# Header Error Protection for Multimedia Data Transmission in Wireless Ad Hoc Networks

Su Yi*, Yufeng Shan*, Shivkumar Kalyanaraman* and Babak Azimi-Sadjadi*†
* Department of ECSE, Rensselaer Polytechnic Institute, TROY, NY 12180
† Institute for System Research, University of Maryland, College Park, MD 20742
Email: yis, shany@rpi.edu, shivkuma, babak@ecse.rpi.edu

**Abstract**

Multimedia applications have different Quality of Service (QoS) requirements. In a multi-hop network, the throughput is essentially important for real-time applications due to their high bit rate requirement. In the wireless networks, error comes from fading, noise, or interference. Link layer error control will have impact on the end-to-end throughput. In this paper 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 and the video performance, via both theoretical analysis and simulation results.

**Index Terms**

Ad hoc network, Header Error Protection, FEC, ARQ, video streaming.

# Header Error Protection for Multimedia Data Transmission in Wireless Ad Hoc Networks

Su Yi*, Yufeng Shan*, Shivkumar Kalyanaraman* and Babak Azimi-Sadjadi*†
* Department of ECSE, Rensselaer Polytechnic Institute, TROY, NY 12180
† Institute for System Research, University of Maryland, College Park, MD 20742

***Abstract*— Multimedia applications have different Quality of Service (QoS) requirements. In a multi-hop network, the throughput is essentially important for real-time applications due to their high bit rate requirement. In the wireless networks, error comes from fading, noise, or interference. Link layer error control will have impact on the end-to-end throughput. In this paper 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 and 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.

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 [3] 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 [10]. 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) [8].

There are some arguments on whether error control should reside at the link layer or at the application layer [4]. 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 [2], but to our knowledge no theoretical result was given and no work has been done in the area of video streaming transmission.

This 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 Effective Throughput Analysis

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 by Gupta and Kumar in [6] 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 [6] 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 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.

### A. Error Models for Link Layer

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.

Binary Markov channel is the first order binary Markov channel model (called Gilbert model [5] for packet transmission). It is shown through analysis and simulation that a first-order Markov process is a good approximation for fading channels [11]. 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 (bit error rate) 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). To be complete, first we consider the extreme case when the retransmission limit is 0, that is to say, there is no retransmission at all, the packet gets dropped whenever a bit error occurs in a packet. First 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 $c\sqrt{\frac{n}{\log n}}$. Since there is no retransmission, a packet is likely to fail to reach the destination unless it succeeds during the transmission at each hop. Thus the aggregate throughput of this extreme scheme is

$$A_0 = c\sqrt{\frac{n}{\log n}}(1-P_{e_1})^h = c\sqrt{\frac{n}{\log n}}(1-P_{e_1})^{\sqrt{\frac{n}{\log n}}} \tag{2}$$

Now 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:

$$\mathbf{P}\{ret = i\} = P_{e_1}^i (1-P_{e_1}) \tag{3}$$

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

$$F(h=1) = \sum_{i=1}^{\infty} \frac{W}{i} \mathbf{P}\{ret = i-1\} = \sum_{i=1}^{\infty} \frac{W}{i} P_{e_1}^{i-1}(1-P_{e_1}) \tag{4}$$

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

$$F(h=1) = -\frac{(1-P_{e_1})\ln(1-P_{e_1})}{P_{e_1}} W \tag{5}$$

For multi-hop networks, 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

$$\begin{aligned} F(h) =& \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{\infty} \cdots \sum_{i_h=1}^{\infty} (\frac{W_1}{i_1} + \frac{W_2}{i_2} + \cdots + \frac{W_h}{i_h}) \\ & \cdot \mathbf{P}\{ret = i_1 - 1\} \mathbf{P}\{ret = i_2 - 1\} \cdots \mathbf{P}\{ret = i_h - 1\} \\ =& \mathbf{E}[\frac{W_1}{i_1}] + \mathbf{E}[\frac{W_2}{i_2}] + \cdots + \mathbf{E}[\frac{W_h}{i_h}] \\ =& -\frac{(1-P_{e_1})\ln(1-P_{e_1})}{P_{e_1}} (W_1 + W_2 + \cdots + W_h) \end{aligned} \tag{6}$$

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

$$A_1 = -c\sqrt{\frac{n}{\log n}} \frac{(1-P_{e_1})\ln(1-P_{e_1})}{P_{e_1}} \tag{7}$$

Note in Eqn(6), we set the retry limit to $\infty$. 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 $\infty$, the aggregate throughput $A_1$ will be less than the result give in Eqn(7). 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(7). We use the results given by the infinity retry limit for the standard 802.11 protocol and header CRC.

[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.

### 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:

$$P_{e_2} = \sum_{i=1}^{k+r} (1-p)^{k+r-i} p^i \begin{pmatrix} k+r \\ i \end{pmatrix} \tag{8}$$

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:

$$A_2 = -c\sqrt{\frac{n}{\log n}} \frac{(1-P_{e_2})\ln(1-P_{e_2})}{P_{e_2}} \tag{9}$$

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. BCH codes are well known codes for binary data transmission, especially good for large block codes [8]. $m$ 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:

$$P_{e_3} \leq \sum_{i=t+1}^{k+m} (1-p)^{k+m-i} p^i \begin{pmatrix} k+m \\ i \end{pmatrix} \tag{10}$$

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:

$$A_3 = c\sqrt{\frac{n}{\log n}}(1-P_{e_3})^{\sqrt{\frac{n}{\log n}}} \tag{11}$$

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 (10), we have $P_{e_3} = 2.9884 \times 10^{-4}$ and $3.0578 \times 10^{-6}$ 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(11) 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

We use the no retransmission extreme scheme as a baseline for comparison. This basic scheme and header FEC scheme have the same format on the aggregate throughput, so by comparing $P_{e_1}$ with $P_{e_3}$ it is quite clear the latter has the advantage in terms of the scaling property of the throughput capacity. The other two ARQ schemes 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.

Fig.1 and Fig.2 show the aggregate throughput and per-node throughput as a function of the number of nodes in the network, respectively. Curves for header CRC and header FEC almost lap over each other. The difference between the performance of these two schemes and the others 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. Also, the bad performance for the base line (no retransmission scheme) is definitely undesirable. 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).

## 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 [1]. We build new protocol models based on our proposed schemes and

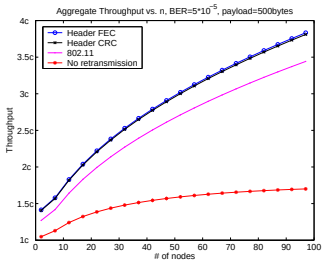



Fig. 1. Aggregate Throughput as a Function of $n$ for Different Schemes

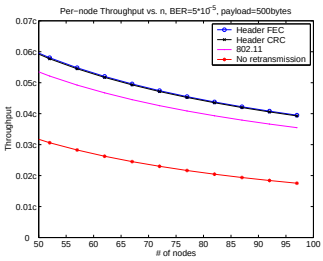



Fig. 2. Per-node Throughput as a Function of $n$ for Different Schemes

integrate them into *ns2*. Because of the lack of competition of no-retransmission scheme and it's unpopularity in real applications, we do not simulate it. We use a default packet retry limit of 4 for both 802.11 and our proposed header CRC protocol. The traffic type we use is CBR (Constant Bit Rate) traffic over UDP (User Datagram Protocol). The packet size used in all simulations is 500 bytes.

### 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. All of our simulations use 2Mbps radio. 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. The parameters for Markov are $p_{01} = 2.5 \times 10^{-5}, p_{10} = 0.5$, which yield an average channel BER $5 \times 10^{-5}$. The duration of each simulation is 2 minutes and the result is averaged upon 20 runs for different node distributions. The per-node throughput is shown in Fig. 3. If we compare Fig.3 with Fig.2, we see they share the same decreasing trend. When nodes are around 100, the throughput improvement by using header CRC or header FEC upon 802.11 is about 18%.

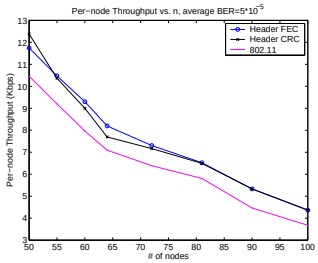



Fig. 3. Simulation Results on Per-node Throughput of Random Networks as a Function of $n$ for Different Schemes

### B. Video Simulations

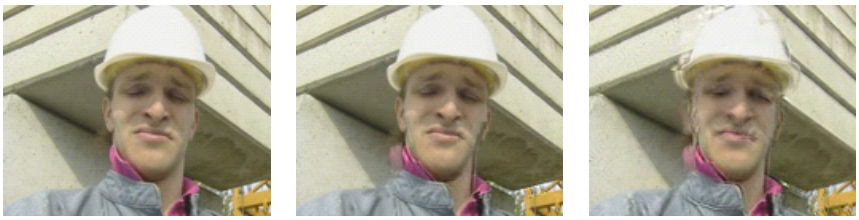
We start our video simulation with a network with only 2 nodes. We put 2 nodes 200 meters apart. Due to the significant overhead added by the exchange of RTS/CTS/ACK, when packet size is 500 bytes, the maximum throughput achievable for the 2 nodes under an error-free wireless environment is slightly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

We test our video simulation using a H.263+ coded bit-stream. "Foreman" video sequence is used with 300 frames length, QCIF (Quarter Common Intermediate Format, Quarter CIF) format [9]. 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 $10^{-5}$, $2.5 \times 10^{-5}$, and $5 \times 10^{-5}$, respectively. The simulation takes 20 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 distortions, 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.

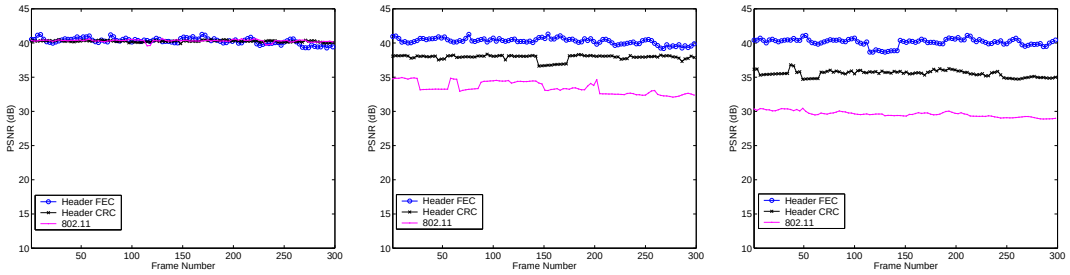
The advantage of using header error protection is more obvious in a multi-hop network, since retransmissions increase the traffic load and limit the throughput. In the next set of simulations we intend to find out the effective throughput of multi-hop networks under the three schemes. We use a single 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. The video traffic is sent from the first node to the last node, traveling through all the intermediate nodes. Fig.6 illustrates simulation results of the maximum throughput for a single 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 [7] (their throughput is a little bit higher since they do not have error model in *ns2*).

Header CRC performs not as good as header FEC, because



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

Fig. 4. Some Sample Video Clips for Different Schemes




(a) Average BER=$10^{-5}$ (b) Average BER=$2.5 \times 10^{-5}$ (c) Average BER=$5 \times 10^{-5}$

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

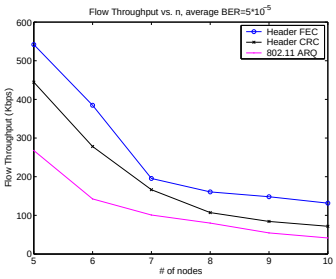



Fig. 6. Throughput as a Function of Chain Length $n$ for Different Schemes

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.

## 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. 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 and multi-hop chain networks. 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 queueing. This is one of the reasons the 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.

## References

[1] The network simulator- ns-2.

[2] 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, August 2004.

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

[4] David A. Eckhardt and Peter Steenkiste. Improving wireless lan performance via adaptive local error control. In *Proc. ICNP'98*, 1998.

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

[6] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, March 2000.

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

[8] Shu Lin and Daniel J. Costello. *Error Control Coding, 2nd Ed.* Pearson Education, 2004.

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

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

[11] M. Zorzi, R. R. Rao, and L. B. Milstein. Arq error control for fading mobile radio channels. *IEEE Transactions on Vehicular Technology*, 46(2):445–455, May 1997.