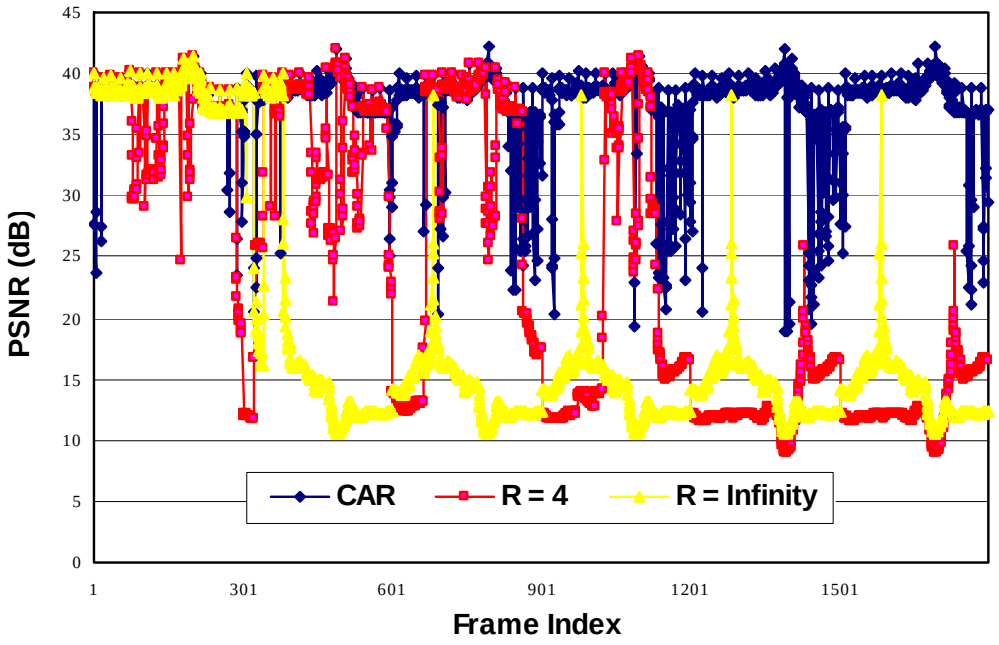



**Figure 11.** PSNR of the Y Component

## 6.2 Experimental Evaluation

Today's 802.11 products integrate the count-based retransmission operation in the chipset. Unfortunately, modifying the chipset to replace the conventional mechanism with proposed time-based scheme is not possible. Instead, we present a software implementation that incorporates the time-based operation into the count-based design. This preliminary implementation does not fully reflect the TAR algorithm, but it is sufficient to demonstrate the practicality and behavior of TAR. In this section, we first describe our software implementation and we then present some experimental results.

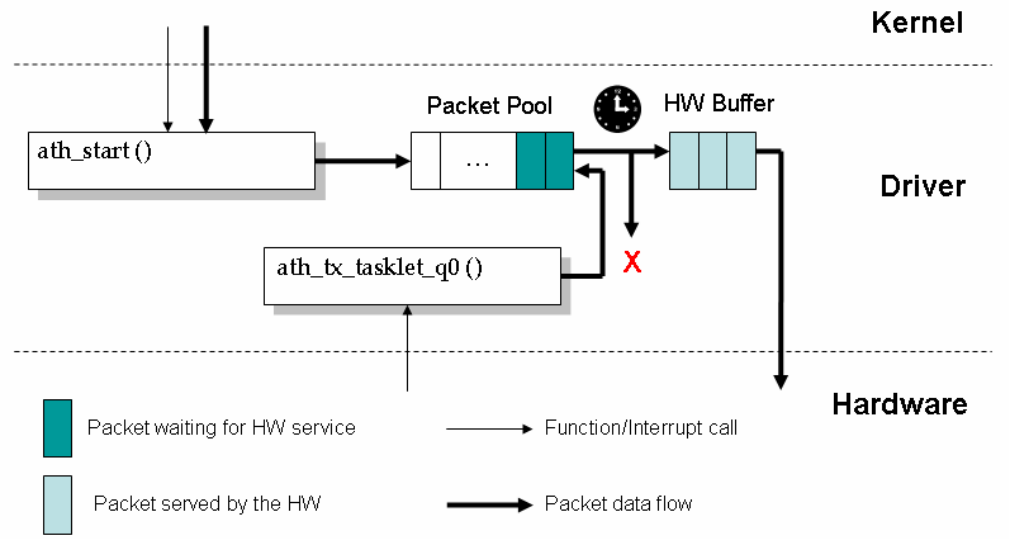



**Figure 12.** Software structure of TAR on the Atheros MADWIFI platform

### 6.2.1 Implementation

We use the Atheros AR 5212 chipset as our implementation platform. Our TAR implementation runs on Linux and uses the Multiband Atheros Driver from the WiFi (MADWIFI) project, available on SourceForge [18]. Atheros provides considerable flexibility in the MAC layer, allowing us to implement TAR. The video server application uses RTP's extension capability [13] to insert the retransmission deadline into each video packet.

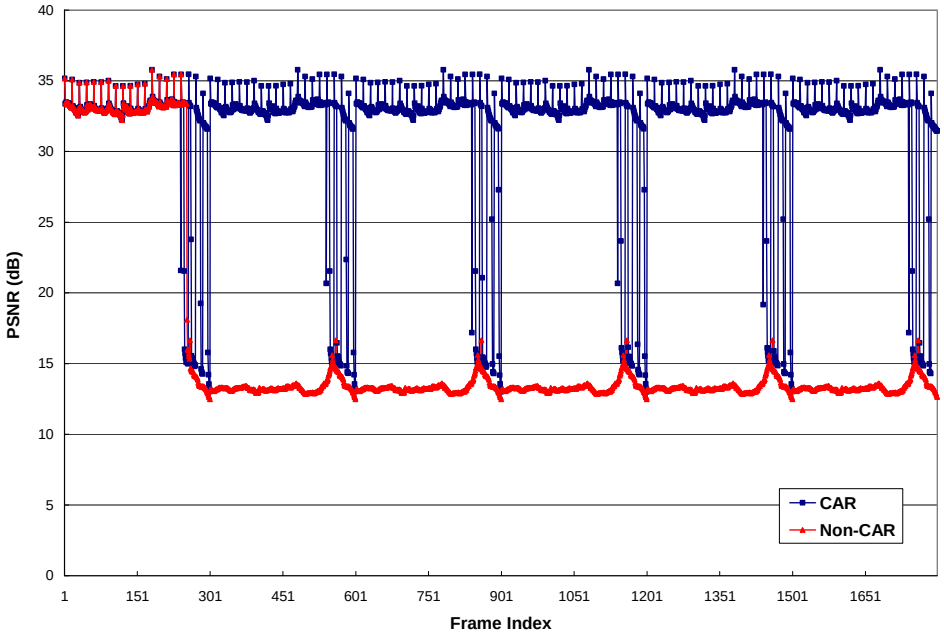
Figure 12 shows the operation of our software implementation. Whenever the protocol stack wants to transmit a new packet, the driver appends the packet to the tail of a driver-maintained "packet pool". The driver also regularly retrieves information on the current occupancy of the hardware buffer from the network interface. If the occupancy is below some threshold, it moves additional packets from the packet pool to the hardware buffer. The driver is also notified of the transmission status of completed packet by the interrupt handling routine ath_tx_tasklet_q0(). If a packet transmission failed due to excessive retries, the driver compares the retransmission deadline of that packet with the current time and it reinserts the packet at the head of the packet pool if the retransmission deadline has not expired; it discards the packet otherwise. By keeping occupancy of the hardware buffer low and by using a small (fixed) retry count we can approximate the operation of TAR by having the retransmission of packets be controlled by the per-packet retransmission deadline. Note that reinserting a failed packet at the head of the packet pool may cause unordered delivery. However, the out-of-order range is small since we use a low hardware buffer threshold.

### 6.2.2 Evaluation results

We conducted experiments in two scenarios. In the first scenario, a TAR-aided wireless client is associated with an 802.11b access point (AP). The video server and client are running on the wireless client and AP, respectively. In this scenario, we manually set the link layer rate to 2Mbps to simulate a distant client with rate adaptation support. The test video sequence is "foreman" encoded in MPEG-4 CIF format, using the same parameters as in Section 6.1. Late packets are discarded and not delivered to the decoder. In the second scenario, another wireless client is added to the IBSS, serving as a contending station injecting heavy competing traffic (using broadcast *ping*) into the network. In this scenario, the manual setting of the transmission rate is removed so the video server station now uses the highest rate (11Mbps) to connect with the AP. Otherwise, the same configuration is used as in the first scenario.

The experimental results are shown in Figure 13 and Figure 14. We observe that in both scenarios, TAR provides better video quality in terms of PSNR. When the bandwidth is not sufficient to support the full video

rate, TAR can deliver superior video quality by protecting the reference frames with a longer retransmission deadline.




.

**Figure 13.** PSNR of the Y Component in Scenario 1

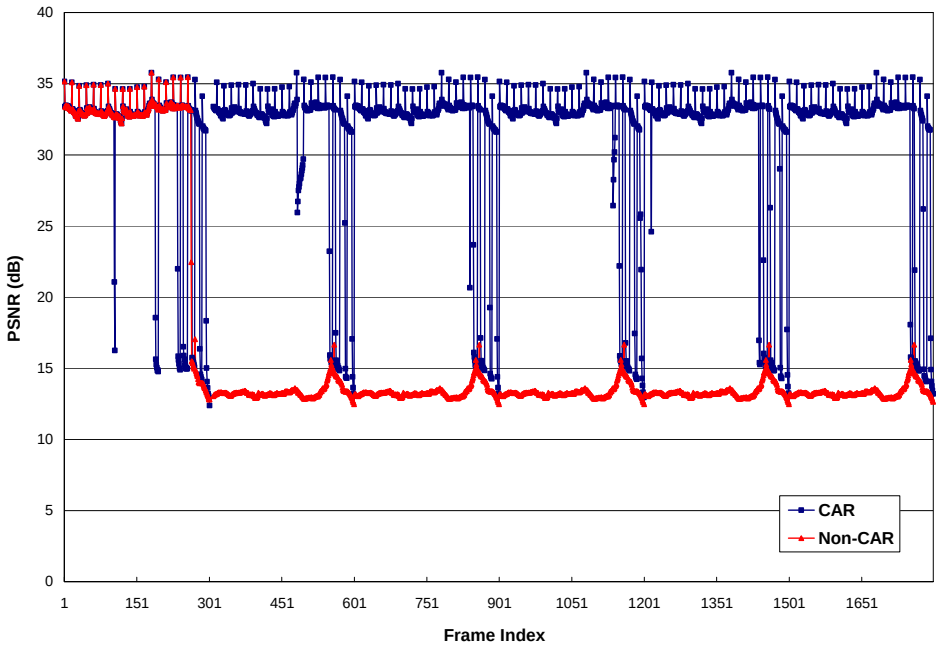



**Figure 14.** PSNR or the Y Component in Scenario 2

# 7. Related Work

Researchers have proposed several cross-layer designs to combat the challenges of wireless video streaming [7][14][19-22]. It is believed that the conventional layered strategy does not always result in optimal performance for wireless video streaming. Moreover, protection strategies are often replicated at multiple layers, causing unnecessary overhead. In [14], van der Schaar *et al*. propose a novel adaptive cross-layer protection strategy for enhancing the robustness and efficiency of video transmission. Solutions for MAC retransmission, application-layer forward error correction, scalable coding, and adaptive packetization are evaluated. In [19], Shan and Zakhor present an adaptation mechanism in which an application layer packet 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, thus reduce delay at the receiver compared with application level FEC solutions.

Representative work on the MAC layer retransmission includes [20-22]. Li and van der Schaar [20] consider the MAC retry limit and sending buffer occupancy and suggest a heuristic for determining the operating point at which packet loss due to buffer overflow and link errors is minimal. Unlike [20], our focus is on solving the problem of late packets because it is more critical than transmit buffer overflow. Liebl *et al*. [21] develop an integrated scheduler and drop strategy in the link layer. The proposed method requires accurate channel state information, which is not easy to obtain in practice. In [22], Liebl *et al*. incorporate information about the video stream structure and future channel behavior into the scheduling algorithm. A solution based on a simplified rate-distortion model that relies on one single metric per user is proposed. Our work can be viewed as a special case of [22]. However, our approach does not rely on knowledge of future channel behavior and on the availability of rate-distortion-quality triples in the base station.

All the abovementioned prior work uses simulation to validate the proposed solutions. We have implemented a version of TAR for the Atheros chipset, allowing us to do a more realistic evaluation and demonstrating the practicality of TAR.

# 8. Conclusions

In this paper, we present a Time-based Adaptive Retry (TAR) mechanism for video streaming over 802.11-like WLANs. The TAR algorithm dynamically considers the impact of competing traffic and the wireless transmission errors. Instead of adopting a static, pre-defined retry limit, TAR dynamically determines whether to (re)transmit or discard a packet based on the *retransmission deadline* attached to that packet. We propose a simple method to assign retransmission deadlines, exploiting the temporal relationship and error propagation characteristics of different video frames. An analytical comparison between TAR and a representative cross-layer solution is presented to demonstrate the benefit of time-based adaptive retry. Simulation results show that TAR outperforms the conventional persistent retry and fixed retry limit mechanisms considerably in terms of packet loss, channel utilization, and user-perceived visual quality.

We also implemented a version of TAR for the Atheros chipset. In our implementation, the video server application uses RTP's extension capability to embed the retransmission deadline into video packets. Experimental results show that TAR outperforms the count-based retransmission mechanism significantly.

# References


[1] K. Sripanidkulchai and T. Chen, "Network-Adaptive Video Coding and Transmission", Proc. Visual Communications and Image Processing, Vol. 3653, p. 854-861, San Jose, Jan. 1999.

[2] 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.

[3] S. H. Kang and A. Zakhor, "Packet Scheduling Algorithm for Wireless Video Streaming", in Proc. Packet Video Workshop 2002, Pittsburgh, USA, April 2002.

[4] J. Xin, C. W. Lin, and M. T. Sun, "Digital Video Transcoding", *Proceedings of IEEE*, vol.93, no. 1, pp. 84-97, January 2005. (invited paper)

[5] J. Liu, X. Chu and J. Xu, "Proxy Cache Management for Fine-Grained Scalable Video Streaming", Infocom 2004, Hong Kong, China.

[6] T. P.-C. Chen and T. Chen, "Fine-Grained Rate Shaping for Video Streaming over Wireless Networks", *IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), Volume: 5, on page(s): V- 688-91 vol.5, Hong Kong, 2003.

[7] E. Setton, T. Yoo, X. Zhu, A. Goldsmith and B. Girod, "Cross-layer Design of Ad Hoc Networks for Real-Time Video Streaming", *IEEE Wireless Communications Magazine*, vol. 12, no. 4, pp. 59-65, August 2005, *Invited paper*.

[8] 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.

[9] Motion Pictures Experts Group, "Overview of the MPEG-4 Standard", ISO/IEC JTC1/SC29/WG11 N2459, 1998.

[10] ITU-T Recommendation H.263, "Video Coding for Low Bit Rate Communication", Feb. 1998.

[11] M. Carroll and T.A. Wysocki, "Fading Characteristics for Indoor Wireless Channels at 5GHz Unlicensed Bands", *SYMPOTIC*, Oct. 2004.

[12] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function”, *IEEE Journal on Selected Areas in Communications*, Volume: 18, Issue: 3, on page(s): 535-547, March 2000.

[13] J. Ott, S. Wenger, S. Fukunaga, N. Sato, K. Yano, A. Miyazaki, K. Hata, R. Hakenberg, C. Burmeister, “Extended RTP Profile for RTCP-based Feedback (RTP/AVPF)”, Internet Draft, draft-ietf-avt-rtcp-feedback-02.txt, March 2002.

[14] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive Cross-Layer Protection Strategies for Robust Scalable Video Transmission over 802.11 WLANs", *IEEE Journal on Selected Areas in Communication*, Vol. 21, Issue 10, Dec. 2003, pp. 1752–1763.

[15] OPNET Technologies, Inc., http://www.opnet.com/.

[16] R. Punnoose, P. Nikitin, and D. Stancil, “Efficient Simulation of Ricean Fading within a Packet Simulator”, in *Proc. IEEE Vehicular Technology Conference*, pages 764-767, Sept. 2000.

[17] Atheros Commuincations, *Atheros Wireless LAN 2.4/5-GHz 802.11a/b/g 108 Mbps Turbo Radio-on-a-Chip WLAN Networking Products and Technology Overview*.

[18] Madwifi, http://sourceforge.net/projects/madwifi

[19] Y. Shan and A. Zakhor, "Cross Layer Techniques for Adaptive Video Streaming over Wireless Networks", *IEEE International Conference on Multimedia and Expo ( ICME)*, Aug. 2002.

[20] Q. Li, M. van der Schaar, "Error Protection of Video over Wireless Local Area Networks Through Real-time Retry Limit Adaptation", *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2004.

[21] G. Liebl, H. Jenkac, T. Stockhammer, and C. Buchner, "Joint Buffer Management and Scheduling for Wireless Video Streaming", *Proc. IEEE International Conference on Networking (ICN)*, April 2005.

[22] G. Liebl, M. Kalman, and B. Girod, "Deadline-Aware Scheduling for Wireless Video Streaming", *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, July 2005.

# Appendix

In this appendix, we formally prove $n_{tar}$ is always larger than or equal to $n_{fec\_arq}$. From (10) and (11), we can obtain

$$\begin{aligned}
n_{tar} &= N \cdot r \cdot (1-p) = N \cdot \left( \sum_{i=0}^{R} (i+1)(1-p)p^i + (R+1)p^{R+1} \right)(1-p) \\
&= N \cdot \left( (1-p)(1+2p+3p^2+4p^3+\ldots+(R+1)p^R) + (R+1)p^R \right)(1-p) \\
&= N \cdot \begin{pmatrix} 1+2p+3p^2+4p^3+\ldots+(R+1)p^R \\ \quad -p-2p^2-3p^3-\quad \ldots \quad -Rp^R \end{pmatrix}(1-p) \\
&= N \cdot (1+p+p^2+\ldots+Rp^R)(1-p) \\
&= N \cdot (1-p^{R+1}).
\end{aligned}$$

Substitute (7) and (8) into (9), we have

$$\begin{aligned}
n_{fec_arq} &= K \cdot (1-p_{RS}) + \sum_{i=N-K+1}^{N} (N-i)\frac{K}{N} C_i^N {p_R}^i (1-p_R)^{N-i} \\
&= K \cdot \sum_{i=0}^{N-K} C_i^N {p_R}^i (1-p_R)^{N-i} + \sum_{i=N-K+1}^{N} (N-i)\frac{K}{N} C_i^N {p_R}^i (1-p_R)^{N-i} \\
&= K \cdot \sum_{j=N}^{K} C_{N-j}^N {p_R}^{N-j} (1-p_R)^j + \sum_{j=K-1}^{0} j\frac{K}{N} C_{N-j}^N {p_R}^{N-j} (1-p_R)^j \quad \text{where } N-i=j \Rightarrow i=N-j \\
&\le \sum_{j=N}^{K} j \cdot C_{N-j}^N {p_R}^{N-j} (1-p_R)^j + \sum_{j=K-1}^{0} j \cdot C_{N-j}^N {p_R}^{N-j} (1-p_R)^j \\
&= \sum_{j=0}^{N} j \cdot C_{N-j}^N {p_R}^{N-j} (1-p_R)^j \qquad \text{(Mean of the binomial distribution)} \\
&= N \cdot (1-p_R) \\
&= N \cdot (1-p^{R+1}) \\
&= n_{tar}.
\end{aligned}$$

Therefore, we conclude $n_{fec\_arq} \le n_{tar}$.