HEADER ERROR PROTECTION FOR MULTIMEDIA DATA TRANSMISSION IN WLANS

Su Yi, Yufeng Shan*, Shivkumar Kalyanaraman* and Babak Azimi-Sadjadi*◇*

*Department of ECSE, Rensselaer Polytechnic Institute, TROY, NY 12180
◇Intelligent Automation Inc. Rockville, MD 20855

ABSTRACT

In a wireless LAN, the throughput is essentially important for multimedia applications due to their high bit rate requirement. In this paper, we propose two link layer error protection schemes that improve the effective throughout of wireless networks. Error control is applied to the packet header and packet payload 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, error in payload may be corrected. We compare 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.

1. INTRODUCTION

Throughput and bit error rate are two important factors for wireless multimedia data transmissions. Many error control techniques have been proposed for error correction at both link layer and application layer. In this paper, we focus on how to improve the application layer effective throughput, by providing header CRC and header FEC at link layer, to efficiently support wireless multimedia data delivery.

Current 802.11 wireless LAN MAC protocol [1] is designed for reliable data transmission. One bit error in the link layer packet could cause the drop of the whole packet in the receiver side, even though other bits of the packet are successfully received. We argue that this approach is not optimal for multimedia data delivery, since the bit errors could be corrected at application by error control codes or the packet could be used by error resilience CODECs. Therefore, in order to efficiently support multimedia data transmission, we propose a new wireless link layer protocol, called HEP (Header Error Protection), where we only protect the header part, and packets with errors are forwarded to the next node or passed to the application for correction. The main purpose of protecting header is the need of carrying the correct routing information for packet forwarding. Besides, application header includes important information for video bitstream which is critical when decoding. Therefore header protected in this paper involves IP header and application header.

There are some interesting works on performance enhancement of video transmission over wireless networks. For instance, in [2], layered coding coupled with unequal error protection obtained by using different retry limits at the link level has been shown to balance the link erasure rate and the overflow rate. Zheng and Boyce [3] propose a modified version of the UDP protocol to accommodate Internet-to-wireless video traffic. A new protocol stack design is proposed to allow bi-directional information exchange so that the physical, link layers can communicate with the application layer. There are also arguments on whether error control should reside at link layer or at application layer [4]. Here, we provide another option - do part of the error control at the link level and leave some work done at the application level. Specifically we propose two header error protection schemes and analyze their impact on the throughput of the wireless networks. Our proposed scheme has the following two main advantages: increased effective bandwidth in conjunction with application layer error control codes; and reduced end-to-end delay, which are very important for multimedia streaming.

2. HEADER ERROR PROTECTION AND PERFORMANCE EVALUATION

In this paper we talk about link layer PDU, yet we still use the general term *packet* instead of *frame*, to differentiate with the term *frame* in video transmission. We consider three packet error control schemes, IEEE 802.11, our proposed header CRC and header FEC schemes. IEEE 802.11 MAC protocol uses a stop-and-wait ARQ with a positive ACK after each packet. The CRC checksum protects the whole packet so we call this scheme *packet CRC*. Our proposed *header CRC* aims to protect the header (including IP header and application header), not the whole packet. A packet retransmission is triggered if error detected in header part. In *header FEC*, the network nodes use an error control coding technique (FEC) to protect the header information from bit errors. Thus, the network might deliver the packet through with error in payload. A retransmission is issued when FEC redundancy fails to correct the errors in header, although it rarely happens in our simulations.

In this section, we evaluate the throughput performance of the three schemes. Consider an ad hoc network with n nodes randomly located in a unit area domain. It was shown in [5] that under a Protocol Model for interference, the average hop number h is assumed to be $\sqrt{\frac{n}{\log n}}$, and 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 result indicates a vanishing throughput performance as the network scales. The evaluation uses statistical approximations with these average values. All results will only hold with high probability.

2.1. Packet CRC in 802.11

In 802.11, usually there is a limit on the number of times a WLAN card can retransmit a packet (e.g., 4 times). The single hop packet error probability is defined as P_{e_1} for packet CRC. For simplicity, we use a simple random error channel in performance analysis, with bit error probability p. Since the errors are independent, $P_{e_1} = \sum_{i=1}^{q} (1-p)^{q-i} p^i \left(\begin{array}{c} q \\ i \end{array} \right)$, where q is the packet length (in bits).

First we assume there is no limit on the number of retransmissions. Given the probability of packet 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 $\mathbf{P}\{ret = i\} = P_{e_1}^i(1 - P_{e_1})$. If a flow only has one hop distance and the bandwidth is W, then the effective throughput of this flow is

$$\begin{aligned} F(h=1) &= \sum_{i=1}^{\infty} \frac{W}{i}\mathbf{P}\{ret = i-1\} = \sum_{i=1}^{\infty} \frac{W}{i} P_{e_1}^{i-1}(1-P_{e_1}) \\ &= -\frac{(1-P_{e_1})\ln(1-P_{e_1})}{P_{e_1}} W \end{aligned} \quad (1)$$

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

$$\begin{aligned} F(h) &= \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{\infty}\cdots\sum_{i_h=1}^{\infty}\left(\frac{W_1}{i_1} + \frac{W_2}{i_2} + \cdots + \frac{W_h}{i_h}\right) \\ &\cdot \mathbf{P}\{ret = i_1 - 1\}\mathbf{P}\{ret = i_2 - 1\}\cdots\mathbf{P}\{ret = i_h - 1\} \\ &= -\frac{(1-P_{e_1})\ln(1-P_{e_1})}{P_{e_1}}(W_1 + W_2 + \cdots + W_h) \end{aligned} \quad (2)$$

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 throughput. Therefore, we have

$$A_1 = -c\sqrt{\frac{n}{\log n}}\frac{(1-P_{e_1})\ln(1-P_{e_1})}{P_{e_1}} \quad (3)$$

Note in Eqn(2), we set the retry limit to ∞. In fact, the effect of a positive retry limit has diminishing return. For typical BER it is easy to prove that result of the 4 retry limit can be well approximated by Eqn(3). Therefore we use the results given by Eqn(3) for the standard 802.11 protocol and the proposed header CRC.

2.2. Header CRC

The probability that any error detected in header is $P_{e_2} = \sum_{i=1}^{k+r}(1-p)^{k+r-i}p^i \begin{pmatrix} k+r \\ i \end{pmatrix}$ 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 using header CRC is:

$$A_2 = -c\sqrt{\frac{n}{\log n}}\frac{(1-P_{e_2})\ln(1-P_{e_2})}{P_{e_2}} \quad (4)$$

Keep in mind 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.

2.3. Header FEC

FEC redundant bits are added to protect header part of each packet. BCH codes are well known codes for binary data transmission, especially good for large block codes [6]. 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 $P_{e_3} \leq \sum_{i=t+1}^{k+m}(1-p)^{k+m-i}p^i \begin{pmatrix} k+m \\ i \end{pmatrix}$

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 FEC:

$$A_3 = c\sqrt{\frac{n}{\log n}}(1-P_{e_3})^h = c\sqrt{\frac{n}{\log n}}(1-P_{e_3})^{\sqrt{\frac{n}{\log n}}} \quad (5)$$

We put the FEC protection bits to the tail of the packet, because errors tend to be in burst. This way the header and the protection bits are less likely to be corrupt simultaneously. Numerical result shows to protect header, 1 or 2 redundant bytes are enough.

2.4. 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 effective throughput. Fig.1 shows the throughput of the three schemes vs number of nodes in networks. The results are generated using MATLAB and based on the equation derived above. Factor c in the y-axis is the same as that in the above equations.

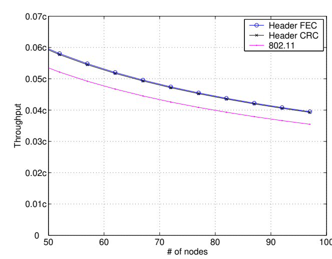


Fig. 1. Per-node Throughput as a Function of n

Our analytical results are valid for large networks (large n). We choose the amount of nodes from 50 to 100 in this plot to meet the practical scenario. Curves for header CRC and header FEC almost lap over each other. The gap between the performance of these two HEP schemes and 802.11 indicates that the performance inefficiency of the current protocol can be improved using header error control. 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. Header CRC is better for handling burst errors and header FEC 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).

3. SIMULATIONS

In this section, we evaluate our proposed HEP schemes using *ns2*[7]. We build new protocol models based on our proposed schemes and integrate them into *ns2*. The default packet retry limits is set to 4 for long packets and 7 for short packets - in both 802.11 and our proposed HEP protocols. We use RTP over UDP/IP with disabled checksum at UDP segment (or we can call it an extreme case of UDPLite). The packet size used in all simulations is 500 bytes. In addition, all of our simulations use 2Mbps radio.

A binary Markov channel model is used as channel error models in simulation. It can be 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 $\frac{p_{01}}{p_{01}+p_{10}}$.

3.1. Ns2 Simulations

Following the set up in section 2, we compare our proposed schemes with 802.11 in a random network. 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 CBR rate of the random traffic is chosen in order to place the network in a saturation state. In this state, there is 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, header CRC, and header FEC protocol. The parameters in Markov model 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 per-node throughput is shown in Fig.2. The simulation results reflect statistically significant analysis based on a 95% confidence interval shown with error bars. If we compare Fig.2 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. Nevertheless when nodes are around 100, the average throughput improvement by using header CRC or header FEC upon 802.11 is about 18%.

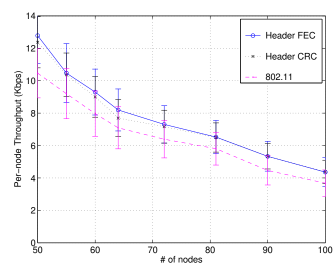


Fig. 2. Simulation Results on Per-node Throughput

3.2. Video Simulations

Video experiments are performed to show the corresponding video quality of using the three schemes, and the throughput capacity each scheme can achieve. The video sequence is "Foreman", QCIF, 30fps, total 300 frames. Video coder is H.263+ with error resilience feature.

3.2.1. A Single-hop Scenario

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, given packet size 500 bytes, the maximum throughput achievable for the 2 nodes under an error-free wireless environment is slightly above 1Mbps. Packets are then sent evenly spaced over approximately 4 milliseconds, resulting a data transmission rate 1Mbps. In this test, different channel conditions are used. Simulation takes 50 runs in each scenario for all the schemes. Note in order to correct the residual bit errors at the end user for the header CRC and header FEC strategies, we use a rate 1013/1023 BCH code at the application layer payload. This high rate code has a simple generator polynomial and has little overhead.

The averaged PSNR charts are shown in Fig.3. 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. Fig.3 indicates that 802.11 is much more sensitive to the channel BER than the other two. As a matter of fact seen from the expression of P_{e_1},P_{e_2}, and P_{e_3}, BER has a much higher impact on the packet loss in 802.11 than in HEP. This also shows that HEP can achieve reduced bandwidth fluctuation under a time-varying channel.

3.2.2. A Multi-hop chain topology with Cross Traffic

The advantage of using HEP is more apparent in multi-hop networks, since retransmissions increase the traffic load and limit the throughput. We place ad hoc nodes to set up a linear topology with cross traffic. In this grid, shown in Fig.4, the video traffic is carried from node S to node R. Besides this main traffic, there is cross traffic from node S_1 to R_1, S_2 to R_2, ..., and 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: $p_{01} = 1.25 \times 10^{-5}, p_{10} = 0.5$.

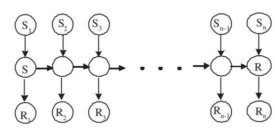
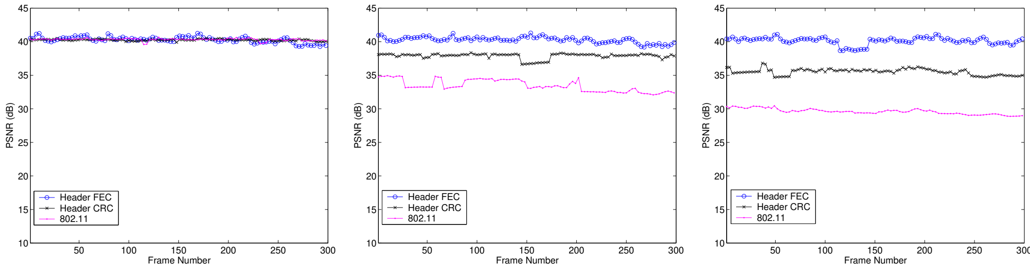


Fig. 4. A Chain Topology with Cross Traffic

In this chain topology with cross traffic, nodes in the main chain have to compete with each other and with the nodes transmitting cross traffic. The RTS/CTS/DATA/ACK mechanism requires nodes near the transmitting node or receiving node to keep silent. If a transmitting node 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.

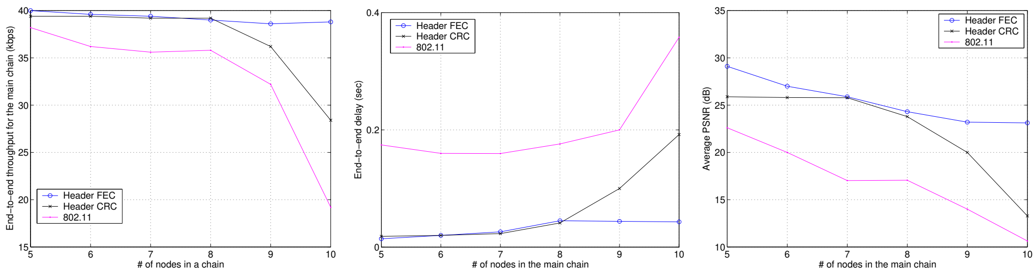
Fig.5(a) gives the result for the end-to-end throughput performance of the chain topology with cross traffic. The video is at 40kbps, with the average channel error rate 2.5×10^{-5}. The cross traffic is also at 40kbps. The plot illustrates the flow throughput of the main chain (middle chain) traffic versus the chain length n. Fig.5(b) 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.

802.11 MAC mechanism cannot discover the optimum schedule of transmissions on its own. Each node in a network experiences



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

Fig. 3. PSNR vs. Frame Number for Different Schemes in a Single-hop Network



(a) End-to-end Throughput (b) End-to-end Latency (c) Average PSNR

Fig. 5. Performance of the Chain Topology with Cross Traffic

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. 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 theory predicted, especially near the network saturation point. That is why the original 802.11 scheme has a much lower performance than the header protection schemes.

Fig.5(c) shows the average PSNR 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 evident for all three schemes. In this case, HEP, especially header FEC can effectively improve the PSNR values compared with 802.11, thus prevent the video quality from degrading rapidly.

4. CONCLUSIONS

This paper proposes two header error protection (HEP) 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 HEP 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 under a single hop network, 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 of 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.

5. REFERENCES

[1] 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.

[2] Q. Li and M. Van Der 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.

[3] 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.

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

[5] 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.

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

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