RZ 3382 (# 93428) 11/26/01
Electrical Engineering 5 pages

Research Report

Receiver Packet Combining in OFDM-based Wireless LANs

André Noll Barreto

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
aba@zurich.ibm.com

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties). Some reports are available at http://domino.watson.ibm.com/library/Cyberdig.nsf/home.

IBM Research
Almaden · Austin · Beijing · Delhi · Haifa · T.J. Watson · Tokyo · Zurich

Receiver Packet Combining in OFDM-based Wireless LANs

André Noll Barreto

IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract

In a packet transmission system erroneous packets can be stored and combined with later repetitions of the same packet to increase the data throughput. This technique is known as packet combining, and its application to the IEEE802.11a standard will be investigated in this letter. The achievable performance gain will be obtained under restrictions imposed by the standards.

I. Introduction

Wireless local-area networks (W-LAN) are becoming increasingly popular because of their high data rates, low costs, and easy installation. Systems based on the IEEE 802.11b standard operating on the 2.4 GHz ISM band are already widespread, and support data rates of up to 11 Mbps. The next generation of W-LANs is under development, and both the European HIPERLAN/2 and the North-American IEEE 802.11a standards operate in the 5 GHz frequency band and utilise coded orthogonal frequency division multiplexing (OFDM) to transmit at data rates of up to 54 Mbps [1].

W-LAN standards are all packet-based transmission systems and rely on error-correcting codes (ECC), cyclic redundancy checks (CRC) and automatic repeat request (ARQ) schemes to guarantee nearly error-free transmission of data packets in a hostile wireless environment.

High-speed W-LAN systems are targeted at static or slow-moving applications in an indoor environment, where typically the channel changes very slowly. For instance, at walking speeds ($V \simeq 1$m/s) with carrier frequency $f_c = 5$GHz the maximum Doppler spread is $f_d = V f_c/c = 16.67$Hz, which yields a coherence time $T_c \simeq 1/2f_d = 30$ms, corresponding to 15 MAC frames in HIPERLAN/2 (there is no fixed frame length in IEEE802.11a). Under these channel conditions, a packet may have to be retransmitted many times or with a large delay between retransmissions until it is received without errors.

A significant improvement can be achieved if retransmitted packets are combined before decoding. Even erroneous packets contain some valuable information, and, instead of discarding them, they can be stored and combined with the signal received at subsequent retransmissions. This approach, known as packet or code combining, was extensively investigated in the literature, for instance in [2]–[7] and, unlike conventional ARQ schemes, it was shown to permit reliable transmission even under very low signal-to-noise ratios (SNR). Packet combining can be very effective in increasing throughput, and requires no modification in existing standards. In this letter we investigate the application of packet combining to an IEEE 802.11a transmission link.

II. System Description

Let bold letters represent data vectors. The input data packet of N_{pack} bits $d(n)$ is divided into blocks of N_b bits $\mathbf{d}_i$, corresponding to the i-th OFDM symbol. These are coded with a rate-R_c convolutional code, and the $N_c = N_b/R_c$ code bits $\mathbf{b}_i$ are then mapped to $K_d = N_c/\log_2(M)$ QAM or QPSK symbols $\mathbf{x}_i$, where M is the constellation size. K_p pilot and K_z zero subcarriers are introduced, and the signal goes through a K-point inverse fast Fourier transform (IFFT), with $K = K_d + K_p + K_z$. To the time-domain signal thus obtained we add at the beginning of each K-sample symbol a cyclic prefix of G samples, which eliminates multipath interference up to a delay spread of $T_G = GT_s$, where T_s is the sampling interval. This signal is filtered, converted to radio frequency, and transmitted through a multipath channel.

The receiver performs the reverse operations. The received signal is filtered, converted to baseband, and sampled at a rate $1/T_s$. The cyclic extension is removed, and a fast Fourier transform (FFT) performed. The zero and pilot subcarriers are removed, and the received signal at the k-th subchannel after this operation is

$$r_{i,k} = h_k x_{i,k} + \nu_{i,k}, \tag{1}$$

where h_k is the gain of the k-th subchannel and $\nu_{i,k}$ a complex white noise component with variance $\sigma^2 = N_0$.

Based on channel estimates $\hat{h}_k$, we can equalise the received signal to obtain

$$y_{i,k} = \frac{r_{i,k}}{\hat{h}_k}. \tag{2}$$

Assuming that perfect channel estimates are available we have

$$y_{i,k} = x_{i,k} + \nu'_{i,k}, \tag{3}$$

where $\nu'_{i,k}$ is a complex Gaussian random variable with variance

$$\sigma'^2_k = N_0/|h_k|^2. \tag{4}$$

We consider a soft-input Viterbi decoder at the receiver. Let us suppose that BPSK modulation was employed, i.e., $x_{i,k} = 2(b_{i,k} - 1)$. The input of the Viterbi decoder is the log-likelihood ratio

$$\Lambda_{i,k} = \log \frac{\Pr\{y_{i,k}|b_{i,k} = 1\}}{\Pr\{y_{i,k}|b_{i,k} = 0\}}, \tag{5}$$

which, as ν_k has a Gaussian distribution, is given by

$$\Lambda_{i,k} = \frac{2}{\sigma^2} y_{i,k}|h_k|^2 = \frac{2}{\sigma^2} r_{i,k} h_k^*. \tag{6}$$

Because σ^2 is the same for all symbols and all subcarriers, the multiplicative term $2/\sigma^2$ can be ignored by the Viterbi algorithm.

The physical-layer parameters used throughout this work are those from IEEE802.11a, but they also apply to HIPERLAN/2 with only marginal modifications.

III. Packet Combining

Now suppose the same information is transmitted over L different packets. This can be seen as transmission with a new code of rate $R'_c = R_c/L$, composed of the concatenation of the original R_c-code and a repetition code. As shown in [3] this new code has distance properties just slightly less favourable than those of a maximum-free-distance code of the same rate. The gist of this code-combining scheme is that as the coding rate is reduced, the same information can be transmitted reliably at a lower SNR. The L repeated versions of a given code bit $y_{i,k,l}$, with $0 \leq l < L$, belong to the same branch of the Viterbi decoder, and we need now to obtain the log-likelihood ratio

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

with

$$\Lambda_{i,k,l} = r_{i,k,l} h_{k,l}^*. \tag{8}$$

From the equations above we can see that optimum code combining is equivalent to a maximal-ratio packet combining. Similar expressions can be obtained for higher-order Gray-coded QPSK or QAM modulation, except that there each symbol x_k corresponds to several code bits, and a different expression may be needed for each code bit.

Implementation in IEEE 802.11a

In IEEE 802.11 the ARQ scheme simply consists in retransmitting a packet if no positive acknowledgement (ACK) is received, which renders the implementation of packet combining straightforward. However, some system-specific aspects have to be taken into account.

We first need to consider that the information bits are scrambled before encoding, such that repeated packets are not exactly the same. Let $\mathbf{d}$ be the desired information bit sequence. At the l-th packet the scrambling sequence $\mathbf{c}_l$ is employed, and the data sequence

$$\mathbf{d}_l = \mathbf{c}_l \oplus \mathbf{d} \tag{9}$$

is transmitted, where $\oplus$ is the binary addition operator.

Now let $f_c(\cdot)$ represent the coding operation. Because linear convolutional codes are employed, we have

$$\mathbf{b}_l = f_c(\mathbf{d}_l) = f_c(\mathbf{c}_l) \oplus f_c(\mathbf{d}). \tag{10}$$

Hence, scrambling can be accounted for by changing (7) to

$$\Lambda_{i,k}^{L} = \sum_{l=0}^{L-1} (2c_{i,k,l} - 1)\Lambda_{i,k,l}, \tag{11}$$

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

As already mentioned, the optimum code combining scheme corresponds to a maximal-ratio combining of the received signal $r_{i,k,l}$. We can hence store either the weighted values of the received signal or the log-likelihood ratio of each code bit. Both approaches amount to the same with BPSK modulation, but for higher-order modulation schemes each symbol corresponds to several code bits. The former approach has the advantage of requiring less storage memory, but the latter is more flexible, as it can accommodate packets retransmitted using different modulation schemes.

The scrambling sequence is obtained from the first seven bits of the 16-bit SERVICE field, which is transmitted at the beginning of each packet. This means that we must first detect these bits without employing packet combining. Furthermore, a MAC header is transmitted at the beginning of each packet. Among other things, this header contains the packet target address and a packet sequence number, and packet combining should only be applied if these parameters are the same as for the stored packets. This implies that the MAC header must be also detected without employing packet combining. The MAC header is typically 24 bytes long for a data packet, and it 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 correspond to the frame check sequence (FCS), as well as the six termination bits of the convolutional code may be different.

The considerations above mean that packet combining can be applied only on the pure data bits, but not on the overhead bits. We can however expect packet lengths of several hundreds of bytes, up to a maximum of 2312 bytes, such that the overhead loss is kept to a minimum. The total overhead amounts to approximately 31 bytes. As long as the SNR is not too low, an erroneous packet contains usually just a few bit errors, which are much more likely to occur among the more numerous data bits. With this in mind it is easy to see that packet combining is also effective when applied on the data bits only.

For packet combining the physical layer must have some knowledge about the overlying MAC layer, in particular the header and FCS size, and the position of the sequence number and address within the header must be known. This conflicts in principle with the ISO/OSI layered model, but no modifications in the standards are required. Furthermore, packet combining is transparent to the MAC layer, which does not have to be informed whether packet combining is used. The retransmission scheme is still under control of the MAC layer, but the packets received from the physical layer are less likely to be in error after a few retransmissions. However, the packet buffering can be rendered more efficient if the physical layer is informed by the MAC layer whether a packet was received correctly, which is decided based on the FCS bits.

A false packet combination can in principle occur, i.e., different packets can be combined if any error in the header causes their addresses and sequence number to be equal. This is however very unlikely, as these parameters are 20 bytes long. In any case the falsely combined packet would be detected by the frame check in the MAC layer with very high probability.

IV. Performance Results

In this section we investigate the performance of a IEEE 802.11a system employing packet combining through simulation. We consider transmission over a frequency-selective Rayleigh-fading channel, more specifically the ETSI type-A small-office channel model [8]. We also consider a simple retransmission scheme, in which erroneous packets are simply retransmitted at the next frame. The channel is considered to be constant until a packet is received correctly, which is reasonable in the nearly-static environment expected for W-LANs,

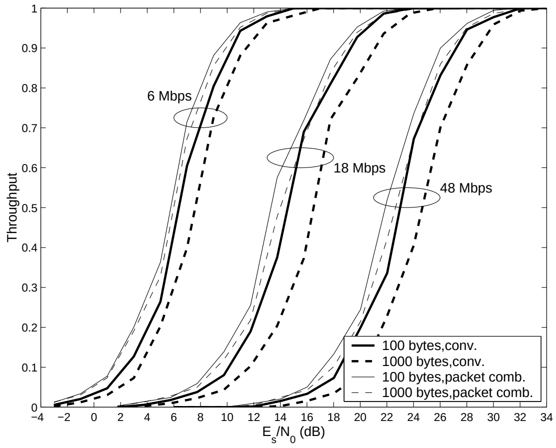

Fig. 1. Throughput with packet combining and different packet sizes

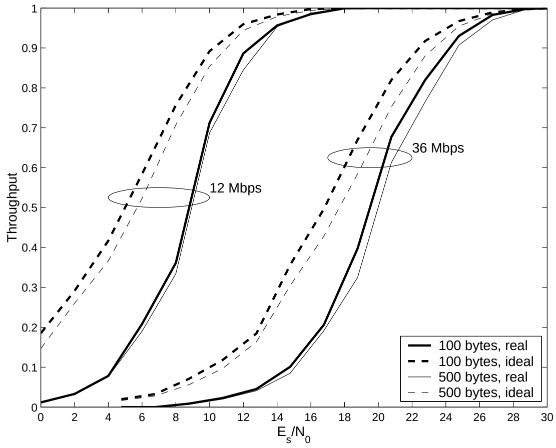

Fig. 2. Throughput with ideal and realistic packet combining

as long as not too many retransmissions are needed. To that purpose we limit the number of retransmissions to ten, after which the packet is considered to be lost.

In Fig. 1 the throughput with and without packet combining is displayed for different packet sizes and different data rates (different transmission modes). The throughput is defined here as the number of correctly received packets divided by the total number of packets transmitted. We can see that the performance improves little with packet combining for short packets, as in this case a great part of the packet errors occur in the header, which cannot be combined. Longer packets have a higher packet error rate (PER), and hence a lower throughput, but the performance gain is much higher when packet combining is employed, because for low error rates most errors are likely to occur in the data sequence, whose reliability can be increased through packet combining.

As we can see for instance from the performance results with short packets in Fig. 1, the performance of packet combining is limited by the error rate of the MAC header, which cannot be improved through combining. In Fig. 2 we display the throughput gains that can be expected if ideal packet combining is performed, i.e., assuming perfect header detection, and they are quite substantial. This means that if the MAC header can be transmitted with higher reliability, for example by employing lower data rates for the header than for the data, we could achieve significant performance gains, especially at low SNRs. This would however require a modification in the standards. Note that the throughput curves for packet combining rapidly approaches zero for low SNRs, but only because the number of repetitions is limited to ten. If infinite repetitions were allowed, the curves would tend only asymptotically to zero with decreasing SNRs.

References

[1] R. Van Nee, G. Awater, M. Morikura, H. Takanashi, M. Webster, and W. Halford, "New high-rate wireless LAN standards," *IEEE Commun. Mag.*, vol. 37, no. 12, pp. 82–88, Dec. 1999.

[2] P. S. Sindhu, "Retransmission error control with memory," *IEEE Trans. Commun.*, vol. COM-25, no. 5, pp. 473–479, May 1977.

[3] 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.

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

[5] F. Adachi, S. Ito, and K. Ohno, "Performance analysis of a time diversity ARQ in land mobile radio," *IEEE Trans. Commun.*, vol. 37, no. 2, pp. 177–183, Feb. 1989.

[6] B. A. Harvey and S. B. Wicker, "Packet combining systems based on the Viterbi decoder," *IEEE Trans. Commun.*, vol. 42, nos. 2/3/4, pp. 1544–1557, Feb./March/April 1994.

[7] K. R. Narayanan and G. L. Stuber, "A novel ARQ technique using the turbo coding principle," *IEEE Commun. Lett.*, vol. 1, no. 2, pp. 49–51, Mar. 1997.

[8] "Criteria for comparison," Tech. Rep. EP BRAN#9, 30701F, ETSI, Sophia Antipolis, France, July 1998.