Supporting Video Applications with Header Error Protection in 802.11-based Networks

Su Yi, Yufeng Shan, Shivkumar Kalyanaraman and Babak Azimi-Sadjadi
Department of ECSE, Rensselaer Polytechnic Institute, TROY, NY 12180
Email: yis, shany@rpi.edu, shivkuma, babak@ecse.rpi.edu

Abstract

In a multi-hop network, the throughput is essentially important for real-time applications due to their high bit rate requirement. We investigate the performance of multimedia applications in wireless networks and the impact of various error control protocols. In particular, we propose a two stage error control scheme that improves the effective throughout of wireless networks. We apply error control to the packet header and packet load separately. The network intermediate nodes either use header FEC or header CRC checksum to successfully transport the packets from the source to the destination. Only at the destination, the error of the load is corrected. We compare the proposed schemes with 802.11 protocol and show that header error protection strategy can effectively increase the throughput, reduce the delay, and improve the video performance, via both theoretical analysis and simulation results.

I. Introduction

Unlike general data transmission which needs error free delivery at each protocol layer, multimedia data can tolerate bit errors in a received packet. Some applications, such as voice over IP or video streaming, have a higher data rate requirement than accuracy requirement. In addition to congestion related packet loss and delay, that is seen in wired packet switched networks, wireless networks have to deal with a time varying, error prone, physical channel that in many instances is also severely bandwidth constrained. As such, the methods needed for wireless multimedia applications are fundamentally different from wired ones. Protocol design, such as link layer error control may impact the performance of the network and these applications.

This work is supported by ARO under contract number DAAD19-00-1-0559.

One bit error in the link layer packet could cause the drop of the whole packet in the receiver side, even though the other bits of the packet are successfully received. This is acceptable for general data transmission, since one bit error in a file can make the whole file inaccessible. On the other hand, this may not be optimal for multimedia data transmission due to the loss tolerance of multimedia data. With partial data losses, the receiver may still decode the successfully transmitted part in a packet with desired visual quality. Therefore, at the receiver or the relays, instead of dropping the whole packet, a multimedia system can use the successfully transmitted bits in a received but corrupt packet, in order to reduce the bandwidth utilization.

Based on the above considerations, we found that error control in current 802.11 MAC protocol [1] is not efficient for supporting multimedia data transmission due to its bit error sensitivity. Therefore, in order to efficiently support multimedia data transmission we propose a new wireless link layer protocol. Even if the packet is received with some bit errors, the link layer still need to pass the packet to application layer. This approach is especially important in our proposed protocol, since we want to use the successfully received bits for multimedia applications. We call the proposed scheme HEP (Header Error Protection). A similar idea was previously used in ATM (Asynchronous Transfer Mode) which provides link-layer error correction for the packet header rather than for the entire packet [2]. A header error for both 802.11 MAC protocol and HEP based MAC protocol disrupts the transmission. Thus, the header information should be specially protected. Since the header is a small part of the packet the computational overhead of header error control is small. Error control techniques are used in this paper to protect the header information from being corrupt. Two categories of error control techniques are considered: Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ) [3], [4], [5], [6].

There are some arguments on whether error control should reside at the link layer or at the application layer [7]. We provide another option - do part of the error control at the local level and leave some work done at the application level. Specifically we propose several header error protection schemes and analyze their impact on the throughput of the wireless networks. Recently some approaches of allowing some errors in data packets were proposed in speech transmission [8], [9]. However, video applications differ from voice applications in the bandwidth requirement, and the delay tolerance because of the buffering at the receiver side of the video streaming. Moreover, the segment size of voice applications is much smaller than that of video applications, thus error has a higher impact on voice segment than on video segment.

There are also some interesting works on performance enhancement of video transmission over wireless networks. For instance, layered coding coupled with unequal error protection obtained by using different

retry limits at the link level has recently been shown to deliver interesting results [10], [11]. In [12], a modified version of the UDP Lite protocol [13] is proposed. It features a checksum for the packet header, and at the same time it provides an interface to forward all the information supplied by the CRC failures in link-layer frames to the application layer, to improve error location inside the packets. This protocol is combined with an unequal error protection scheme applied to fixed-size link-layer frames in a 3G wireless scenario. Other works suggest to limit the UDP Lite partial checksum to the packet header [14], focusing on schemes that add redundancy at the data link level, and allowing packets containing errors to be forwarded to the applications. However, the quality of the received multimedia streams is not measured. In this paper we would like to exploit the idea of protecting packet data and packet header differently in video transmissions.

The rest of the paper is organized as follows: header error protection strategies are introduced along with their throughput analysis in Section II; in Section III, we show our simulation results and followed by the conclusions in Section IV.

II. Header Error Protection and Performance Evaluation

In wireless network the throughput is a key characteristic, especially for real-time applications, which require high bandwidth utilization to satisfy end users. Consider an ad hoc network with n nodes randomly located in a domain of area one square meter. It was shown in [15] that under a Protocol Model for interference, such a network could provide a per node throughput of $O(\frac{1}{\sqrt{n \log n}})$ bits/sec. In this case, the total end-to-end capacity of the entire network is $O(\sqrt{\frac{n}{\log n}})$. This result indicates a vanishing throughput performance as the network scales.

The *effective throughput* we discuss in this paper is defined as the the fraction of channel bandwidth that is used to successfully transmit packets if every node is transmitting in full utilization of bandwidth. Also this effective throughput is under the impact of packet[1] error control. We will consider three packet error control schemes. We start with the ARQ scheme in the current Wireless LAN MAC layer protocol IEEE 802.11. After that, we propose two kinds of header error protection scheme: *header CRC* and *header FEC*. These two schemes are compared with the original ARQ strategy used in 802.11 protocol.

It was shown that under a Protocol Model for interference, if there are n nodes randomly placed in a network domain, the average hop number h is assumed to be $\sqrt{\frac{n}{\log n}}$; each node in the network can

[1]In this paper we talk about link layer error control, yet we still use the term *packet* instead of *frame* for general use, and to differentiate with the term *frame* in video transmission.

transmit at an average rate of $\frac{c}{\sqrt{n \log n}}$ bits/sec, where c is a constant. This paper uses the mathematical approximations with these average values. The critical idea is to consider random topologies and traffic patterns, to allow the number of nodes to go to infinity, and to compute the performance asymptotically. In this manner, statistical averaging is introduced. All results will only hold with high probability, i.e., with probability approaching unity as the number of nodes approaches infinity. However this is a small concession to make, given the wealth of results that can be obtained in this manner.

A. Error Models for Link Layer

Measurement and analysis of the error characteristics of a wireless network is itself a research topic. Even worse, the characteristics are very environment dependent. For this reason, varieties of experimental data is presented for wireless LANs [16]. We focus on the network performance rather than the modeling of the physical layer error properties. So we only think about a virtual bit channel, not a physical communication channel.

In this paper we use the Binary Symmetric Channel (BSC) model with error probability p and a binary Markov channel model as our channel error models. For a BSC error model, data are transmitted on a channel with the error probability p. This is a memoryless model where the noise bits are produced by a sequence of independent trials. Each has the same probability $1 - p$ of producing a correct bit and probability p of producing a bit error. p is then the bit error rate (BER) for the wireless link.

Binary Markov channel is the first order binary Markov channel model (called Gilbert model [17], [18] for packet transmission). It is shown through analysis and simulation that a first-order Markov process is a good approximation for fading channels [19], [20]. The model is described by the transition matrix

$$\begin{bmatrix} 1 - p_{01} & p_{01} \\ p_{10} & 1 - p_{10} \end{bmatrix}$$

where p_{01} (p_{10}) is the probability that the transmission of current bit is unsuccessful (successful), given that the previous transmission was successful (unsuccessful). Note that $\frac{1}{p_{10}}$ represents the average length of a burst of errors, and the average BER is given by $\frac{p_{01}}{p_{01}+p_{10}}$.

B. Packet CRC in 802.11

In IEEE 802.11, the ARQ is a stop-and-wait ARQ with a positive ACK after each packet. The CRC checksum protects the whole packet. Usually there is a limit on the number of times that WLAN cards retransmit a packet (e.g., 4 times). Simply consider the single hop packet error probability, defined as

P_{e_1} for this packet CRC scheme. For the BSC, the errors are independent, so

$$P_{e_1} = \sum_{i=1}^{q} (1-p)^{q-i} p^i \begin{pmatrix} q \\ i \end{pmatrix} \tag{1}$$

where q is the packet length (in bits).

Under our assumptions, there are n nodes in the network, the aggregate throughput without considering packet dropping is $c\sqrt{\frac{n}{\log n}}$. First we assume there is no limit on the number of retransmissions. Given the probability of error P_{e_1}, the average number of retransmissions for a single hop has a geometric distribution with successful probability of $1 - P_{e_1}$. Thus the probability of number of retransmissions (excluding the first transmission) in one hop is:

$$\mathbf{P}\{ret = i\} = P_{e_1}^i (1 - P_{e_1}) \tag{2}$$

If a flow only has one hop distance and the bandwidth is W, then the effective throughput of this flow is

$$F(h=1) = \sum_{i=1}^{\infty} \frac{W}{i} \mathbf{P}\{ret = i-1\} = \sum_{i=1}^{\infty} \frac{W}{i} P_{e_1}^{i-1} (1 - P_{e_1}) \tag{3}$$

Note $\frac{d}{da}(\sum_{i=1}^{\infty} \frac{a^i}{i}) = \sum_{i=1}^{\infty} a^{i-1} = \frac{1}{1-a}$ when $|a| < 1$. So $\sum_{i=1}^{\infty} \frac{a^i}{i} = \int \frac{1}{1-a} = c_0 - \ln(1-a)$. Let $a = 0$ to solve the constant value we get $c_0 = 0$. Then we have

$$F(h=1) = -\frac{(1 - P_{e_1}) \ln(1 - P_{e_1})}{P_{e_1}} W \tag{4}$$

For multi-hop networks, since the error statistics for each hop is independent, the retransmission for different wireless links are independent. Now suppose a flow has experienced h hops, consuming bandwidth $W_1, W_2, ..., W_h$ of each link respectively. Then the aggregate effective throughput of this flow is

$$\begin{aligned} F(h) &= \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{\infty} \cdots \sum_{i_h=1}^{\infty} (\frac{W_1}{i_1} + \frac{W_2}{i_2} + \cdots + \frac{W_h}{i_h}) \\ &\quad \cdot \mathbf{P}\{ret = i_1 - 1\} \mathbf{P}\{ret = i_2 - 1\} \cdots \mathbf{P}\{ret = i_h - 1\} \\ &= \mathbf{E}[\frac{W_1}{i_1}] + \mathbf{E}[\frac{W_2}{i_2}] + \cdots + \mathbf{E}[\frac{W_h}{i_h}] \\ &= -\frac{(1 - P_{e_1}) \ln(1 - P_{e_1})}{P_{e_1}} (W_1 + W_2 + \cdots + W_h) \end{aligned} \tag{5}$$

Since all the $W_1, W_2, ..., W_h$ add up to the network aggregate throughput $c\sqrt{\frac{n}{\log n}}$, summing up all the flows in the network adds up to the total aggregate effective throughput of the 802.11 protocol. Therefore, we have

$$A_1 = -c\sqrt{\frac{n}{\log n}} \frac{(1 - P_{e_1}) \ln(1 - P_{e_1})}{P_{e_1}} \tag{6}$$

Note in Eqn(5), we set the retry limit to ∞. It is easy to see from the expression that $F(h)$ is an increasing function of the retry limit. Thus, when retry limit is less than ∞, the aggregate throughput A_1 will be less than the result give in Eqn(6). In fact, the effect of a positive retry limit has diminishing return. For typical BER it is easy to show that the 4 retry limit in 802.11 can be approximated with by Eqn(6). We use the results given by the infinity retry limit for the standard 802.11 protocol and header CRC.

C. Header CRC

Header CRC aims to protect the header, not the whole packet. There is a retransmission for the packet if error detected in header part. The main purpose of protecting header is the need to carry the correct destination address for IP forwarding, and source address for end-to-end ACK. A CRC is needed for detecting errors in this information. The probability that any error detected in a header is:

$$P_{e_2} = \sum_{i=1}^{k+r} (1-p)^{k+r-i} p^i \begin{pmatrix} k+r \\ i \end{pmatrix} \tag{7}$$

where k is the header size, and r is the CRC bits.

In a similar form with previously introduced packet CRC, the aggregate effective throughput of networks is:

$$A_2 = -c\sqrt{\frac{n}{\log n}} \frac{(1-P_{e_2})\ln(1-P_{e_2})}{P_{e_2}} \tag{8}$$

Note that the factor part $-\frac{(1-P_{e_2})\ln(1-P_{e_2})}{P_{e_2}}$ is a monotone decreasing function of P_{e_2}. This factor decreases from 1 to 0 as P_{e_2} increases from 0 to 1. This is consistent with heuristic expectations, because one expects the throughput to increase when packet error probability decreases.

D. Header Error Control Coding

In header error control coding, the network nodes use an error control coding technique to transmit the header information without error. Therefore, the network always has the correct address of the destination. In this scheme, the network might deliver the packet through with error in payload.

We also call this scheme *header FEC*, since FEC is added to the header. In header FEC, a retransmission is issued when the redundancy fails to correct the errors in header. Therefore it is in fact a hybrid ARQ for only protecting header part. BCH codes are well known codes for binary data transmission, especially good for large block codes [4]. m protection bits are added to each header for error correction. For a t-error-correcting linear code, it is capable of correcting a total of 2^m error patterns, including those with

t or fewer errors. So the probability that the decoder commits an erroneous decoding in one packet is upper bounded by:

$$P_{e_3} \leq \sum_{i=t+1}^{k+m} (1-p)^{k+m-i} p^i \begin{pmatrix} k+m \\ i \end{pmatrix} \tag{9}$$

A packet is likely to fail to reach the destination unless it succeeds during the transmission at each hop. Given the probability that a packet will be dropped in one-hop transmission P_{e_3}, it is easy to get the aggregate throughput of the network using header error coding:

$$A_3 = c\sqrt{\frac{n}{\log n}}(1-P_{e_3})^h = c\sqrt{\frac{n}{\log n}}(1-P_{e_3})^{\sqrt{\frac{n}{\log n}}} \tag{10}$$

We put the FEC protection bits to the tail of the packet, because errors tend to be in burst. If we let the redundant bits be faraway from the header, the header and the protection bits are less likely to be corrupt at the same time.

The efficiency of coding requires the information message to be as small as possible. On the other hand, the more redundancy bits added, the more reliable the transmission would be. The question is how many bytes exactly we would encode. Considering IP header is 20 bytes, we now suppose 30 bytes are to be protected by error detection or correction, since there are important information in headers from other layers as well. This header protection configuration can be adapted to different applications. For binary BCH codes, we choose codes that satisfy block length of $k+m = 255$ bits, $k = 247$ bits, and $t = 1$ bit. This combination is the closest to 30 bytes (240 bits) header. We then use 8 error correction bits to correct 1 bit error for 247 bits. So 1 byte extra can protect 30 bytes of header. Substituting these numbers in (9), we have $P_{e_3} = 2.9884 \times 10^{-4}$ and 3.0578×10^{-6} with $p = 10^{-4}$ and 10^{-5}, respectively. That means to protect the header that is no longer than 30 bytes, 1 byte is enough. Also since P_{e_3} is so small, A_3 in Eqn(10) could be seen as asymptotically approaching to $c\sqrt{\frac{n}{\log n}}$ when $n \to \infty$.

E. Comparison of the Effective Throughput of These Schemes

By comparing P_{e_1} with P_{e_2} and P_{e_3}, it is quite clear the latter two have much lower values thus the proposed HEP schemes have the advantage in terms of the effective throughput. The two ARQ schemes - packet CRC and header CRC - also have the same form of throughput expressions, with different factors that depend on the packet error rate. For a clear illustration, we use MATLAB to generate numerical simulation plots of the throughput with an increasing number of nodes for these four different schemes using the throughput expressions derived above. Factor c in the y-axis is the same as that in the above equations.

The parameters used in these plots are: $p = 5 \times 10^{-5}$, payload $q = 500 \times 8$ bits, header $k = 240$ bits, error correction bits $m = 8$, CRC bits $r = 8$, and error correcting capability $t = 1$ bit. Some intermediary results are: $P_{e_1} = 0.1910, P_{e_2} = 1.5217 \times 10^{-4}, P_{e_3} = 7.5634 \times 10^{-5}$. P_{e_3} used here is the upper bound number, i.e., the worst case scenario.

Fig.1 show the average per-node throughput as a function of the number of nodes in the network. Our analytical results are valid for large networks. We choose the amount of nodes from 50 to 100 in this plot also to meet the practical needs. Curves for header CRC and header FEC almost lap over each other. The difference between the performance of these two HEP schemes and 802.11 indicates that the gap between the not-so-good performance of the current protocols and the theoretical results can be reduced using header error control. The curves for header FEC and header CRC demonstrate a better scale property than the packet CRC scheme used in 802.11. This result may help in the design of different protocol stacks according to different requirements. For applications having high requirements for data rate and less requirements for accuracy of data, the header error control is especially helpful. The choice between header error coding and header CRC depends on questions like whether or not the coding and decoding energy is a factor, if the reverse link is desired, etc. Other concerns may affect the choice as well, which include ARQ is better for handling burst errors and header error coding can be adaptive to the link error environment (e.g., if the link error rate increases, the protection bits can be added to correct more errors with little cost).

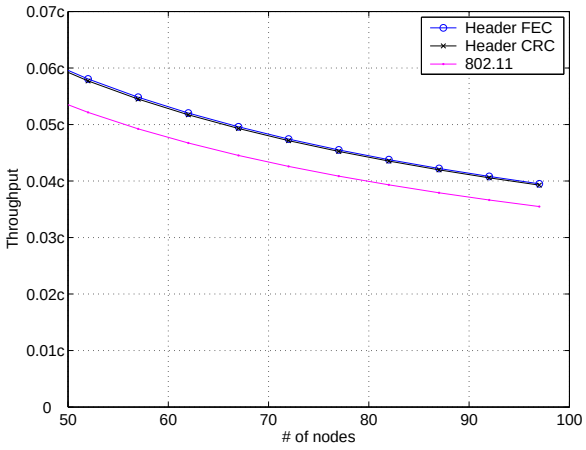


Fig. 1. Per-node Throughput as a Function of n for Different Schemes

III. SIMULATIONS

In this section, we evaluate our proposed header error protection schemes by multimedia simulations. The network simulator we use is *ns2* plus wireless extension [21]. We build new protocol models based on our proposed schemes and integrate them into *ns2*. We use the default packet retry limits - 4 for long packets and 7 for short packets - in both 802.11 and our proposed HEP protocols. The traffic type we use is Constant Bit Rate (CBR) traffic over User Datagram Protocol (UDP). The packet size used in all simulations is 500 bytes. In addition, all of our simulations use 2Mbps radio.

A. Ns2 Simulations

In order to model a scenario that is closer to reality, we simulate our protocol and 802.11 protocol on a random network. *Random network* is defined in section II. Nodes are placed uniformly at random in a square domain, and the traffic pattern is random in this network. In *ns2*, the default setting of antenna parameters results in an effective transmission range of 250 meters. The average node density is set to 75 nodes per square kilometer to guarantee the connectivity of the network. In order to get the capacity of the network, we let each node send packets to a randomly chosen destination. The CBR rate of the traffic is chosen in order to place the network in a saturation state. In this state there is some slight packet loss and if CBR rate is increased the network aggregate throughput will not increase statistically.

We simulate random networks scaled from 50 nodes to 100 nodes under 802.11 protocol, header CRC protocol, and header FEC protocol with a Markov channel model. Fig.2 gives an example of random network node placement with 100 nodes. The parameters for Markov are $p_{01} = 2.5 \times 10^{-5}, p_{10} = 0.5$, which yield an average channel BER 5×10^{-5}. The duration of each simulation is 2 minutes and the result is averaged upon 200 runs for different node distributions. The average per-node throughput is shown in Fig.3. The simulation results reflect statistically significant analysis based on a 95% confidence interval. If we compare Fig.3 with Fig.1, we see they share the same decreasing trend. The sharper decrease in the simulation results indicate the inefficiency of the MAC scheduling. When network scales, the distributed MAC protocol can hardly give an optimal solution to achieve the theoretical capacity, which was observed in [22]. Nevertheless when nodes are around 100, the throughput improvement by using header CRC or header FEC upon 802.11 is about 18%.

B. Video Simulations

1) A Single-hop Scenario: We use video experiments to show the different video quality of three schemes, and the difference in throughput capacity each scheme can achieve. We start our video simulation

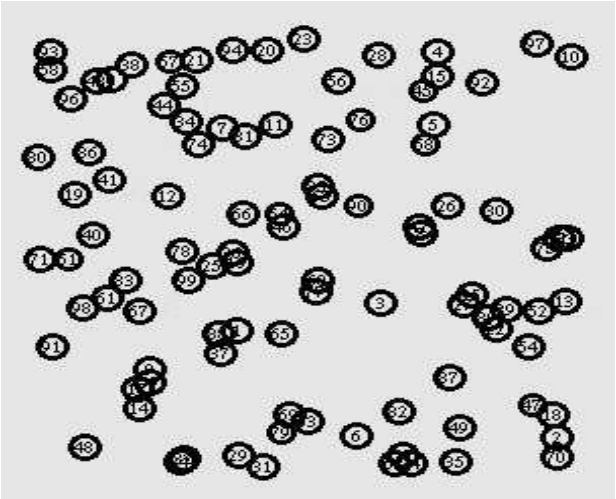


Fig. 2. Random Network Model Where Nodes Randomly Spread around a Square

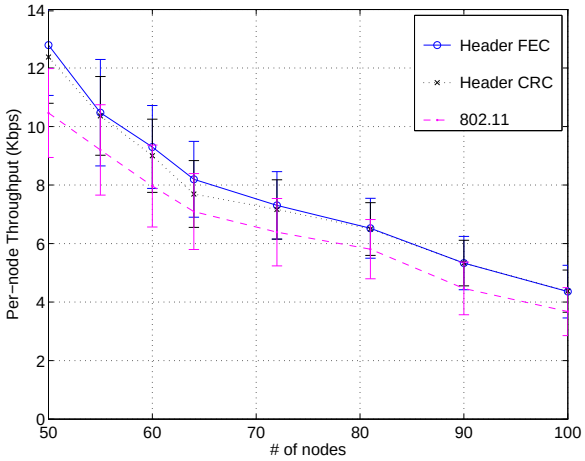


Fig. 3. Simulation Results on Per-node Throughput of Random Networks as a Function of n for Different Schemes. The error bars show the data within 95% confidence interval around the mean value, the pointed data.

with a network with only 2 nodes. We put 2 nodes 200 meters apart. Due to the significant overhead added by the exchange of RTS/CTS/ACK, when packet size is 500 bytes, the maximum throughput achievable for the 2 nodes under an error-free wireless environment is slightly above 1Mbps.

We test our video simulation using a H.263+ coded bitstream. “Foreman” video sequence is used with 300 frames length, QCIF (Quarter Common Intermediate Format, Quarter CIF) format [23]. Given the maximum throughput achievable for this single-hop scenario, we use multimedia streaming experiments to evaluate the performance of the three schemes. The encoded bit stream is divided into 500 byte packets.

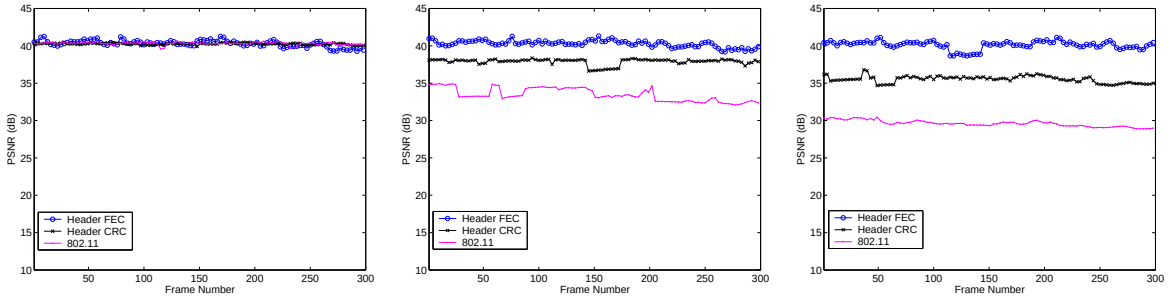
These packets are transmitted evenly spaced over approximately 4 milliseconds, thus the data transmission rate is 1Mbps. In this test, different channel conditions are used. First scenario, $p_{01} = 0.5 \times 10^{-5}, p_{10} = 0.5$, then $p_{01} = 1.25 \times 10^{-5}, p_{10} = 0.5$; and last $p_{01} = 2.5 \times 10^{-5}, p_{10} = 0.5$. The average BER for these scenarios are 10^{-5}, 2.5×10^{-5}, and 5×10^{-5}, respectively. The simulation takes 50 runs in each scenario for all the schemes. Note in order to correct the residual bit errors at the receiver for the header CRC and header FEC strategies, we use a rate 1013/1023 BCH code at the application layer. This high rate code has a simple generator polynomial and has little overhead.

The quality of the three schemes are easy to differentiate visually. Header FEC performs best, without obvious discernable artifacts, especially in the first two scenarios. Header CRC is not as good but quite acceptable. 802.11 comes the last and gets much worse with bad channel condition. Fig.4 shows some sample frames of the video simulation for the last scenario. The averaged PSNR (Peak Signal to Noise Ratio, a commonly used picture quality measurement) charts are shown in Fig.5. The reason that 802.11 performs poorly as channel condition gets worse is that it has packet losses due to the limited retransmissions. Packet loss is unlikely to be recovered by high rate FEC. In addition, packet loss affects much larger area in a video frame than bit errors, which makes it unaffordable.

(a) Header FEC (b) Header CRC (c) 802.11

Fig. 4. Some Sample Video Clips for Different Schemes

2) A Multi-hop Chain Scenario: The advantage of using header error protection is more obvious in a multi-hop network, since retransmissions increase the traffic load and limit the throughput. In the next set of simulations we intend to find out the effective throughput of multi-hop networks under the three schemes. We use a single traffic chain model to avoid the effect of the interference by other traffics. There are n nodes placed in a straight line, and each neighboring nodes are separated by 200 meters, shown in Fig.6. The video traffic is sent from the first node to the last node, traveling through all the intermediate nodes. Parameters of the Markov error model are: $p_{01} = 2.5 \times 10^{-5}, p_{10} = 0.5$. Fig.7 illustrates simulation



(a) Average BER=10^{-5} (b) Average BER=2.5×10^{-5} (c) Average BER=5×10^{-5}

Fig. 5. PSNR vs. Frame Number for Different Schemes

results of the maximum throughput for a single chain. Let *chain length* be the number of nodes in a chain. Each curve represents the flow throughput of different protocols when the chain length n increases from 5 to 10. The curve for 802.11 throughput performance is basically consistent with that in [24] (their throughput is a little bit higher since they do not have error model in *ns2*).

Fig. 6. A Single Chain with Multi-hops from Sender S to Receiver R

Header CRC performs not as good as header FEC, because headers cannot be recovered by FEC, there are still some packet drops due to too many retries. Even though header CRC and header FEC consume some extra bandwidth for the application FEC overhead, the effective throughput is higher than that of 802.11. Simulation results show that there is some potential in the throughput improvement for the header error protection schemes, especially when network gets large and hop number increases.

End-to-end latency is also an important issue for video applications. We also evaluate the delay performance for this chain multi-hop network. Both the saturated and the unsaturated network performance is evaluated. In the saturation state, nodes are more likely to cope with other transmitting neighbors. For this reason, there will be more packet drops at the interface queue than for the unsaturated network. In the unsaturated state, the traffic load is light, so the collisions are rare. The results from Fig.8 and Fig.9 show that the average end-to-end latency of the saturated network is in fact slightly less than that of the unsaturated network, because more packets are dropped via collision. More discussion about the saturation is presented via the following simulations. The header FEC strategy is especially superior than

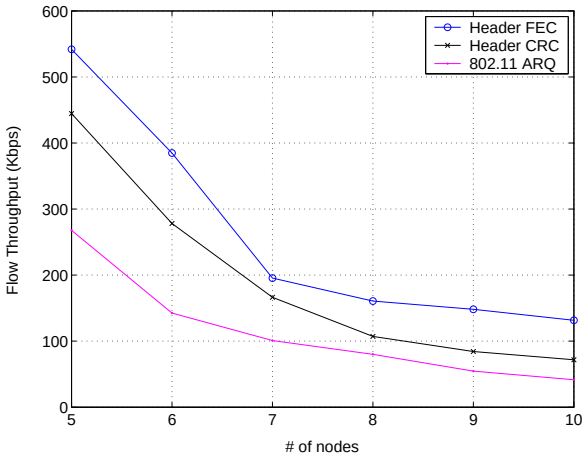


Fig. 7. Throughput vs. Chain Length n for Different Schemes

the other two schemes in the unsaturated network, which applies to most of the real cases. The fluctuation of the curve reflects the even and odd number of nodes in the chain. The schedule of the MAC protocol is sub-optimal, and the fairness is not well addressed.

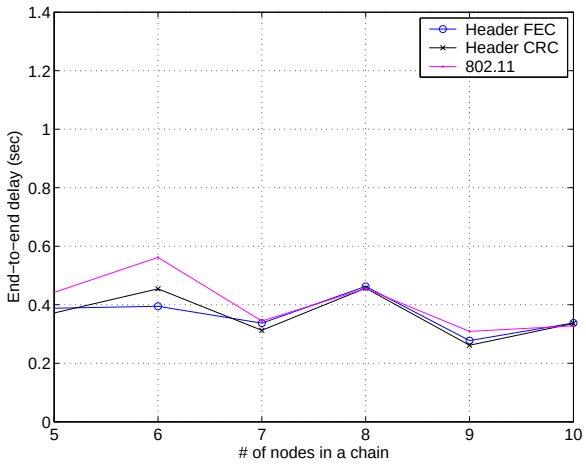


Fig. 8. End-to-end Latency in a Saturated Network for Different Schemes

In the simulated unsaturated network, we study the trace file of our simulations and find that the packet drops, which may happen at any node, come from two causes: the interface queue overflow and the excess of the retry limit. Both causes are directly related to the link layer retransmission. More retransmissions lead to occupied interface queue. They also cause exceeding the retry limit of data packet or RTS message. Fig.10 shows the packet loss rate of the chain network for three schemes. The packet

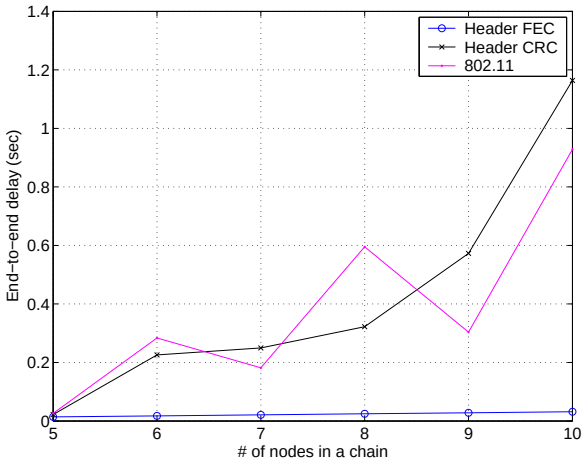


Fig. 9. End-to-end Latency in an Unsaturated Network for Different Schemes

loss rates are always 0% for the header FEC in this single chain network, which further supports the above statements. The average PSNR for the video streaming versus the chain length is shown in Fig.11. The transmission rate is 40bps, which corresponds to the unsaturated network condition. This low data rate results in a relatively low source coding quality. As shown in Fig.11, the average PSNRs are lower than those in previous single hop network that uses a much higher data rate. It also shows that header FEC has a static performance on average PSNR with chain length, whereas performance of both header CRC and 802.11 decrease sharply after chain length reaches 8. This indicates that multi-hop scenario exaggerates the video quality discrepancies between those three schemes.

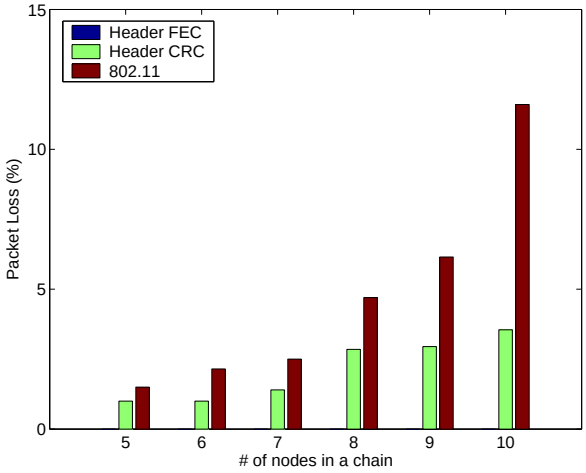


Fig. 10. Packet Loss vs. Chain Length n for Different Schemes

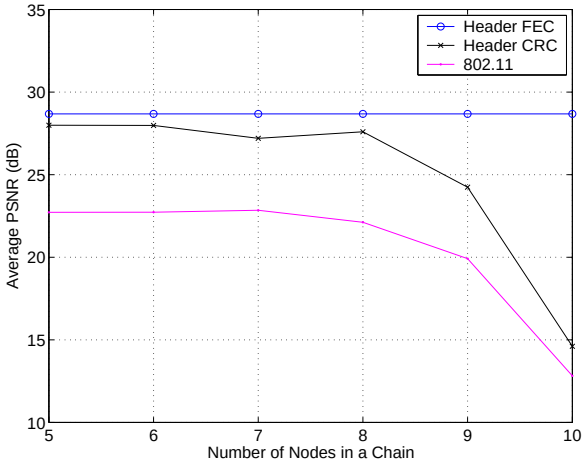


Fig. 11. Average PSNR vs. Chain Length n for Different Schemes

3) A Multi-hop Chain Topology with Cross Traffic: In the previous chain topology simulation, the network does not have much load caused by contention, since only one single traffic is carried from the source to destination. The only contention is between the nearby relay nodes. In order to get more results for the performance of a multi-hop network, we place more ad hoc nodes to set up a linear topology with cross traffic. In this grid, shown in Fig.12, the main traffic is a CBR carried from node S to node R. This node placement is the same as in the chain topology. Besides this main traffic, there are cross traffic from node S_1 to R_1, node S_2 to R_2, ..., and node S_n to R_n, with n as the main chain length. These cross traffic also needs the nodes in the main chain as relays. The distance between each horizontally and vertically adjacent nodes is 200m. Parameters of the Markov error model are: $p_{01} = 1.25 \times 10^{-5}, p_{10} = 0.5$.

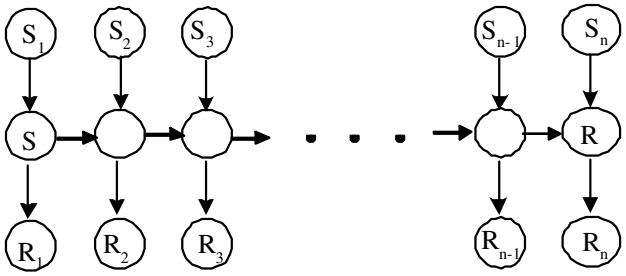


Fig. 12. A Chain Topology with Cross Traffic

In this chain topology with cross traffic, the source of the main chain together with the relays of this chain have to compete with the nodes transmitting cross traffic. The RTS/CTS/DATA/ACK mechanism requires nodes near the sender or receiver to keep silent. Since the transmission range in our simulation

set up is 250m, there will be much contention around each transmission pair. If a sender experiences a collision when sending RTS, it will choose an exponential random backoff time and retransmit after the backoff. The retransmission will continue if it fails, until it reaches the retry limit. It is obvious that the number of retries has impact on the end-to-end delay. It also increases the possibility of the collision for the surrounding nodes, which further increases the delay of each transmission. The proposed header protection strategy can largely reduce the number of retransmissions, thus it has the potential to reduce the end-to-end delay, which is important for a real time application.

Fig.13 gives the result for the end-to-end throughput performance of the chain topology with cross traffic. The video is in 40kbps, with the channel error rate 2.5×10^{-5}. The cross traffic is also in a 40kbps rate. It plots the flow throughput of the main chain (middle chain) traffic versus the chain length n. Fig.14 shows the delay performance of the main chain for the same setting. In fact, the end-to-end throughput and end-to-end delay performance are closely related. The higher the throughput, the shorter the delay. This is because that the longer delay indicates more retransmission, which leads to more collision and thus more possible packet drops. It is somewhat misleading that the delay performance in this network with heavy traffic is better than that in the previous single chain network (Fig.9). The reason of this is that the packet loss rate, which is shown in Fig.15, is much higher in this multi-flow network than in the single-flow network. Some packets are dropped instead of surviving after a long time delay. Hence the delay performance itself does not necessarily reflect the video performance.

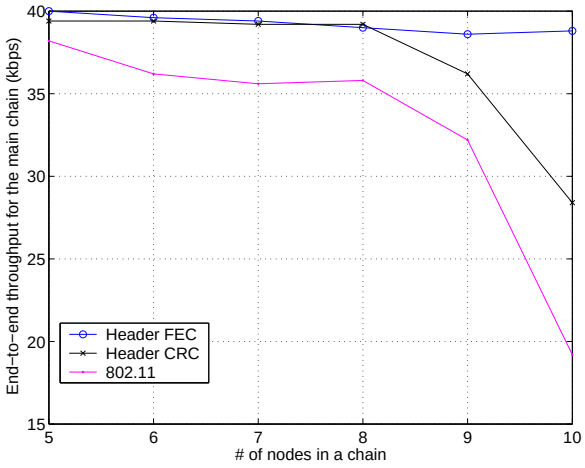


Fig. 13. End-to-end Throughput vs. Main Chain Length n for Different Schemes

It is worth mentioning that the sending rate for the middle chain source, 40kbps, acts like a threshold of the optimal sending rate for this grid topology with main chain length 10. If the sending rate is

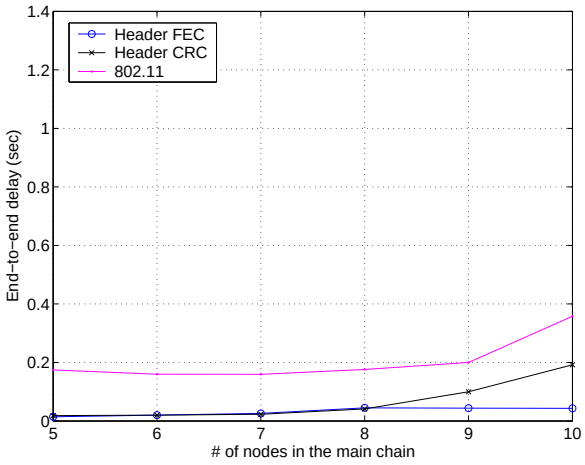


Fig. 14. End-to-end Latency vs. Main Chain Length n for Different Schemes

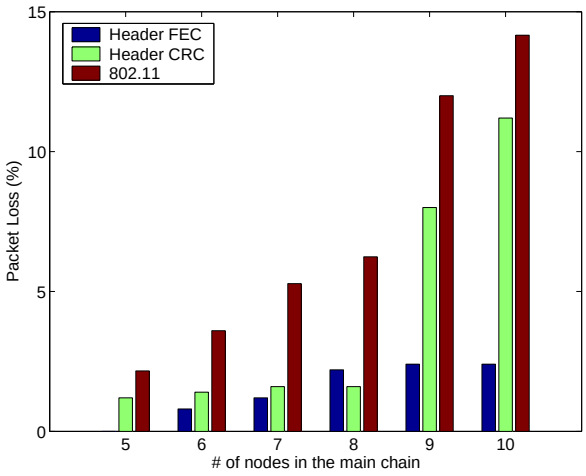


Fig. 15. Packet Loss vs. Main Chain Length n for Different Schemes

increased, the end-to-end throughput will drop. We also regard this threshold as the saturation point. This point depends on the node placement, and the traffic pattern for the whole network. Below this point, the network is in an unsaturate state, and the throughput will be close to the sending rate. If the sending rate exceeds the saturation point, the throughput will drop, sometimes very sharply. This drop of throughput implies the MAC scheduling inefficiency. That is also to say, 802.11 MAC cannot discover the optimum schedule of transmissions on its own. Each node in a network experiences different degree of competition. For example, nodes at the edges of the grid have fewer competitors than those in the middle of the grid. So some bandwidth is wasted by transmitting packets that are eventually dropped at

some nodes with higher degree of contention.

Another concern is that the interference range is always larger than the transmission range. Thus a node trying to transmit may not have received RTS or CTS that can inform how long it has to wait, yet it may fail because of the interference. In this case, it chooses a random backoff time and sends again. The next try might fail again because it does not has the information about the interfering nodes. Due to the limited retries and limited interface queue buffer length, too many retransmissions, either from contention or packet CRC check, can make things much worse than expected, especially near the saturation point. That is why the original 802.11 scheme has a much lower performance than the header protection schemes. In any case, the saturation of the network should be avoided.

Fig.16 show the average PSNR in dB as a function of number of hops, with different error control approaches used. Due to the introduction of the cross traffic, the degradation of the performance as the number of nodes increases is more evident than the single flow scenario for all three schemes. In this case, HEP, especially header FEC can effectively improve the PSNR values from the 802.11 scheme, thus prevent the video quality from degrading rapidly.

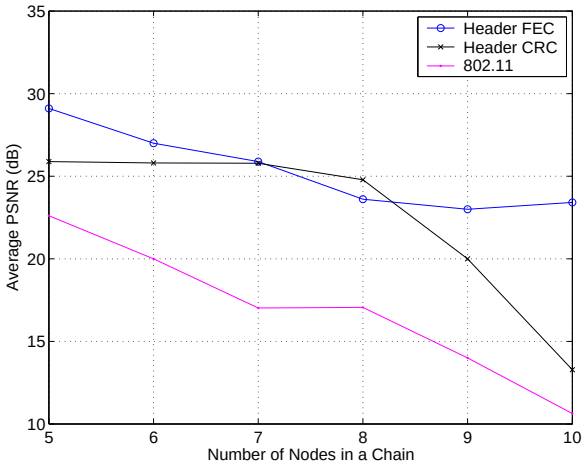


Fig. 16. Average PSNR vs. Main Chain Length n for Different Schemes

IV. Conclusions

This paper proposes two header error protection schemes, header CRC and header FEC, in order to give a solution to improve the performance of multimedia transmissions. Both header CRC and header FEC only need insignificant changes at the packet header if 802.11 is kept as the MAC protocol. Network simulation results show that under a random network scenario header error protection takes advantage of

FEC or ARQ to reduce the number of dropped packets at relaying nodes, thus can improve the throughput of the network. We also examine their video streaming performances together with 802.11 protocol under a single hop network, a multi-hop chain and a cross traffic abundant multi-hop chain network. They present better qualities than 802.11 does in terms of the visual effects observed by experimenters and the PSNR results. Packet losses induced by bit error checking not only impair the video quality but also diminish the maximum throughput a network can achieve.

Since the link layer does not perform any error correction or detection for the whole packet, the payload error at the destination may be higher than the acceptable limit. Therefore, we propose to use end-to-end error control coding for the application layer, wherever it is needed. Application layer FEC is needed not only because of the channel errors, but also because of the packet losses caused by congestion. This is one of the reasons we propose not to do local error protection for the whole packet. Whether to do end-to-end error control coding or not, and how efficient the codes should be depend on the requirement of the application.

A cross layer approach that integrates application and link layer should be considered, such as how to protect data information according to the priority of classes of the data, and choose FEC packets adaptively to the application requirement, etc. We leave these investigations to our future research work.

REFERENCES

[1] IEEE Computer Society LAN MAN Standard Committee, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York: IEEE Std. 802.11-1999, 1999.

[2] U.Black, *ATM Foundation for Broadband Networks, vol. 1, 2nd Ed.* Upper Saddle River, NJ: Prentice Hall PTR, 1999.

[3] S. Lin, D. J. C. Jr., and M. J. Miller, "Automatic-repeat-request error control schemes," *IEEE Communications Magazine*, vol. 22, no. 12, pp. 5–16, December 1984.

[4] S. Lin and D. J. Costello, *Error Control Coding, 2nd Ed.* Pearson Education, 2004.

[5] R. Fantacci and M. Scardi, "Performance evaluation of preemptive polling schemes and arq techniques for indoor wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 248–257, May 1996.

[6] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Upper Saddle River, NJ: Prentice Hall, 1995.

[7] D. A. Eckhardt and P. Steenkiste, "Improving wireless lan performance via adaptive local error control," in *Proc. ICNP'98*, 1998.

[8] I. Chakeres, H. Dong, E. M. Belding-Royer, A. Gersho, and J. D. Gibson, "Allowing errors in speech over wireless LANs," in *Proceedings of the 4th Workshop on Applications and Services in Wireless Networks (ASWN)*, Boston, MA, Aug. 2004.

[9] A. Servetti and J. C. D. Martin, "Link-level unequal error detection for speech transmission over 802.11 wireless networks," in *Special Workshop in Maui (SWIM), Lectures by Masters in Speech Processing*, Maui, HI, Jan. 2004.

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

[11] S. Krishnamachari, M. V. D. Schaar, S. Choi, and X. Xu, "Video streaming over wireless LANs: A cross-layer approach," in *Proc. Packet Video Workshop*, Nantes, France, Apr. 2003.

[12] H. Zheng and J. Boyce, "An improved udp protocol for video transmission over internet-to-wireless networks," *IEEE Trans. on Multimedia*, vol. 3, no. 3, pp. 356–365, September 2001.

[13] L.-A. Larzon, M. Degermark, S. Pink, L.-E. Jonsson, and G. Fairhurst, "The UDP-lite protocol," draft-ietf-tsvwg-udp-lite-02.txt, 2003.

[14] G. Ding, H. Ghafoor, and B. Bhargava, "Error resilient video transmission over wireless networks," in *Proc. IEEE Workshop on Software Technologies for Future Embedded Systems*, Hakodate, Hokkaido, Japan, May 2003, pp. 31–34.

[15] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. IT-46, no. 2, pp. 388–404, March 2000.

[16] D. Eckhardt and P. Steenkiste, "Measurement and analysis of the error characteristics of an in building wireless network," in *Proceedings of the ACM SIGCOMM '96 Symposium on Communications Architectures and Protocols*, August 1996, pp. 243–254.

[17] E.N.Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1265, September 1960.

[18] E.O.Elliot, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977–1997, September 1963.

[19] M. Zorzi, R. R. Rao, and L. B. Milstein, "On the accuracy of a first-order markov model for data transmission on fading channels," in *Proceedings of ICUPC*, Tokyo, Japan, Nov. 1995, pp. 211–215.

[20] ——, "ARQ error control for fading mobile radio channels," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 2, pp. 445–455, May 1997.

[21] "The network simulator - ns-2," http://www.isi.edu/nsnam/ns/.

[22] P. Gupta, R. Gray, and P. R. Kumar, "An experimental scaling law for ad hoc networks," http://decision.csl.uiuc.edu/ prkumar/html-files/ps-files/exp.pdf, 2001.

[23] A. Netravali and B. Haskell, *Digital Pictures: Representation, Compression, and Standards (2nd Ed)*. New York, NY: Plenum Press, 1995.

[24] J. Li, C. Blake, D. D. Couto, H. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *ACM MobiCom'01*, July 2001.