A Modified Medium Access Control Algorithm for Systems with Iterative Decoding

Inkyu Lee, *Senior Member, IEEE,* Carl-Erik W. Sundberg, *Fellow, IEEE,* Sunghyun Choi, *Senior Member, IEEE*, and Wonjun Lee, *Member, IEEE*

***Abstract*— Efficient transmission methods for fading radio channels often require an iterative decoder. This is for example the case for systems using turbo codes. Receiver decoder iterations could potentially lead to a latency problem which impacts the performance of the medium access control protocol. In this paper, we present modifications based on the carrier sense multiple access with collision avoidance (CSMA/CA) medium access control (MAC) protocol to accommodate the increased latency in the iterative processing. One area of applications is wireless local area networks (WLANs) with high data rate. The simulation results performed in the IEEE 802.11a WLAN environment by replacing the 802.11a's convolutional coding with turbo coding demonstrate that the proposed algorithm provides a throughput gain over the conventional method.**

***Index Terms*— Medium access control, CSMA/CA, WLAN, iterative decoding.**

I. Introduction

SIGNAL design for overcoming the impairment of fading in radio channels is currently an active area of research. One potential application area of this research is upgrades of wireless local area networks (WLANs) based on the IEEE 802.11 standards [1]. One way of radio link improvement is to use turbo codes with iterative decoding [2], [3]. Another way of improving the radio link is to use space-time coding systems with multiple transmit and receive antennas, also employing iterative decoding [4], [5], [6].

To obtain high data rates, the modem constellation is typically of multi-level M-QAM type [7]. It has been shown that for most systems of this type, an efficient decoder utilizes the turbo principle and performs iterative decoding (ID) [2], [8]. The iterative decoding procedure leads to a potential latency problem that could impact the performance of the medium access control (MAC) protocol.

Fig. 1 shows an example of a receiver structure which contains an iterative decoder. In this example, a serially concatenated coded system [3], [9] is shown where both the inner decoder and the outer decoder are operating iteratively. We emphasize that this is only an example of iterative decoding. The iterative decoding takes place by means of two maximum a posteriori (MAP) decoders using the turbo principle. After the inner decoder computes the reliability values for the received packet, those values are passed on to the outer decoder. Then, the iterations continue until the preset number of iterations has been reached. Since the final output packet is not ready until the last iteration, the total latency is proportional to the packet size and the iteration number.

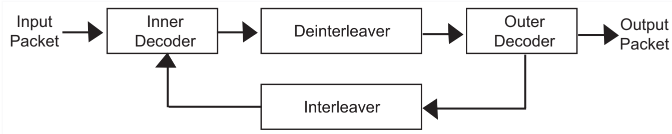

Fig. 1. Iterative decoding structure for serially concatenated system.

The default MAC protocol for the IEEE 802.11 standard [1] for WLANs is called distributed coordination function (DCF), which is a carrier sense multiple access with collision avoidance (CSMA/CA) scheme [10]. In this paper, we review this protocol and propose modifications for allowing the decoder to perform iterations. These iterations typically improve the frame error probability considerably [2], [3], [9], compared to one-pass decoders of Viterbi algorithm type.

The rest of the paper is organized as follows: In Section II of this paper, we present a modification to the Medium Access Control (MAC) operation which incorporates the increased latency in the iterative processing. In Section III, performance evaluation is carried out to validate the proposed algorithm. The paper concludes with discussion in Section IV.

II. MAC Layer Operations with Iterative Decoding Scheme

The MAC layer specified by the 802.11 standard [1] is based on CSMA/CA. Since it is not possible to detect a collision in the wireless medium, the collision avoidance scheme is employed instead of the collision detection method [10]. This MAC mechanism is characterized by the immediate transmission of a positive acknowledgement (ACK) by the destination station, upon successful reception of a packet transmitted by the source station. If the source station does not receive a proper ACK frame within a certain time period called "ACK Timeout", the previous transmission is considered to have failed, and the source attempts to retransmit the packet.

Fig. 2 illustrates the time alignment between a data frame from a source station and an ACK frame from the destination station, as defined in the 802.11 standard. After receiving a correct data frame from the source station, the destination

Manuscript received May 11, 2003; revised April 12, 2004; accepted May 7, 2005. The associate editor coordinating the review of this letter and approving it for publication was T. Boucouvalas. This research was supported by the Ministry of Information and Communication (MIC), Korea, under the Information Technology Research Center (ITRC) support program, supervised by the Institute of Information Technology Assessment (IITA).

I. Lee and W. Lee are with Korea University, Seoul, Korea (e-mail: {inkyu, wlee}@korea.ac.kr).

C.-E. Sundberg is with SundComm, USA (e-mail: cews@ieee.org).

S. Choi is with Seoul National University, Seoul, Korea (e-mail: schoi@snu.ac.kr).

Digital Object Identifier 10.1109/TWC.2006.02003.

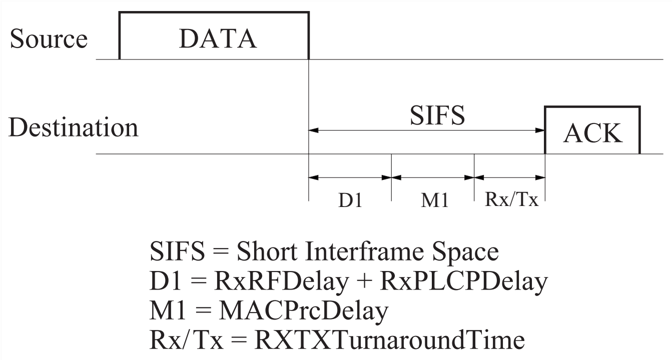

Fig. 2. Basic MAC procedure.

station sends back an ACK frame following a short interframe space (SIFS) period. The value of SIFS depends on the underlying physical layer (PHY). For example, in the case of 802.11a PHY [11], the SIFS period time is set to $16\mu s$. In the PHY, the transmitter starts transmitting a packet consisting of a preamble, a physical layer convergence protocol (PLCP) header and a scrambled and encoded ACK frame after the SIFS period expires. Again, in the case of 802.11a, the duration including the preamble and the PLCP header is defined to $20\mu s$. The D1 and M1 periods in the diagram account for mostly the processing delays in the PHY and the MAC, respectively. Therefore, the PHY should finish decoding the received frame in time during the D1 period such that the MAC has enough time to carry out all the necessary processing such as a cyclic redundancy code (CRC) check for the received data frame before the M1 period ends.

Iterative decoding schemes, e.g. [9], introduce a longer latency in the receiver output, as the final CRC information of the received data frame will not be available until the whole iteration process is completed. It takes longer for the destination station to send the ACK response. That implies that the SIFS value should be increased to accommodate the extended ACK response delay. This is undesirable since the total channel throughput is decreased as demonstrated in the subsequent section. In this section, we describe a new MAC mechanism to minimize the impact due to the increased latency in the iterative decoding scheme.

We can alleviate this problem by starting the ACK frame transmission even before the final CRC check information is available. After a data frame has arrived, the receiver starts the iterative decoding process for the received packet. After the first iteration (or some more iterations if time permits), the decoded frame is sent to the MAC, while the receiver continues the iteration process. Now, the MAC checks if the received frame is intended for itself by looking at the receiver address field. Upon positive match of the address field, the MAC prepares the ACK frame by putting together the MAC header including the recipient address without checking the CRC of the received data frame. Then, this ACK frame is sent to the PHY and the transmitter begins to transmit the ACK frame starting from the preamble.

At the same time, the receiver continues the iterations, and after the final iteration (assuming the transmitter is still transmitting the preamble and the PLCP header), the decoded frame is sent to the MAC. Then, the MAC checks the CRC information. If the CRC is correct, no further action is required. In the case the CRC turns out to be incorrect after the final iteration, the MAC alters the recipient address field, which has been set to the source station address after the first iteration, to its own address. We are able to modify this address information as long as the PHY has not finished transmitting the preamble and PLCP header. Changing the address information at this moment should not be a very complicated operation, especially, since the address is replaced by a pre-determined and fixed one, i.e., its own address. When this address change occurs, the transmitter (i.e., source) of the data packet does not receive an ACK frame with its address. Then, the source attempts to transmit the data packet again via CSMA/CA. Note that today's MAC implementation will work exactly in this manner since the data transmitter expects the correct reception of an ACK "with its own address" within the ACK timeout according to the 802.11 MAC.

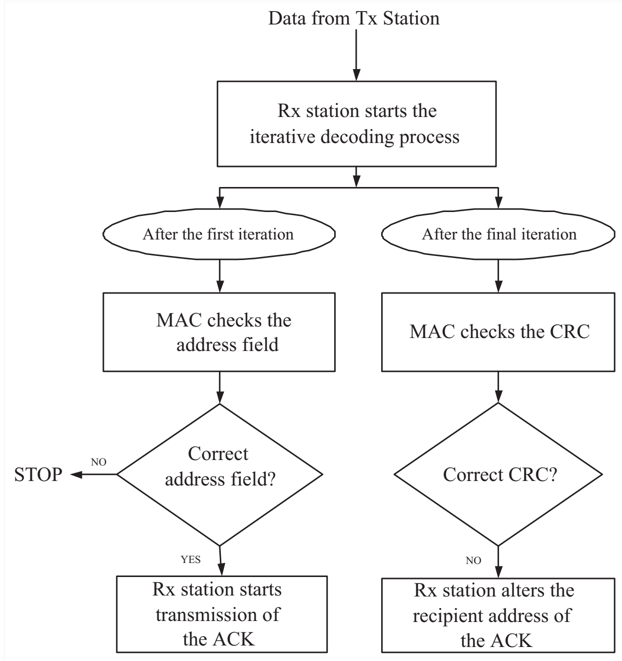

Fig. 3. Flow chart of the proposed algorithm.

This modified scheme allows the iterative decoder to complete all iterations in $36\mu s$ combining the $16\mu s$ SIFS period and the $20\mu s$ preamble/header duration, which is much longer than a few μs (i.e., D1 in Fig. 2) given to the original MAC operation. Note that in this modified scheme the address field sent to the MAC before all iterations are completed may be more vulnerable for errors, as we may not be able to iterate sufficiently on that field. In addition, the PHY needs to be able to start transmission while the receiver still performs the iterations on the received data frame. Nevertheless, the modified MAC operation allows the iterative process to be continued well over the SIFS period. Fig. 3 shows the flow chart of the proposed MAC algorithm.

III. Performance Evaluation

In this section, we evaluate the utility of the proposed scheme based on the throughput performance. In fact, it is

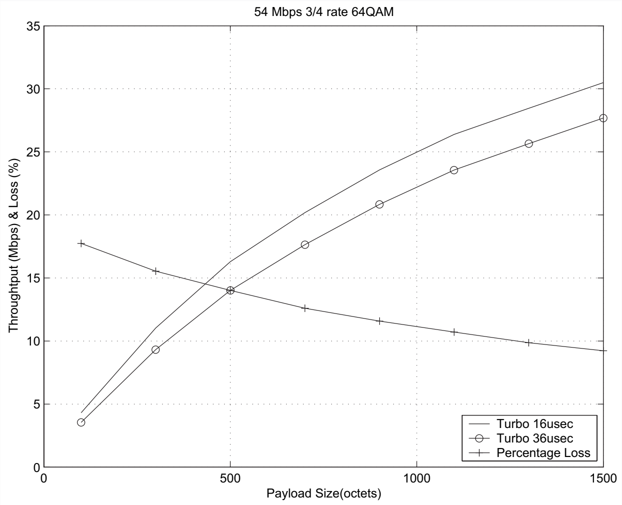

Fig. 4. Throughput vs. payload size for error-free channel.

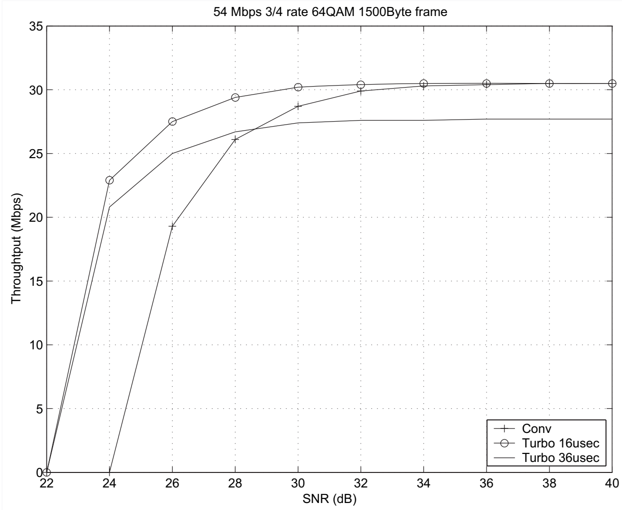

Fig. 5. Throughput vs. SNR for Turbo and Convolutional coding schemes.

not straightforward to quantitatively demonstrate the utility of the proposed scheme since it depends on the decoding time of the employed Turbo decoder. For example, if the decoding times of the turbo code and the 11a's convolutional code are the same, it is apparent that there will not be any advantage to use the proposed scheme. Here, assuming that the Turbo decoding time is 20 μs larger than the 11a's convolutional decoding time, we suppose that SIFS = 36 μs should be used in order to satisfy the ACK response timing if the proposed scheme is not employed, while the original SIFS, i.e., 16 μs, can be used if the proposed scheme is employed. Note that the SIFS value is PHY-dependent, and hence this value does not need to be 16 μs as in 802.11a when we define a new PHY by replacing the convolutional codes of 802.11a with turbo codes. It should be also noted that if SIFS=36 μs is used, DIFS becomes 54 μs since DIFS=SIFS+2*SlotTime, where SlotTime=9 μs. Accordingly, two versions of 802.11a-like WLAN with turbo codes instead of the corresponding convolutional codes are compared: (a) one with SIFS = 16 μs (as defined in the original standard), and (b) the other with SIFS = 36 μs (for the system without employing the proposed scheme).

We can easily see how these different SIFS values affect the system throughput through a simple numerical analysis [12], [13]. The following assumptions are made for the analysis. One sender and one receiver operate with the DCF mode. The sender always has frames to transmit. Each frame has a fixed-size payload. Finally, this throughput is determined at the Link Layer Control (LLC) Service Access Point (SAP), which is the interface between the MAC and its immediate higher-layer, i.e., the 802.2 LLC. In this analysis, we use 54 Mbps for the data frame transmissions and 24Mbps for the ACK transmissions.

Fig. 4 shows the throughput performance as the payload size of the data frame increases. Channel errors are not considered in order to emphasize the impact of the SIFS value difference. Apparently, the larger SIFS, the larger MAC overheads, thus degrading the throughput performance. We observe that there is a considerable loss in the throughput by using the larger SIFS value, e.g., 9%-18% loss as the payload size varies from 1500 to 100 octets. Note that 1500 octets represent the practically longest payload size in the 802.11 WLANs, since the maximum frame size allowed for the Ethernet, which typically interworks with the 802.11, is 1500 octets.

Now, we compare the throughput performance for erroneous channel environments in orthogonal frequency division multiplexing (OFDM) systems defined in the IEEE 802.11a standard. Basically, the assumptions used for the error-free analysis above still hold except for the channel errors. For turbo coding, a constituent code with polynomial (7,5) in octal form is applied with 5 iterations. A typical indoor wireless channel with a 5 tap exponentially decaying fading profile is applied. Fig. 5 depicts the throughput performance for (1) turbo coding with SIFS=16 μs, (2) turbo coding with SIFS=36 μs, and (3) convolutional coding with SIFS=16 μs (i.e., 802.11a) for the case of 54 Mbps rate and 1500 octet frame. We also allow up to 6 retransmissions upon transmission failure. Apparently, turbo coding attains higher throughput performance than convolutional coding for low SNR values. As expected, turbo coding with SIFS=16 μs achieves a throughput better than that of the 11a convolutional coding for all SNR values. However, turbo coding with SIFS=36 μs achieves a worse throughput compared to the 11a convolutional coding for large SNR values. This is because the impact of the time loss due to the larger SIFS value is much larger than that of the coding gain using turbo codes in this SNR range. Note that in this high SNR region, there is virtually no retransmissions due to the lack of channel errors with both turbo and convolutional coding schemes.

IV. Discussion and Conclusions

A modification of the MAC operation is introduced which allows longer latency in iterative decoding schemes. By modifying the MAC operation, the iterative decoding technique can be employed in systems with the CSMA/CA scheme without sacrificing channel throughput for the 802.11 wireless LAN systems. Simulation results confirm the validity of the proposed scheme.

Acknowledgment

The authors would like to thank W. Diepstraten and X. Wang for helpful discussions on the MAC issue.

REFERENCES

[1] IEEE 802.11, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, 1999.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. of IEEE International Conferences on Communications '93*, pp. 1064–1070, May 1993.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, pp. 909–926, May 1998.

[4] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 744–765, March 1998.

[5] I. Lee, A. Chan, and C.-E. W. Sundberg, "Space-Time Bit-Interleaved Coded Modulation for OFDM systems," *IEEE Trans. Signal Processing*, pp. 820–825, March 2004.

[6] I. Lee, A. Chan, and C.-E. W. Sundberg, "Space-Time Bit-Interleaved Coded Modulation for OFDM systems in Wireless LAN Applications," in *Proc. ICC '03*, pp. 3413–3417, May 2003.

[7] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1983.

[8] J. Hagenauer, "The Turbo Principle: tutorial introduction and state of the art," in *Proc. of International Symposium on Turbo Codes*, pp. 1–11, 1997.

[9] I. Lee, "The Effect of a Precoder on Serially Concatenated Coding Systems with ISI Channel," *IEEE Trans. Commun.*, pp. 1168–1175, July 2001.

[10] W. Stalling, *Data and Computer Communications*. Prentice Hall, 1997.

[11] IEEE 802.11, *Supplement to Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band*, 1999.

[12] D. Qiao and S. Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," in *Proc. ICC*, pp. 1995–2000, June 2001.

[13] D. Qiao, S. Choi, and K. G. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs," *IEEE Trans. on Mobile Computing (TMC)*, vol. 1, pp. 278–292, October/December 2002.