Receiver-based Packet Combining in IEEE 802.11a Wireless LAN

Mikael Gidlund
Telecommunication Research Group
Department of Information Technology and Media
Mid Sweden University
SE-851 70 Sundsvall, Sweden

***Abstract*— In this paper, we evaluate the performance of employing packet combining in 802.11a Wireless LAN (WLAN). Packet combining is well suited for wireless LANs since it provides a diversity gain to the system that can reduce the interference problem and enhance the system performance. Performance results show that a receiver employing packet combining provides significant gains in throughput and reductions in packet error probability when compared to with a receiver that does not employ combining. It is also found that for large packet sizes the gain is greater than for smaller packet sizes, because errors are more likely to occur in the header than in the data.**

I. Introduction

Recently, interest for wireless data transmission has grown rapidly in the past few years, spurred by the massive growth of Internet and other multimedia applications which all demands high-data rates and are bandwidth hungry. Therefore, new wireless networks with broadband capabilities are being sought to provide high speed integrated services with cost effective support for Quality-of-Service (QoS). IEEE 802.11a is a Wireless Local Area Network (WLAN) designed to provide high-speed communication, up to 54 Mbit/s, between portable devices attached to a IP or ATM backbone network [1]. IEEE 802.11a operates in the 5 GHz frequency band and utilizes Orthogonal Frequency Division Multiplexing (OFDM) as transmission scheme.

The IEEE 802.11 uses a simple ARQ consisting of just retransmitting packets if no acknowledgement has been received at the transmitter. This make packet combining to a feasible technique to utilize for increasing the system performance. Packet combining is performed (by the receiver) at the codeword level by combining the results of decoding the current packet with previous packets to make better decisions about that packet. Because the technique requires minimal additional complexity over standard approaches, it is applicable in a wide range of wireless data system applications. Packet combining has previously been investigated and showed substantial performance gain. In 1985, Chase [2] introduced code combining technique, which represents a technique for combining a minimum number of repeated packets which are encoded with a rate R to obtain a lower rate. If the receiver has L packets that have caused a retransmission request, these packets are concatenated to form a single packet encoded with rate R/L. Metzner and Chang [3], and Wicker [4] have proposed methods for combining packets using diversity (symbol-by-symbol) combining. In [6], the authors evaluated the performance of simple packet combining in wireless LAN, but no system-specific requirements was considered. However, that contribution showed that packet combining is well suited for WLAN's, especially 802.11, since CSMA/CA has a structure that allows the use of packet combining without considerable increase in complexity.

In this paper we investigate the possibility and usefulness of utilizing packet combining in 802.11a in order to evaluate if we can achieve some gain in terms of throughput without changing the standard. The packet combining technique used in this paper is based on code combining. Simulations show that we achieve a substantial performance gain for large packet sizes and high data rates.

The paper is organized as follows: Section II gives a brief overview of the IEEE 802.11a standard. In section III we present the used system model and packet combining technique. In section IV, we discuss the implementation aspects and in section V we discuss the simulation results. Finally, we conclude the work in section VI.

II. Overview of IEEE 802.11

A. Media Access Control in 802.11

The IEEE 802.11 has two different access methods, the mandatory Distributed Coordination Function (DCF) and the optional Point Coordinator Function (PCF). The latter aims at supporting real-time traffic.

DCF is the basic access mechanism of IEEE 802.11, and uses *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) algorithm to mediate the access to the shared medium. Before a data frame is sent, the station senses the medium. If the channel is idle for at least a *Distributed Interframe Space* (DIFS) period of time, the frame is transmitted. Otherwise, a backoff time T_{bkoff} is chosen randomly in the [0, CW), where CW is the so called $ContentionWindow$, called as $CW_i = 2^{k+i-1} - 1$, where i is the number of attempts to transmit the frame that has been done, and k is a constant defining the minimum contention, CW_{min}. After the medium has been detected idle for at least a DIFS, the backoff timer is decremented by one for each time slot the medium remains idle. When the backoff timer reaches zero, the frame is transmitted. Upon detection of a collision, a new backoff time is chosen and the backoff procedure starts over.

TABLE I

EIGHT PHY MODES OF THE 802.11A PHY

Mode	Modulation	Code Rate	Data Rate	BpS
1	BPSK	1/2	6Mbps	3
2	BPSK	3/4	9Mbps	4.5
3	QPSK	1/2	12Mbps	6
4	QPSK	3/4	18Mbps	9
5	16-QAM	1/2	24Mbps	12
6	16-QAM	3/4	36Mbps	18
7	64-QAM	2/3	48Mbps	24
8	64-QAM	3/4	54Mbps	27

After a successful transmission, the contention window is reset CW_{min}.

Another important characteristics of the MAC layer is that an acknowledgment (ACK) frame will be sent by the receiver upon successful reception of the data frame. Only after receiving an ACK frame correctly, the transmitter assumes successful delivery of the corresponding data frame.

B. Physical Layer in 802.11a

The physical layer of IEEE 802.11a employs OFDM transmission where the data stream is split into several parallel streams of reduced rate with each substream modulating a separate subcarrier. A key feature of the IEEE 802.11a PHY is to provide 8 PHY modes with different modulation schemes and coding rates. As listed in Table I, the OFDM system supports user-selectable data rates between 6 and 54 Mbps. Forward error correction is performed by bit interleaving and rate-1/2 convolutional code. The higher code rates are obtained by puncturing the rate-1/2 code.

III. SYSTEM MODEL

A. OFDM Baseband model

The 802.11a uses OFDM for transmitting data since it provide high data rates and can also improve transmission echo and distortion resulting from multi-path propagation and radio frequency interference. The basic principles of OFDM technology has been described in [7].

We assume a OFDM system with N subcarriers, a bandwidth of W Hz, a symbol length of T seconds, of which T_{cp} seconds is the length of the cyclic prefix, the transmitter uses the following waveforms

$$\phi_t = \begin{cases} \frac{1}{\sqrt{T-T_{cp}}} e^{j2\pi \frac{W}{N} k(t-T_{cp})} & \text{if } t \in [0, T] \\ 0 & \text{Otherwise .} \end{cases} \quad (1)$$

where $T = N/W + T_{cp}$ and $\phi_k(t) = \phi_k(t + N/W)$ when t is within the cyclic prefix $[0, T_{cp}]$, since $\phi_k(t)$ is rectangular pulse modulated on the carrier frequency kW/N. The waveforms $\phi_k(t)$ are used in the modulation and the transmitted baseband signal for OFDM symbol number i is

$$s_i(t) = \sum_{k=0}^{N-1} x_{k,i}\phi_k(t - iT) \quad (2)$$

where $x_{0,l}, x_{1,i}, \cdots, x_{N-1,i}$ are complex numbers from a set of signal constellation point. When an infinite sequence of OFDM symbols is transmitted, the output from the transmitter is then

$$s(t) = \sum_{l=-\infty}^{\infty} s_i(t) = \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N-1} x_{k,i}\phi_k(t - iT). \quad (3)$$

The mobile radio channel is modeled as a tapped delay line with impulse response

$$h(t) = \sum_{i=0}^{P-1} \alpha_i(t)\delta(t - \tau_i) \quad (4)$$

where α is complex Gaussian random variable tap weight, τ is the relative time delay of the ith path, and P is the number of paths. The received signal $r(t)$ can be expressed as follows

$$r_{i,k}(t) = h_k(t)x_{i,k}(t) + \eta_{i,k}(t) \quad (5)$$

where $\eta_{i,k}$ is additive white Gaussian noise with variance $\sigma^2 = N_0$ and h_k is the gain of the k-th subchannel. Based on channel estimates $\hat{h}_k$, we can equalize the received signal to obtain the signal estimates $y_{i,k} = \frac{r_{i,k}}{\hat{h}_k}$. With the symbol and channel vector estimates $\mathbf{y}$ and $\hat{\mathbf{h}}_\mathbf{k}$ respectively we can obtain the log-likelihood ratio of the code bits Λ. If we assumes perfect channel estimation we have $y_{i,k} = x_{i,k} + \eta'_{i,k}$ where $\eta'_{i,k}$ is a complex Gaussian random variable with variance $\sigma_k'^2 = N_0/|h_k|^2$. At the receiver a soft-input Viterbi decoder is used. With BPSK modulation, the input to the Viterbi decoder is the log-likelihood ratio [8]

$$\Lambda_{i,k} = \log \frac{Pr\{y_{i,k}|b_{i,k} = 1\}}{Pr\{y_{i,k}|b_{i,k} = 0\}} \quad (6)$$

which has a Gaussian distribution as $\Lambda_{i,k} = \frac{2}{\sigma^2} y_{i,k}|h_k|^2 = \frac{2}{\sigma^2} r_{i,k}h_k^*$. The multiplicative term $2/\sigma^2$ can be ignored by the Viterbi decoder since σ^2 is the same for all symbols and subcarriers [9].

B. Packet combining

To apply packet combining in 802.11a, we are using code combining as described in [2]. At the receiver, L different packet with same information is stored in a buffer. For the L repeated versions of the a given code bit $y_{i,k}$, with $0 \leq l < L$, we can express the log-likelihood ratio as

$$\begin{aligned} \Lambda_{i,k}^L &= \log \frac{Pr\{y_{i,k,0}, y_{i,k,1}, \cdots, y_{i,k,L-1}|b_{i,k} = 1\}}{Pr\{y_{i,k,0}y_{i,k,1}, \cdots, y_{i,k,L-1}|b_{i,k} = 0\}} \\ &= \log \prod_{l=0}^{L-1} \frac{Pr\{y_{i,k,l}|b_{i,k} = 1\}}{Pr\{y_{i,k,l|b_{i,k}} = 0\}} \\ &= \sum_{l=0}^{L-1} \Lambda_{i,k,l} \end{aligned} \quad (7)$$

where $\Lambda_{i,k,l} = r_{i,k,l}h_{k,l}^*$. It can easily be seen in (7) that optimum code combining is equivalent with maximum-ratio packet combining.

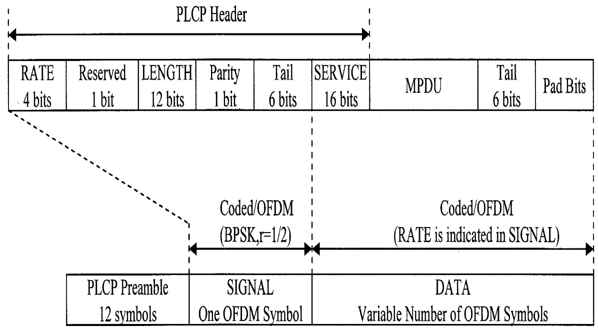


Fig. 1. PPDU frame format of the IEEE 802.11a OFDM PHY

TABLE II
IEEE 802.11A OFDM PHY CHARACTERISTICS

Characteristics	Value	Comments
tSlotTime	$9\mu s$	Slot time
tSIFSTime	$16\mu s$	SIFS Time
tDIFSTime	$34\mu s$	DIFS = SIFS+2· Slot
aCWmin	15	min contention window size
aCWmax	1023	max contention window
PLCPPreamble	16μs	PLCP preamble duration
PLCPSIG	4μs	PLCP SIGNAL duration
Symbol	4μs	OFDM symbol interval

IV. Implementation

When implementing packet combining in IEEE 802.11a, we have to consider some system specific aspects. We should mention that the implementation of packet combining in this case is quite straightforward since the IEEE 802.11 uses a simple ARQ scheme consisting of just retransmitting packets if no positive acknowledgment has been received at the transmitter.

In IEEE 802.11 standard, the PLCP protocol unit (PPDU) frame format provides the asynchronous transfer of MAC sublayer MPDUs from any transmitting Access Point (AP) to all receiving mobile stations (STAs) within a WLAN's Basic Service Set (BSS). The PPDU consists of PLCP preamble, PLCP header, MPDU, tail bit and pad bits as shown in Fig. 1 [1]. The PLCP preamble field, with a duration of $PLCPPreamble$, is composed of 10 repetitions of a short training sequence and 2 repetitions of a long training sequence. The PLCP header except the SERVICE field, with the duration $PLCPSig$ constitutes of a single OFDM symbol, which is transmitted with BPSK modulation and rate-1/2 convolutional coding. The tail bits are used too return the convolutional codec to the "zero state", and pad bits are used to make the resulting bit string to be a multiple of OFDM symbols. Each OFDM symbol interval, $Symbol$, is $4\ \mu s$. The 16-bit field SERVICE field of the PLCP header and the MPDU, represented by DATA, are transmitted at the data rate specified in the RATE field. Table II lists the related characteristics for the IEEE 802.11a PHY. What one must consider when implementing packet combining in this case, is that the information bits are scrambled before encoding, such that repeated packets are not exactly the same, and to minimize the data dc bias and maximum run lengths. The scrambling sequence is obtained from the first seven bits of the 16-bit SERVICE field, which is transmitted in the beginning of each packet [1]. We consider $\mathbf{d}$ to be the desired information bit sequence. At the l-th packet the scrambling sequence $\mathbf{s}_l$ is employed, and the data sequence can be expressed as $\mathbf{d}_l = \mathbf{s}_l \oplus \mathbf{d}$ where $\oplus$ is the binary addition operator. Now we let $z_s(\cdot)$ represent the coding operation. Since we are using linear convolutional codes in IEEE 802.11a, we have $\mathbf{b}_l = z_s(\mathbf{d}_l) = z_s(\mathbf{s}_l) \oplus z_s(\mathbf{d})$. Hence, scrambling can then be accounted for by changing Eq. (7) to

$$\Lambda_{i,k}^{L} = \sum_{l=0}^{L-1} (2s_{i,k,l} - 1)\Lambda_{i,k,l}, \tag{8}$$

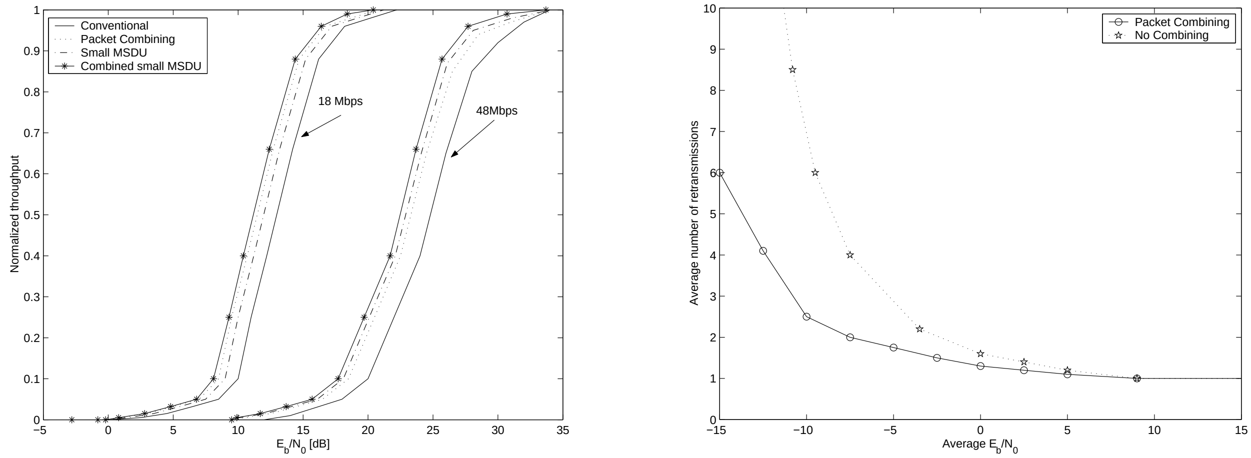
where $s_{i,k,l}$ corresponds to the k-th subchannel of the i-th symbol of the l-th coded scrambling sequence.

Another system-specific item we must consider is that a MAC header is transmitted in the beginning of each packet. This MAC header contains the packet target address and packet sequence number, and packet combining should only be applied if these parameters are the same as for the stored packets in the receiver buffer. This implies that the MAC header must be detected without employing packet combining. The MAC header may differ in different retransmissions of the same packet, e.g., in the retry bit, which is zero for the first transmission of a packet and one otherwise. Therefore, also the last four bytes of a packet, which corresponds to the frame check sequence (FCS), as well as the six termination bits of the convolutional code may be different.

With these two technical properties in consideration, we can conclude that we can only apply packet combining on pure data bits, but not on the overhead bits. Usually we have packet lengths of several hundreds of bytes, up to a maximum of 2312 bytes, so the overhead loss is kept to a minimum. If the SNR is not to low, an erroneous packet contains usually just a few bit errors, which are much likely to occur among the data bits. So we can easily see that packet combining is still quite effective if we just apply it on the data bits. The physical layer must have some knowledge about the overlying MAC layer to utilize packet combining, i.e., especially the header and FCS size. A false packet combining can actually occur in the case that the header causes their addresses and sequence number to be equal.

V. Performance Evaluation

In this section, we will try to assess the effects of employing packet combining in IEEE 802.11a. A fully compliant IEEE 802.11a PHY and MAC layer simulation has been developed according to the parameters set in [1]. The multipath radio channel considered in this paper is specified as in [10]. It contains different channel models representing different environments, with tapped delay lines modelled as Rayleigh



(a) Throughput performance for receiver packet combining with different packet sizes (200, 1000).

(b) Number of retransmissions vs SNR.

Fig. 2. Simulation results.

or Rice fading. Channel time variance was modelled with classical Jake's Doppler spectrum corresponding to a terminal speed of 3 m/s on each tap of the channel impulse response [11]. We consider the channel to be constant until a packet is received correctly, which is a reasonable assumption since we can expect an almost static environment for WLANs, as long as not too many retransmission are needed.

In Fig. 2(a) the throughput with and without packet combining for different data rates and packet sizes are presented as a function of SNR. We defined throughput as the number of correctly received packets divided by the number of transmitted packets. It is observed that for large packet sizes we obtain a substantial performance gain when packet combining is employed on data bits. For smaller packet sizes the gain is small, this depends mainly on the fact that most part of the errors occurs in the MAC header, which cannot be combined as mentioned before. If we consider a scenario where we have perfect packet combining performed, i.e., assuming perfect header detection, we can achieve a quite substantial performance gain. We can then conclude that better header protection means higher performance gain. This could for instance be accomplished by employing lower data rates for the header than the data. Another solution can be to protect the header with for example a RS-code [12] and then we will achieve better performance, but this requires a change in the standard. In Fig. 2(b), we can clearly see that packet combining decrease the number of transmissions. For instance when the SNR is -10dB, we just requires two retransmissions in average for the case with packet combining, while the case without no packet combining requires eight retransmissions.

VI. Concluding Remarks

In this paper we have investigated potential of utilizing packet combining in IEEE 802.11a wireless LAN without any changes within the current standard. Packet combining proves to be well suited in IEEE 802.11 which employs CSMA/CA MAC protocol, since only good packets are acknowledged and bad packets are just ignored by the receiver. Simulation results show that a large performance gain can be achieved for large packets, while for smaller packets the performance gain become smaller, because errors are more likely to occur in the header than in the data.

References

[1] IEEE Standard 802.11a, PART II. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: High Speed physical layer in the 5 GHz band. Technical specification ANSI/IEEE Std. 802.11, 1999 Edition, IEEE, Piscataway, New Jersey, USA, August 1999.

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

[3] J. J. Metzner and D. Chang, "Efficient selective ARQ strategies for very noisy and fluctating channels," *IEEE Trans. Commun.*, vol. COM-34, no. 3, pp. 244-249, May 1985.

[4] S. B. Wicker, "Adaptive rate error control through the use of diversity combining and majority logic decoding in a hybrid ARQ protocol," *IEEE Trans. Commun.*,

[5] Kallel S. and Haccoun D., "Sequential decoding with ARQ and code combining: A robust Hybrid FEC/ARQ system," *IEEE Trans. Commun.*, vol. COM-36, no. 7, pp. 773-780, July, 1988.

[6] M. Gidlund and S. B. Slimane, "Performance enhancement of Wireless LANs through packet combining," in *Proc. IEEE VTC'01-spring*, Rhodes, Grecce, May 2001.

[7] R. V. Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*, Artech House publishers, Dec. 1999.

[8] A. Burr, *Modulation and Coding for Wireless Communications*, Prentice-Hall, 2001.

[9] J. Proakis, *Digital Communications*, McGraw-Hill, New York, 2001.

[10] J. Medbo and P. Schramm, "Channel models for HiperLAN/2," ETSI BRAN document no. 3ERI085B, 1998.

[11] W. C. Jakes, *Microwave Mobile Communication*, IEEE Press 1974.

[12] S. Choi, and K. G. Shin, "A class of adaptive hybrid ARQ schemes for wieless links," in *IEEE Trans. Veh. Techn.*, Vol. 50, No. 3, May 2001.