SOME NEW ARQ PROTOCOLS FOR PERSONAL COMMUNICATION SYSTEMS

Alessandro Andreadis, Giuliano Benelli, Andrea Garzelli
School of Engineering, University of Siena
Via Roma, 56, 53100 Siena, Italy
Tel: +39 577 263601; fax: +39 577 263602
e-mail: benelli@unisi.it

ABSTRACT

Mobile communication channels are frequently plagued by severe noise and disturbances such as multipath fading and doppler effects that severely degrade performance. Among the automatic-repeat-request (ARQ) protocols used to improve the communication channel reliability, the stop-and-wait (SW) is positively characterized by simple implementation and negatively by low throughputs. This work describes the application of some new SW protocols that retain the simple implementation of the classical SW schemes, while reducing the transmitter's wait state time to increase throughput. The performance of the modified SW protocols, derived through computer simulations, is shown to be comparable to that of more complex ARQ protocols.

1 INTRODUCTION

Disturbances such as multipath fading and Doppler effects markedly reduce the reliability of the transmitted information in digital mobile communication channels [1]. A common remedy is to use automatic-repeat-request (ARQ) or forward-error-correction (FEC) channel coding techniques to improve performance. In ARQ techniques, when the error-detecting code discovers an error in a received vector, it sends the transmitter a retransmission request via a feedback channel, with the procedure repeated until a correct codeword is received. In FEC techniques, the error patterns introduced by the transmission channel are corrected by an error-correcting code.

The mobile channel is often affected by fading due to the multipath, which varies rapidly, and heavy disturbances. Therefore, powerful FEC techniques, able to correct many error patterns, are generally required to meet the error probability requirements; the decoder complexity can be quite high. In contrast, ARQ techniques use error detecting codes, which present a low implementation complexity and, at the same time, offer a high reliability also on severe fading conditions and disturbances.

The more powerful FEC techniques are used to combat multipath-generated fading because they are capable of correcting large numbers of error patterns and satisfying the error probability requirements albeit at the cost of high decoder complexity. By contrast, the ARQ techniques, which are exceedingly reliable, even with severe fading and distortions, are of low implementation complexity.

The ARQ protocols may be classified into three main categories: stop-and-wait (SW), go-back-N (GBN), and selective repeat (SR) [2-3]. In the case of the SW, the transmitter sends a codeword and, before sending the next codeword, awaits an acknowledgment (positive or negative) from the receiver. The GBN performs more efficiently, since the transmitter sends codewords continuously until a negative acknowledgment is received, and then retransmits the codewords sent during the elapsed time. However, its efficiency dips sharply at high error rates and/or with long round-trip delays. The SR is the most efficient of the three, since only the error blocks need be repeated, but it is of complex implementation and requires an extensive, theoretically infinite, buffer at the receiver side.

The advantage of the SW's simple implementation is offset by its low throughput resulting from the enforced acknowledgment waiting times following each transmission. Even the throughput of the SW modified by Moeneclaey et al. [4], in which each message is transmitted $m \geq 2$ times consecutively, while higher than that of the classical SW for comparatively low signal-to-noise ratios, still falls below acceptable values when signal-to-noise ratios reach medium or high levels. Many other modified ARQ protocols have been recently introduced to improve the performance of communication systems using error detecting codes [5-6].

This work presents some new SW protocols that significantly improve throughput, without relinquishing their main advantage, simple implementation. Their performance has been derived by computer simulation of a channel model for mobile communications. The resulting performance is comparable to those of more complex ARQ protocols such as the GBN and SR protocols. The proposed protocols can be efficiently applied to several channel models, such as the model used for simulations, derived from [7], which is typical of land mobile radio systems.

2 PROPOSED SW PROTOCOLS

This section describes some new SW protocols and their special features. The new protocols, while sharing many characteristics of the classical SW scheme, have a different receiver structure and, in addition, in transmission send consecutive codewords before entering the wait state.

The information source generates symbols from a finite alphabet A with M elements. These symbols are encoded by a code C of type (n,k). Hence, each block of k information symbols is encoded in a codeword $\underline{c}$, n symbols long. Code C is assumed capable of correcting $t \geq 0$ errors.

The received vector $\underline{r}$, n symbols long, is decoded by the C decoder. If an errorless or correctable error pattern is detected, a positive acknowledgment (ACK) is sent to the transmitter through the feedback channel. Conversely, a negative acknowledgment (NACK) is sent for an uncorrectable error pattern. Using D to denote the round-trip delay and v_s to denote the transmission rate, we can write the number N of C codewords that can be transmitted during the round-trip delay as

$$N = [Dv_s/n] \tag{1}$$

where $[x]$ denotes the smallest integer greater than or equal to x. The transmitter can be either in the transmission, retransmission, or wait state. It enters the transmission (retransmission) state after receiving an ACK (NACK) signal from the receiver via the feedback channel, remaining in the transmission (retransmission) state for the duration of a time-slot, i.e., the time required to send a codeword, and then passes to the wait state, where it remains (N - 1) time-slots.

2.1 Modified memory-less SW protocols

The classical SW protocol is of simple implementation and does not require a buffer at the receiver side. However, it is penalized in terms of throughput because of the time it wastes in the wait state, $(N-1)$ time-slots. The first SW protocol described in this section, hereafter designated SW1, reduces wait times, thereby affording a corresponding gain in throughput. In the SW1, the codewords to be transmitted are divided into groups, each composed of $M \geq 1$ codewords. The ith codeword is denoted $\underline{c}_i$, where $i \geq 1$. The M codewords $\underline{c}_{(j1)M+1}$, $\underline{c}_{(j-1)M+2}$, ..., $\underline{c}_{jM}$ constitute the jth codeword group. Let us assume the (j-1)th codeword group has been successfully delivered to the user. The transmitter enters the transmission state and sends the jth codeword group, i.e., the M codewords, one at a time, after which it enters the waiting state. The number of time-slots the transmitter spends in the wait state varies according to the ACK and NACK sequence it receives via the feedback channel. The receiver can be either in the decoding or wait state. In the decoding state, it analyzes the received vector $\underline{r}_{(j-1)M+i}$ for $1 \leq i \leq M$ and, if the received vector is errorless or contains a correctable error pattern, it sends an ACK to the transmitter. Conversely, if $\underline{r}_{(j-1)M+i}$ contains an uncorrectable error pattern, it sends a NACK so that all the successive (M - i) received blocks are discarded, regardless of whether or not they have been erroneously received. Let us assume that the first (i-1) codewords of the jth group have been positively acknowledged, while the ith codeword of the jth codeword group, i.e., the [(j-1)M+i]th codeword, has been negatively acknowledged. Having received the NACK signal, the transmitter immediately enters the retransmission state and sends the ith codeword and the successive M-i codewords of the jth group. When all the M codewords of the jth group have been positively acknowledged, the transmitter enters the transmission state and sends the (j+1)th codeword group. In the SW1, the vectors of a group received after the first block detected in error are discarded whether or not they contain uncorrectable error patterns.

A modified version, designated SW2, has been designed to improve throughput. In the SW2, a buffer is introduced at the receiver side so that codewords of a group received without error or with detectable error patterns can be stored. This solution seems contradictory, as the SW protocol is generally selected when the simplicity of the structure is the major requisite. However, since the buffer required in the SW2 protocol is meant to store M codewords and since M can be set low in relation to N, its implementation complexity can be kept to acceptable levels for numerous real applications, whereas, the SR, which requires a buffer length at least equal to N, is of notable implementation complexity for high values of N. We can now consider the transmission of the jth codeword group. The transmitter sends the M_1=M codewords and enters the wait state, while the receiver decodes all the M vectors and sends the transmitter the ACK or NACK signals for the M codewords. The transmitter remains in the wait state until all the M acknowledgments have been received before entering the retransmission state, during which only the negatively acknowledged codewords are retransmitted. Let us call M_{i-1} the number of negatively acknowledged codewords during the (i-1)th transmission of the jth group ($i \geq 1$), where $M_i \leq M_{i-1}$ and M_1 = M. During the ith transmission of the jth group, the transmitter sends the negatively acknowledged M_i codewords and then enters the wait state, where it remains for N time-slots. The improvement in the SW2 throughput was achieved using a retransmission structure similar to the one proposed by Moeneclaey et al. [4].

In the SW2 protocol, since only the messages detected in error are retransmitted, the number of codewords sent in a transmission cycle varies, with the transmitter sending the same codeword several times consecutively to facilitate correct recovery of the erroneous codeword. The SW2 has been modified to solve this problem. In the modified version, designated SW3, the transmitter waits to receive the sequence of M ACK or NACK signals sent in the previous transmission cycle before choosing a retransmission strategy. Let us denote the number of codewords negatively acknowledged during the (j-1)th transmission cycle of a codeword group by $w^{(j-1)}$. Obviously, $0 \leq w^{(j-1)} \leq w^{(j-2)}$ and $w^{(1)}$=M. If $w^{(j-1)} > 0$, then at least one codeword has been negatively acknowledged in all the (j-1) transmissions of the actual codeword group. We can define $a^{(j)} = [M/w^{(j-1)}]$, where [x] denotes the largest integer lower than or equal to x and $b^{(j)}=M-a^{(j)}w^{(j-1)}$. The receiver transmits the first $b^{(j)}$ blocks $a^{(j)} + 1$ times consecutively, while the remaining $w^{(j-1)}$- $b^{(j)}$ negatively acknowledged codewords of the group are transmitted $a^{(j)}$ times consecutively. In this way, the recovery of the codewords is generally more probable, because consecutive copies of the same message are sent during each transmission cycle.

A simpler protocol, denoted SW4, can be defined by retransmitting each codeword negatively acknowledged $m \geq 2$ times consecutively. In this case, the transmission length (the number of codewords in a transmission cycle) can vary.

2.2 Modified SW protocols with memory

One way of enhancing performance is to introduce a suitable memory at the receiver side and then perform a soft detection of each received symbol [5-6]. The proposed SWi (i = 1, 2, 3, 4) protocols presented in Section 2.1 have been modified by introducing memory at the receiver side, as described in this section; the resulting protocols are called MSWi (i = 1, 2, 3, 4).

The qth message ($q \geq 1$), consisting of k information symbols, is encoded in a codeword $\underline{c}_q$ of n symbols through a code C of type (n,k). The code is able to correct $t \geq 0$ errors and detect $s > t$ errors. The received vector corresponding to the jth transmission of $\underline{c}_q$ is denoted by $\underline{r}_q(j)$. Before transmission, each symbol c_i for $1 \leq i \leq n$ is sent to a modulator, which generates the waveform $s_i(t)$ in the interval [(i - 1)T,iT]. Signal $s_i(t)$ depends on the modulation scheme, which here is assumed binary frequency keying (FSK), although the method can easily be extended to any other type of modulation. Let us define f_o the carrier frequency, and f_1 the frequency deviation.

Assuming a noncoherent detection of the FSK signal due to fading in the communication channel, we shall consider the jth transmission of codeword $\underline{c}_q$ for $j \geq 1$. With $R_{j,i}(1)$ and $R_{j,i}(2)$ denoting the signals at the filter outputs centered on frequencies f_o - f_1 and $f_o + f_1$ respectively, the decision variable for the ith symbol during the jth transmission of $\underline{c}_q$ is $R_{j,i} = R_{j,i}(1) - R_{j,i}(2)$. Signal $R_{j,i}$ for $1 \leq i \leq n$ is sent to a quantizer with n_L positive and n_L negative quantization intervals. A weight, μ_m, is associated with the mth quantization interval for $1 \leq m \leq n_L$. We also assume that $\mu_m = \mu_{-m}$. Using $\tau_{j,i}$ to indicate the weight associated with the ith symbol of the jth received vector, the result is obviously $\tau_{j,i} = \mu_m$ if $R_{j,i}$ lies in the mth quantization interval. The receiver forms a cumulative report $\underline{z}_j$, whose ith component is defined as

$$z_{j,i} = z_{j-1,i} + sign(R_{j,i})\tau_{j,i} \tag{2}$$

with $z_{0,i} = 0$. Once vector $\underline{z}_j$ has been determined, the receiver computes a vector $\underline{w} = \{w_i\}$ for $1 \leq i \leq n$ with, as the ith component w_i, the hard detection of $z_{j,i}$ defined as

$$\begin{array}{l} w_i = 1 \quad if \quad z_{j,i} > 0 \\ w_i = 0 \quad if \quad z_{j,i} < 0 \end{array} \tag{3}$$

The decoder analyzes the vector $\underline{w}$ associated with the jth codeword at the buffer top. If $\underline{w}$ contains no errors or a correctable error pattern, the jth codeword is delivered to the user, and an ACK is sent to the transmitter. Otherwise, a NACK is sent through the feedback channel and the cumulative report (2) is stored and used when retransmissions of the same message are received.

3 RESULTS AND COMPARISONS

The performance of an ARQ scheme is generally characterized by the bit error probability p_b at the decoder output and the throughput T. As the bit error probability of the proposed schemes is the same as that of the classical ARQs without memory and lower when the MSWi protocols with memory are used, this parameter will not be analyzed.

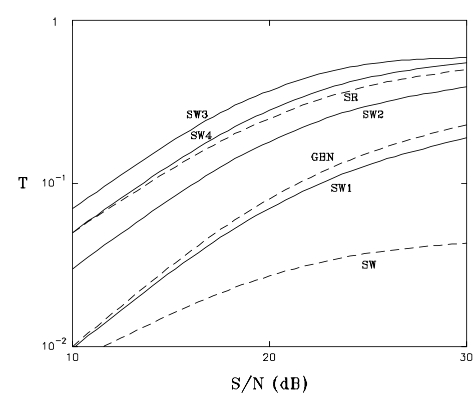

Figure 1: Throughputs of the proposed SW protocols and of classical schemes versus the signal-to-noise ratio for N = 50 and M = 10

Throughput T is defined as the ratio between the number of information symbols transmitted in time τ

and the number of information symbols transmittable in the same time by an uncoded system. The throughput is strongly dependent on the type of ARQ protocol.

Although the SR protocol is the most efficient, its implementation requires a theoretically infinite buffer at the receiver side. With a finite buffer, the throughput decreases to lower than that achievable with an infinite buffer, especially for the high and medium error rates common in mobile communication systems. We shall assume that code C is a perfect error-detecting code, i.e., that it is capable of detecting all the error patterns introduced by the transmission channel. This assumption enables us to determine a lower throughput bound than if real codes are used. A type (255, 225) code is assumed throughout. The mobile channel for the simulation of the ARQ throughputs has been modeled according to the method described by Smith [7].

Figure 1 shows the throughput versus the signal-to-noise ratio for the modified SW protocols for N = 50 and M = 10 and compares them to the throughputs of some classical ARQ protocols. The SW2 clearly shows a higher throughput than the GBN. Figure 1 shows also a surprising result. The SW3 and SW4 protocols present higher throughput values than the classical SR scheme with infinite buffer. This result is a consequence of the fading characteristics. In all the continuous transmission protocols, such as the SR scheme, the fading, being a process with memory, can span many consecutive codewords. In the SW protocols the number of consecutive codewords affected by a fading occurrence is often lower with respect to the continuous schemes, because of the presence of the waiting periods. Therefore, the average number of consecutive codewords affected by a fading occurrence is lower than in the SR protocol. Although the performance of the two protocols SW3 and SW4 is very similar, the experimental results show that SW3 outperforms SW4 for high signal-to-noise ratios and high values of M.

The use of memory affords further performance benefits. Figure 2 shows the throughputs of the MSW1 protocol versus the signal-to-noise ratio in the case N = 10 and M = 10. The behaviours of MSW1 and MSR1 protocols are similar for M = 10, with no buffer required in the case of the MSW1 scheme. Similar results have been obtained for N = 50 and for the protocols MSW2, MSW3 and MSW4.

4 CONCLUSIONS

The stop-and-wait protocols described in this paper are structured to reduce the time the transmitter spends in the wait state for the purpose of improving throughput to the levels of more complex algorithms. The protocols have been successfully applied to mobile communication systems where simple implementation is a prime requisite. Their performance has been derived by computer simulation of a channel model for mobile communications. The resulting performance is comparable to those of more complex ARQ protocols such as the GBN and SR protocols.

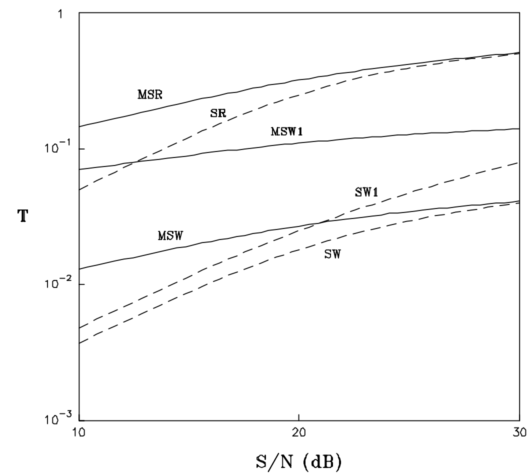

Figure 2: Throughput performance of some protocols with memory for N = 10 and M = 10

5 REFERENCES

[1] W.C.Y. Lee, *Mobile Communications Engineering*, McGraw-Hill, New York, NY., 1982.
[2] S.Lin, D. Costello, Jr., and M.J. Miller, "Automatic-Repeat-Request Error-Control Schemes," *IEEE Communications*, Vol. 22, pp. 5-17, December 1984.
[3] R. A. Comroe and D.J. Costello Jr., "ARQ Schemes for Data Transmission in Mobile Radio Systems," *IEEE Transactions on Selected Areas in Communications*, Vol. SAC-2, pp. 472-481, July 1984.
[4] M. Moeneclaey, H. Bruneel, I. Bruyland and D.Y. Chung, "Throughput Optimization for a Generalized Stop-and-Wait ARQ Protocol," *IEEE Transactions on Communications*, Vol. COM-34, pp.205-207, February 1986.
[5] G. Benelli, "An ARQ Scheme with Memory and Soft Error Detectors," *IEEE Transactions on Communications*, Vol. COM-33, pp. 285-288, March 1985.
[6] C. Lau and C. Leung, "*Performance Analysis of a Memory ARQ Scheme with Soft Decision Detectors*," IEEE Transactions on Communications, Vol. COM-34, p. 827832, August 1986.
[7] J.I. Smith, "A Computer Generated Multipath Fading Simulation for Mobile Radio," *IEEE Transactions on Vehicular Technology*, Vol. VT-24, August 1975.