STANDARD COMPATIBLE ERROR CORRECTION FOR MULTIMEDIA TRANSMISSIONS OVER 802.11 WLAN

*Enrico Masala, Antonio Servetti, Juan Carlos De Martin**

Dipartimento di Automatica e Informatica / * IEIIT–CNR
Politecnico di Torino, Italy
E-mail: [masala | servetti | demartin]@polito.it

ABSTRACT

In this paper, we analyze a standard compatible error correction technique for multimedia transmissions over 802.11 WLANs that exploits, when available, the information of previous erroneous transmissions. The basic idea is to store erroneous frames for error correction purposes. More specifically, at the receiver each bit is estimated with a majority criterion. The performance of different standard compliant error recovery techniques have been evaluated using actual transmission experiments in various channel conditions. Optimal tradeoffs between complexity, memory and perceived quality have been determined studying the quality gains that can be achieved for the specific case of multimedia applications. Perceived quality has been evaluated using objective measures, e.g. ITU-T PESQ for voice and PSNR for video. Results show that the majority combining approach is particularly effective for multimedia communications, even in very noisy scenarios. Gains up to about one unit on the MOS scale for speech and up to 5-6 dB PSNR in case of video have been measured with respect to the standard ARQ technique.

1. INTRODUCTION

Real-time multimedia transmissions over the 802.11 wireless communication protocol [1] are gaining popularity due to the wide range of their possible applications, e.g. video streaming, videotelephony, video surveillance. Transmissions over wireless channels, however, are challenging due to the extreme variability of the radio channel. For these reasons, the 802.11 standard protocol incorporates an error detection mechanism and an automatic repeat request (ARQ) technique to retransmit corrupted packets.

The 802.11 standard ARQ technique has been designed for generic data transmission, therefore optimizations are possible for specific scenarios. For instance, combinations of FEC codes and retransmission mechanisms have been proposed for multimedia transmission. Those hybrid ARQ techniques often provide strong performance improvements, but the majority of them also need to be implemented at the application level, therefore requiring modifications of existing multimedia applications.

Packet combining techniques [2], instead, do not require modifications at the application level. They exploit the multiple transmissions typical of ARQ schemes. The basic idea is to store the previous transmissions of the packet, even if erroneous, and then to attempt packet recovery. Any recovery technique is applicable, because each correction attempt can be verified by means of the packet checksum, already present in the 802.11 MAC packet. Moreover, note that this approach do not require any modification to the 802.11 standard, and no additional retransmissions are required, compared to the 802.11 standard ARQ technique.

This work was supported in part by MIUR, Project FIRB PRIMO, http://primo.ismb.it

bytes: 2	2	6	6	6	2	0–2304	4
Frame Control	Duration/ ID	Address1	Address2	Address3	Sequence Control	Frame Body	FCS

← MAC Header →

Fig. 1. Frame format of an 802.11 data frame MPDU.

In the context of the packet combining approach, techniques based on the xor operation applied to pair of packets have been proposed [3] [4] to detect error positions. Correction is then attempted by brute force inversion of bits in every error position, but the applicability is limited by complexity constraints. Others propose to combine packets using a majority criterion to determine the values of each bit [5]. Diversity can also be efficiently employed to combine independent copies of the same packet [6].

None of these works, however, focused on the specific case of multimedia transmissions. Moreover, for simplicity's sake, uniform error distributions are often assumed and performance is typically shown for the case of just two or three combined packets.

The aim of this paper is to study packet combining techniques that do not require any modification of the 802.11 standard in order to extend their applicability to multimedia communications in realistic environments. The performance of different techniques is evaluated using actual transmission experiments in various channel conditions. Optimal tradeoffs between complexity, memory and perceived quality are determined studying the quality gain that can be achieved by multimedia applications for different settings of the studied techniques. Perceived quality has been evaluated using objective quality measures, such as the ITU-T perceived speech quality (PESQ) standard for voice and the peak signal-to-noise ratio (PSNR) for video.

The paper is organized as follows. In Section 2 we briefly review the 802.11 wireless standard. Section 3 illustrates in details the different packet combining schemes employed in the experiments. Section 4 explains the experimental setup and analyzes the results. Techniques are first compared in terms of generic network performance metrics, then perceived quality results for speech and video are presented. Conclusions are drawn in Section 5.

2. THE IEEE 802.11 STANDARD

The IEEE 802.11 standard covers two layers of the OSI reference model: the *medium access control* (MAC) and the *physical* (PHY) layer. We consider in detail the MAC part of the standard that is responsible for channel allocation procedures, frame formatting, error checking, fragmentation, and reassembly.

The fundamental function that provides fair access to the channel and best effort service is the *distributed coordination function* (DCF) that is based on a *carrier sense multiple access* with *collision avoidance* (CSMA/CA) algorithm. To deal with the collision problem, and other severe sources of errors such as interference, fading, and attenuation, the 802.11 protocol incorporates positive acknowledgments, i.e., all transmitted frames must be acknowledged if correctly received. If no ACK is returned, just after a transmission, the frame is scheduled for retransmission, until a maximum retransmission limit is reached.

Figure 1 shows the generic 802.11 MAC data frame. We briefly describe its format as a reference for the following sections. For a detailed discussion of the MAC standard refer to [1]. The frame type (and subtype) of the current transmission is detected by means of the Frame Control field. It provides flags for protocol version, fragment, and retransmission identification. The Duration field is used to let all stations know how long the medium is expected to remain busy for the transmission in progress. An 802.11 frame contains three address fields. The general rule of thumb is that Address 1 is used for the destination, Address 2 for the source, and Address 3 field is used for filtering by the receiver. The 16-bit Sequence Control field is composed of a 12-bit sequence number subfield and a 4-bit fragment number subfield. A unique sequence number is given to each high-level frame. When frames are retransmitted, the sequence number is not changed. Finally, each 802.11 MAC frame uses a 32-bit checksum, Frame Check Sequence (FCS) field, to verify the data-unit integrity. All fields in the MAC header and the body of the frame are covered by the FCS.

3. ENHANCED ARQ STRATEGIES

In the current IEEE 802.11 MAC standard no attempt is made to correct erroneous packets: error detection provided by the FCS requires a retransmission even for a single erroneous bit. A relatively simple and standard-compatible way to improve the reliability of WLAN communications is to retain received erroneous frames which are normally discarded by the standard ARQ. Memory ARQ schemes combine several of such corrupted packets at the receiver to attempt to reconstruct the original error free packet. The average number of combined copies varies according to the channel condition, thus the effective degree of protection is dynamic. Each information packet, in fact, contains a parity check sequence for error detection so that the receiver can determine when to stop the packet combining algorithm because the original packet has been fully recovered.

Unlike conventional 802.11 receivers, in the described methods the receiver stores an erroneous received packet before requesting a retransmission. For the purpose of error control, every different data MPDU can be identified by the 16-bit sequence control field that indicates its sequence number. Using the 32-bit FCS field at the receiver, the received packet can be checked for errors. If it is error-free, a positive acknowledgment is sent to the transmitter, inhibiting further retransmissions and the packet can be forwarded to the next hop or to the application level.

If it is not, the packet is dropped, but stored in the receiver buffer waiting for a retransmission. All the received packets are then processed by the error correction algorithm to be described.

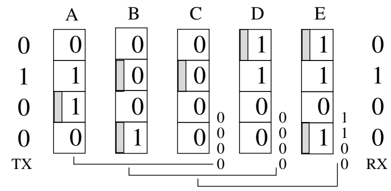

Fig. 2. Example of packet combining with majority decision. Frame 0100 is transmitted five times, bit errors (marked with gray boxes) are present in every copy. Bit values are estimated using every set of the last three received frames, and all five transmissions.

If the procedure is not able to recover a correct packet further retransmissions are necessary. Transmissions are repeated till a correct frame is received, the data in the cumulative buffer of received packets is correctable, or the maximum retransmission limit is reached.

A first combination scheme, here referred to as *xor combining* [3][4], consists in xor-ing two erroneous copies to locate errors in both packets. The decision process then involves a brute-force bit-by-bit inversion of the located bit error positions and checking for correctness using the FCS. When two copies are erroneous this operation fails if there is at least one bit position in which both copies have an error, or alternatively, if the total number of erroneous locations exceeds a given N_{max}. To make the algorithm implementable, in fact, an upper limit of computational complexity must be defined limiting practical values of N_{max} to 10, 11, or 12. Given a buffer size greater than two packets, more than one combination of packets is available for xor-ing. If no error recovery is possible, however, a retransmission is sought.

A second combination scheme, hereafter *majority combining* [5][6], is proposed to overcome the performance limitation of working on packet pairs only and uses the last three received erroneous copies of a packet. If xor combining fails, then a bit-by-bit majority decision can be performed to construct a new packet where a bit is one if it is one at least in two of the combined copies. The error correction algorithm succeeds if no bit-error overlaps occur. This idea can then be extended to cover combinations of more than three copies where a majority decision over the last M packets is used as an estimate of the transmitted bits.

The concept of combining packets to obtain a more reliable estimate of the transmitted bits can offer a significant advantage especially in applications where repeated packets can present a significantly different error rate such as in case of a wireless connection. For these applications, the possibility of selecting packets allows to avoid severely damaged ones. For instance in [7], all the possible combinations of M stored packets are considered by exploiting the availability of an error detection code to verify the correctness of their combination. For example, assume that, for a given information packet, the maximum number of transmissions allowed is four and the received packets corresponding to the four transmissions are A, B, C, and D. In this *generalized majority combining* scheme, after the last transmission, the receiver can combine all the possible triplets together with several choices, i.e., ABC, ABD, BCD, ACD. Each combination does not use all available packets, but it is able to obtain good recovery by avoiding potentially highly damaged packets. For instance, a combined ABC or BCD packet may have fewer errors than a combined ABCD packet.

In Fig. 2, we present the case of packet combining with a maximum of five transmissions. With the shown error pattern, majority combining with a receiver buffer that stores only the last three packets cannot correctly reconstruct the frame. Buffering more than three packets improves the error correction capability at the expense of increased memory requirements and complexity because more combinations need to be tested. In fact, combination of packet ABCDE, as well as early combination ACD are capable of reconstructing the correct estimate.

In the following sections, we will compare the performance of the xor combining, the majority combining and the generalized majority combining schemes through experiments under different channel conditions, and for actual multimedia traffic.

4. PERFORMANCE ANALYSIS

In this section our main aim is to present the numerical results of a comprehensive study for the evaluation of the proposed techniques in an 802.11b WLAN environment with an emphasis on their effect on the perceived quality of speech and video communications. Multimedia applications, in fact, particularly benefit from enhanced ARQ techniques because of their high sensitivity to packet losses. The reduction in the number of packets dropped by the network produces significant improvements on the user satisfaction, measured with two quality metrics, namely the perceived speech quality (PESQ) algorithm and the peak signal-to-noise ratio (PSNR).

The performance of packet combining techniques has been analyzed in previous works by means of analytical models representing different environments from binary symmetric channels (BSC) to Gaussian noise (AWGN) channels or Rayleigh fading channels. However, their performance has not yet been tested using experimental error traces. For this study we set up a scenario that consisted of two wireless stations equipped with an 802.11b PCMCIA card transmitting in different channel conditions and environments. The source station sends well-known packets to the destination node which can identify the position of bit-errors using a bit-wise comparison of the received data with the known payload.

In order to capture and analyze erroneous MAC frames a modified version of the wireless device drivers (Linux Wlan-Ng ver 0.2.1-pre20) was developed. When in monitor mode all received packets are processed including successful and unsuccessful (i.e., packets failing the 802.11b standard MAC layer checksum) transmissions. Packets with a bad checksum are then buffered and subsequent retransmissions are used to recover bit-errors as explained in Section 3.

Packet combining is applied matching successive retransmissions by their source address and sequence number pairs. This implies that part of the MAC header must be correctly received otherwise no recovery will be possible. In particular we need at least the destination address, the frame control, the sequence control, and the destination address fields to be error free.

Table 1. Packet combining techniques.

Technique	Description
ARQ	IEEE standard ARQ
XOR-10	xor combining ($N_{max} = 10$)
MA-3	majority combining ($M = 3$)
MA-5	majority combining ($M = 5$)
GMA-4	generalized majority combining ($M = 4$)
GMA-5	generalized majority combining ($M = 5$)

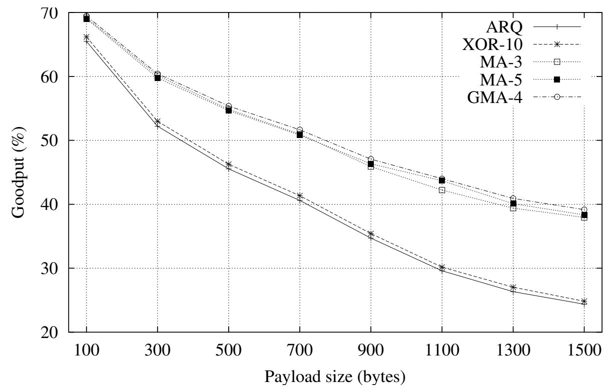

Fig. 3. Goodput as a function packet payload size for different packet combining techniques. Maximum retransmission limit is set to six.

4.1. Experimental results

We carried out experiments to evaluate the performance of the packet combining schemes both for transmission of generic data and for the specific case of multimedia (voice and video) transmission.

In the first set of experiments, we evaluate the system goodput using the combining techniques listed in Table 1. Here we define the goodput as the number of correctly received packets divided by the number of transmitted packets. We also assume that the acknowledgment is error free. Figure 3 shows the goodput performance for different sizes of the packet payload. Channel bit-error rate (BER) is constant and refers to a transmission with mean BER equal to $3.6 \cdot 10^{-3}$. We can see that xor combining does not significantly improve the goodput performance for any packet size. It is important, in fact, to notice that xor-ing two packets locates as many error positions as the sum of the erroneous bits in the two packets. As a consequence, because of the bursty nature of wireless errors, it rarely happens that this number is less than the given N_{max} (the maximum number of error locations that can be checked with acceptable complexity). On the contrary a substantial performance gain is achieved when majority combining is employed on those packets. Furthermore, the improvement effect grows as the packet payload increases.

Comparison of results for MA-5 and GMA-4 shows that memory buffer size can be traded for complexity with a slight yet interesting increase in performance. This confirms that in a WLAN scenario having different combining choices avoiding to combine severely damaged frames is preferable to combining all received packets. The additional delay at the receiver that can be reduced by applying parallel processing techniques, or by testing new possible combinations immediately after the reception of a new retransmission.

The second set of experiments measures the packet combining techniques for the case of voice and video transmission. We simulated transmission of speech compressed with the G.711 coder at 64 kb/s in frames of 20 ms [8]. Each frame is sent in a different packet with maximum six retransmissions. At the receiver the speech frame is concealed if lost (using frame repetition), decoded, and then played out. The perceived quality is evaluated with the ITU-T P.862 perceived speech quality (PESQ) algorithm, on a five-point (1–5) mean opinion score (MOS) scale [9].

Fig. 4 shows that when the channel condition is good all schemes perform well. When the channel deteriorates, the efficiency of the

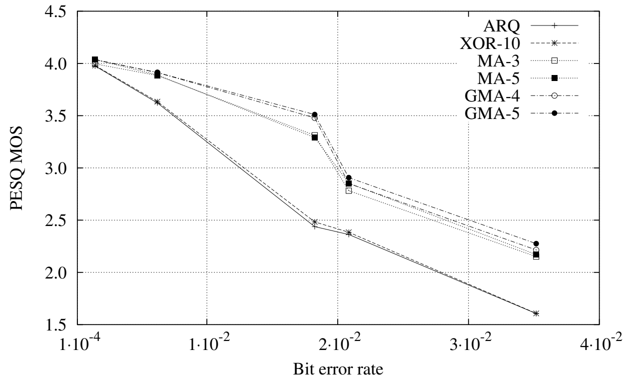

Fig. 4. Performance in terms of PESQ-MOS score for G.711 speech. Packet combining techniques are evaluated in several environment with different mean bit error rates. Retransmission limit is set to six.

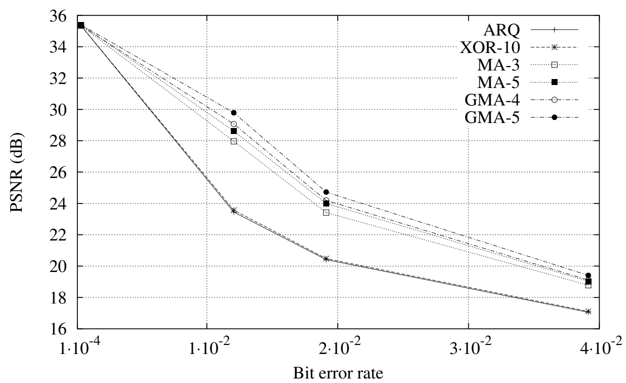

Fig. 5. PSNR performance for H.264-coded video (*foreman* sequence). Packet combining techniques are evaluated in several environment with different mean bit error rates. Retransmission limit is set to six.

standard ARQ and of the xor combining algorithm drops rapidly, while the error correction provided by the majority combining technique maintains the speech quality at a higher level with gains up to 1 in the MOS scale. Moreover, ARQ with generalized majority combining, GMA-4 and GMA-5, performs better than MA-3 and MA-5 resulting in a small, but measurable quality gain.

We also studied the performance of H.264 video transmissions [10] using different standard sequences. We show results for the *foreman* sequence only, but similar results were obtained for other standard test sequences. The sequence has been coded at 15 fps, 125 kbit/s (QCIF), using the intra refresh mechanism to combat packet losses. Slices are composed of three rows of macroblocks, and each slice is put into a separate packet. Up to six retransmissions are allowed for each packet. The decoder uses a temporal concealment technique that substitutes the missing areas of each frame with the ones in the previous frame. Perceptual quality at the receiver is evaluated using the PSNR distortion measure.

Figure 5 shows the performance of the packet combining techniques in case of video transmission. It is clear that the majority combining techniques can sustain twice as much channel bit error rate compared to the standard ARQ technique. When considering the same channel conditions, gains up to 5-6 dB are observed. The xor-based technique provides very similar performance compared to the standard ARQ technique, in accordance with the above results. Finally, note that the ARQ with generalized majority combining (GMA-4/-5) can deliver a consistent quality improvement with respect to the standard majority combining techniques (MA), with gains up to 1 dB.

5. CONCLUSIONS

We studied an 802.11 standard compliant error correction technique for multimedia transmissions. Information of previous erroneous transmissions is exploited to recover packets by means of a majority criterion applied to each bit of the packet. The performance of different standard compliant error recovery techniques have been compared using actual transmission experiments in various channel conditions, showing the optimal tradeoffs between complexity, memory and perceived quality. Gains up to about one unit on the MOS scale for speech and up to 5-6 dB PSNR in case of video were obtained, with respect to the standard ARQ technique, even in very noisy scenarios.

6. REFERENCES

[1] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *ISO/IEC 8802-11, ANSI/IEEE Std 802.11*, 1999.

[2] D. Chase, "Code combining — a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets," *IEEE Transactions on Communications*, vol. COM-33, no. 5, pp. 385–393, May 1985.

[3] P. S. Sindhu, "Retransmission error control with memory," *IEEE Transactions on Communications*, vol. COM-25, no. 5, pp. 473–479, May 1977.

[4] S. S. Chakraborty, E. Yli-Juuti, and M. Liinaharja, "An adaptive ARQ scheme with packet combining," *IEEE Communications Letters*, vol. 2, no. 7, pp. 200–202, July 1998.

[5] M. Gidlund, "Receiver-based packet combining in IEEE 802.11a wireless LAN," in *Proc. IEEE Radio and Wireless Conference (RAWCON)*, August 2003, pp. 47–50.

[6] Y. Liang and S. S. Chakraborty, "ARQ and packet combining with post-reception selection diversity," in *Proc. 60th IEEE Semiannual Vehicular Technology Conference (VTC Fall)*, 2004.

[7] Q. Zhang and S. A. Kassam, "Hybrid ARQ with selective combining for fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 867–874, May 1999.

[8] ITU-T Rec. G.711, "Pulse code modulation (PCM) of voice frequencies," *ITU-T*, November 1988.

[9] ITU-T Rec. P.862, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," *ITU-T*, February 2001.

[10] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, "Advanced video coding for generic audiovisual services," *ITU-T*, May 2003.