Throughput analysis of ARQ Selective-Repeat protocol with time diversity in Markov channels*

Michele Zorzi† and Ramesh R. Rao‡
†Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy
‡Department of Electrical and Computer Engineering, University of California San Diego, USA

Abstract—In this paper, an ARQ selective repeat protocol with time-out mechanism is studied. Transmissions on both the forward and the reverse channels are assumed to experience Markovian errors and therefore the feedback is unreliable. Using renewal theory, exact results for the throughput of the protocol are evaluated. Simulation results, that confirm the analysis, are presented; an application to mobile radio is also discussed.

I. INTRODUCTION

Most communication channels present some impairments. System design often involves modification of the characteristics of the physical channel to satisfy the desired quality of service. Automatic Repeat reQuest (ARQ) techniques achieve reliability by using an error-detection scheme, which is less wasteful of capacity, coupled with a block retransmission scheme, which is more wasteful of capacity [1]. Many variants of ARQ schemes exist. Prominent among them are Go-back-N (GBN) and Selective Repeat (SR). Qualitatively, the former is simpler to implement but possibly wasteful of transmission resources. On the other hand, the latter may require a large buffer space to be set aside at the receiver. Since transmission resources are scarce in mobile radio systems, we are motivated to study SR ARQ schemes for the fading wireless channel.

When ARQ schemes were first studied in the literature, channel errors were assumed to be independent from slot to slot and the feedback channel was assumed to be error free. In the mobile radio channel, errors are likely to be clustered and to occur in bursts on both the forward and the reverse channels. Therefore, Markov models for the channel (the so-called Gilbert model for a burst-noise channel [2]) have to be used.

Kanal and Sastry [3] reviewed the dependent channel, but focused more on FEC techniques. Towsley [4] considered a Markov forward channel model, and focused on queueing performance. A generalized Markov model of higher order was considered in [5], in which the effect of unreliable feedback is not included but all three classic ARQ techniques, namely Stop-and-Wait (SW), GBN and SR, are studied. Kim and Un appear to be the first to study the effect of a Markov model for the feedback channel as well [6], combining the effects of dependent transmissions and unreliable feedback. Similar results are found for some more complex protocols, which include control mechanisms to resolve uncertainties that may arise when feedback information is lost [7]. In this last paper, for analytical convenience, some rules of the actual protocols were disregarded in the analysis. On the other hand, an exact throughput analysis of GBN in these conditions has been carried out by Zorzi and Rao [8].

In this paper, we consider the throughput efficiency of the SR ARQ protocol with timer-control operating over a Markov channel. The Markovian channel assumption is motivated by the observation that fading on wireless channels results in outages that are correlated and not IID. The focus on selective repeat ARQ scheme is motivated by its power efficiency. In the mobile wireless environment the need to conserve battery power can not be over emphasized. The incorporation of time diversity further conserves power and may be entirely satisfactory for data applications. An exact performance analysis technique based on the theory of renewal reward processes and of discrete-time Markov processes is introduced. Simulations and comparisons are also presented.

The paper is organized as follows. In Section II, the channel model is presented, and in Section III the protocol is described. The performance analysis is presented in Sections IV, and in Section V some numerical results are presented.

II. CHANNEL MODEL

In this paper, the channel is modeled, as in [7], i.e., as a Gilbert channel in each direction. Thus the patterns of packet and feedback errors follow two independent first-order Markov models, which are described by the transition matrices

$$M_F = \begin{pmatrix} p & q \\ r & s \end{pmatrix}, \quad M_B = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

(1)

where $p = 1 - q$ ($r = 1 - s$) is the probability that the forward slot $i$ is successful given that the forward slot $i - 1$ was successful (unsuccessful), and similarly for the entries of $M_B$, with reference to the backward channel. Note that $1/r$ and $1/c$ represent the average lengths of the bursts of errors, which are described by geometric r.v.'s.

We assume, in the following, that the packet length is a constant, equal to one time unit. The round-trip delay from the beginning of a transmission to the reception and decoding of the corresponding feedback information is $m$ slots. Hence if a packet transmitted in slot $i$ is negatively acknowledged, it will be retransmitted in slot $i + m$. Positive (ACK) and negative (NAK) acknowledgements can never be confused with each other, i.e., the effect of backward errors is to map the ACK and NAK symbols to an Erasure symbol. Also, we assume that each ACK/NAK carries the information about all previous transmissions. This implies that a packet whose ACK is lost may be subsequently acknowledged by feedback received in the future.

III. PROTOCOL DESCRIPTION

In the following we consider a generalization of the standard version of the Selective Repeat ARQ strategy [1]. The transmitter transmits blocks, whose reception is acknowledged by the receiver, by means of a feedback message that can be an ACK

*This work has been partially supported by CNR and MURST, Italy.
(correct reception) or a NAK (erroneous reception). Blocks are kept in a buffer at the transmitter, and removed from it when acknowledged. NAKs trigger retransmission of the corresponding blocks. It is assumed that under correct operating conditions the feedback arrives at the transmitter \( m \) slots (round-trip delay) after the block transmission started. In the classic version of SR, the absence of feedback is equivalent to a NAK, and the block is immediately retransmitted. In the presence of feedback errors, however, waiting for future feedback information may allow the recovery of some lost acknowledgements, thereby improving the performance.

In order to exploit this, a time-out counter is assigned to each block, and activated when the block is transmitted. When the counter reaches \( t \) slots, the time-out expires, and a retransmission is scheduled in the next slot. The classic version of SR corresponds to \( t = m \); whenever \( t > m \) some time-diversity is introduced, allowing feedback error recovery. In particular, each correct feedback message is assumed to contain ACKs and NAKs about all past blocks (for example, it could identify all erroneous outstanding blocks). Upon the correct reception of such a feedback message, the erroneous blocks are rescheduled for retransmission, and the correct ones are removed from the transmitter memory.

It may be observed that this mode of operation requires some buffer capacity at the receiver. Once a block is in error, all successive blocks must be retained in memory, until the original block is correctly received. This is the only way to ensure that blocks are released in order to the upper layers. Therefore, the ordering of packets is achieved at the price of large memory and possibly long delays. To reduce this effect, some modified schemes have been proposed [1, 9, 10]. Here, we are interested in the basic protocol (which always gives an upper bound to the actual performance), and therefore we assume an infinite buffer length. Also, in order to evaluate the throughput efficiency of the system, an infinite supply of packets is assumed at the transmitter.

The transmitter operates as follows. Consider a register with \( t \) positions, numbered from 1 (leftmost position) to \( t \) (rightmost). The register contains all packets that have been transmitted but about whose reception the transmitter has not yet made a decision. We call these packets outstanding packets. Some of these packets (those in positions 1 to \( m \)) are outstanding because a round-trip delay time has not expired. The others (which may occupy the rest of the register and may be in random number) are outstanding because their feedback was lost, and are being held in waiting in the hope that some future feedback message will acknowledge them as well. Such packets will be referred to as pending. The lower positions are occupied by the more recently transmitted packets. (Note that this does not mean that packets are ordered "by age", since some packets might require retransmission.)

The register operates as follows:

- the current transmission is placed in position 1;
- the transmission whose feedback is expected to return at the end of the current slot is in position \( m \) (note that positions 1 to \( m \) are always full, since we assume an infinite supply of packets);
- positions \( m + 1 \) to \( t \) are for pending packets, i.e., packets which did not receive the due ACK and whose time-out has not expired yet;
- after each transmission, a feedback message about the packet in position \( m \) (and about all pending packets) is expected. It may or may not be received depending on the return channel status;
- upon reception of correct feedback, positions \( m \) to \( t \) are emptied: packets which were successfully transmitted exit the system as correctly received, whereas packets in error are put at the head of the input queue for retransmission;
- after the sampling instant, and after the appropriate actions are taken, everything is shifted to the right; if, prior to this, a packet was in position \( t \), a shift to the right means that its timer has expired, and it is retransmitted in the following slot; otherwise, if position \( t \) was empty (i.e., there were less than \( t \) outstanding packets), the first packet of the transmission queue is transmitted: this corresponds to the retransmission of an errored packet or, if there is none, to a new packet entering the system.

IV. EXACT MARKOV MODEL

Based on the above representation of the protocol by means of a shift register we can define a state whose evolution is Markovian. The system is sampled once every slot, right before the feedback message arrives (or its absence becomes apparent). One component of the state describes the occupancy of the right side of the buffer (right buffer, i.e., positions \( m + 1 \) to \( t \)). In particular, we need know how many places are full (i.e., how many packets are pending). We also need to track whether or not the forward transmission of each pending packet was correct, since the retransmission of packets is selectively requested. Note that in GBN, when a packet is retransmitted, all successive packets are retransmitted as well, so that only the number of pending packets is needed [8]. If \( n \) positions in the right buffer are full, we need \( 2^n \) states, for \( n = 1 \) to \( \Delta = t - m \). Note that \( n \geq 1 \) implies that some past feedback was in error.

Also, position \( m + 1 \) of the shift register keeps track of the success or failure of the last examined forward transmission which, due to the Markovian nature of the channel, has an exponentially distributed time of occurrence (success rate \( 2^{-\alpha} \)).

The set of states described above is adequate for throughput evaluation, since it tracks the state of the channels and the number of pending packets which have been correctly received, which is all we need to determine how many packets are acknowledged by each correctly received feedback message. The packet acknowledgements are the rewards in a renewal reward model of the chain, which allows an exact study of the throughput efficiency of the protocol [8, 11, 12].

A. State representation

As mentioned above, the states must keep track of the number of occupied positions in the right-hand side of the register. Let \( n \) denote a binary variable, \( a_i = 1 \) indicates a success and \( a_i = 0 \) a failure. The contents of the right register can be represented by a vector \( (a_1, a_2, \ldots, a_n) \), where \( a_i \) is associated to the oldest transmission.
receives a correct feedback, and decides about all pending packets. A concise representation of the binary vector, useful when describing the transition structure, is given by its "decimal value", defined as

\[ a = \sum_{i=1}^{n} a_i 2^{i-1}. \]  

(3)

With this notation, all states with \( n \geq 1 \) are uniquely determined by the couple \((n, a)\), \( n = 1, \ldots, \Delta \), \( a = 0, \ldots, 2^n - 1 \). For \( n = 0 \), the above vector cannot be defined; as already mentioned, this state must be subdivided into states 0\( S \) and 0\( F \), in order to retain memory of the last forward outcome: states 0\( S \) and 0\( F \) correspond to a successful feedback transmission (i.e., the register is empty), and to a successful or erroneous forward transmission, respectively.

### B. Transitions and rewards

Let the notation \( X_1 / X_2 \) indicate that \( X_1 \) is the outcome of the forward transmission and \( X_2 \) that of the feedback (with \( X_2 \) being Correct or Erroneous). Assume that the system be in state \((n, a)\); if the feedback is correct, the right register is emptied, i.e., \( n = 0 \), and the forward transmission outcome determines which one of the zero states (0\( S \) or 0\( F \)) is entered. If the feedback is erroneous, the number of pending packets is increased by one, everything is shifted to the right and a new block enters the register. On the other hand, if \( n = \Delta \), the time-out is reached, and \( n \) cannot be increased. It is easy to see that the transitions obey the following rules:

- if C/C, always transition to state 0\( S \);
- if E/F, always transition to state 0\( F \);
- if C/E and 0 < \( n \) < \( \Delta \), then transition to state \((n+1, 2^n + a)\); i.e., \( a_{n+1} = 1 \) is added;
- if E/E and 0 < \( n \) < \( \Delta \), then transition to state \((n+1, a)\); i.e., \( a_{n+1} = 0 \) and the value of \( a \) is not changed;
- if \( n = 0 \), then C/E and E/E will lead to \((1,1)\) and \((1,0)\), respectively;
- if \( X_1/E \) and \( n = \Delta \), \( n \) remains the same, and \( a_1 \) is lost; if C/E, then transition to state \((\Delta, 2^{\Delta-1} + [a/2])\), if E/E, then transition to state \((\Delta, [a/2])\).

The transition probabilities, i.e., the probabilities of the four couples \( X_1 / X_2 \), are easily found from the Markovian structure of the success/failure processes on the two channels. The memory of the past values of \( X_1 / X_2 \) is contained in the states introduced above, and can be shown to be as follows:

- state 0\( S \) corresponds to C/C;
- state 0\( F \) corresponds to C/E;
- states \((n, a)\), \( n \geq 1 \), correspond to C/E if \( a \geq 2^{n-1} \) (i.e., \( a_n = 1 \)), and to E/E otherwise.

As to the rewards, they are earned every time the transmitter receives a correct feedback, and decides about all pending packets. If \( n_s \) is the number of successful pending packets, the transitions to states 0\( S \) and 0\( F \) involve \( n_s + 1 \) and \( n_s \) rewards, respectively. For all other transition, no rewards are earned.

If \( R_{ij} \) and \( D_{ij} \) are respectively the number of rewards and the number of slots associated to a transition from \( i \) to \( j \), and \( R_i \) and \( D_i \) are the averages over the destination, \( j \), a fundamental theorem of renewal reward processes allows us to write [11]:

\[ \eta = \lim_{\tau \to \infty} \frac{R(\tau)}{\tau} = \sum_{i=1}^{N} \pi_i R_i - \sum_{i=1}^{N} \pi_i D_i, \]

(5)

where the last passage is due to the fact that all transitions take one slot, and therefore \( D_i = 1 \) \( \forall \ i \). \( \pi \) is the asymptotic distribution of the Markovian chain, and \( \eta \) is the steady-state (i.e., average, if ergodicity is assumed) throughput efficiency.

As an example, consider the case in which \( \Delta = 2 \) [13, 14]. The number of states in this case is \( N = 8 \). The state space is represented by the vector

\[
\begin{pmatrix}
0 & 0 & 0 & 1 & 1 & 2 & 2 & 2
\end{pmatrix},
\]

and the transition matrix is given by

\[
P = \begin{pmatrix}
a & a & b & s & 0 & 0 & 0 & 0 \\
q & a & p & b & 0 & 0 & 0 & 0 \\
c & s & r & 0 & 0 & ds & 0 & dr \\
c & q & c & p & 0 & 0 & dq & 0 \\
c & s & r & 0 & 0 & ds & 0 & dr \\
c & q & c & p & 0 & 0 & dq & 0 \\
c & q & c & p & 0 & 0 & dq & 0
\end{pmatrix}.
\]

(7)

Finally, the non-zero elements of the reward matrix are

\[
R_{11} = R_{21} = R_{31} = R_{51} = R_{42} = R_{52} = R_{72} = 1;
\]

\[
R_{41} = R_{61} = R_{71} = R_{81} = 2; \quad R_{43} = 3.
\]

and the throughput can be computed according to Eq. (5).

### C. Special cases

Two interesting cases that arise are \( \Delta = \infty \) and \( \Delta = 0 \). Note that, if infinite time diversity were available, the unreliability of the feedback channel could be overcome since, excluding pathological situations, sooner or later a correct feedback will be received. The corresponding performance can therefore be computed as in the case of perfect feedback channel, and is found to be

\[ \eta = 1 - \varepsilon, \]

(10)

where \( \varepsilon \) is the marginal probability that a block is in error, and is equal to

\[ \varepsilon = 1 - \frac{r}{1 - p + r}. \]

(11)

Eq. (10) is identical to the result found in the presence of iid block errors, as reported in [5]. It is obvious that this solution is not practical, since it requires an infinite transmission buffer. However, it is to be regarded as an upper bound, which is very tight as soon as \( \Delta \) is large enough with respect to the average duration of a burst of errors, \( 1/r \).

For the case \( \Delta = 0 \), from the analysis in [7] it is not difficult to show that, in general

\[
\eta = \frac{r(m)}{g(m) + r(m)} \cdot \frac{c(m)}{b(m) + c(m)} = (1 - \varepsilon_f)(1 - \varepsilon_b),
\]

(12)

where the error rates of the two channels, \( \varepsilon_f \) and \( \varepsilon_b \), could be different.
D. Computational complexity of the exact approach

It is clear that the exact approach is very demanding in terms of computational complexity. This is due to the fact that the success/failures of all pending packets must be tracked, giving rise to an exponential complexity, instead of the polynomial complexity required by the same approach when applied to GBN. Therefore, as soon as the amount of time diversity becomes significant, the analysis becomes burdensome. In numerical terms, tracking about 2000 states (i.e., for $\Delta \leq 10$ for the throughput analysis) may represent a working maximum, even though, with more elaborate techniques, it is possible to solve chains with more states. However, the value of an approximate approach with a reduced size of the state space is clear. A bounding technique, which reduces the size of the state space from $2^{\Delta+1}$ to about $\Delta^3$, is proposed in [15].

V. NUMERICAL RESULTS

In this section, we present results assuming that the forward and backward channels have the same parameters.

In Fig. 1, the throughput efficiency is plotted vs. the marginal block error rate, $\epsilon$, for some values of $r$ and $\Delta$: the results of the exact analysis (solid lines) are reported along with some simulation points ($\circ$). It can be seen that the difference in performance obtained using different values of $\Delta$ is significant for high error rates. Also, as expected, for $\Delta > 1/r$, the degree of time diversity effectively combats the unreliability of the feedback: this can be seen from Fig. 1b, where for $r = 0.3$ the performance obtained for $\Delta = 5$ is close to that for $\Delta = 0$ (i.e., perfect feedback). On the other hand, when $1/r$ is significantly larger than $\Delta$, feedback errors severely degrade the performance (see Fig. 1a). The throughput performance vs. $\Delta$ is plotted in Fig. 2. Due to the computational complexity, the exact approach can be used only for small $\Delta$.

A. Application to mobile radio

There is evidence that packet transmission on the fading mobile radio channel can be approximated by means of a Markov model as the one discussed in the above [16]. Analytical models or simulations can be used to determine the transition probabilities for a given Doppler frequency $f_D$ and packet duration $T$. In fact, the correlation between two channel samples at distance $T$ depends only on the product $f_D T$ [17]. Therefore, the above model can be used to analyze systems such
In this paper, we study a Selective Repeat ARQ scheme which exploits time-diversity to recover from feedback errors. A dependent structure for the error processes on the two channels is assumed, and modeled as a Markov chain. This allows us to compute exactly the throughput performance. The methodology enables us to study the protocol performance as function of the average channel error rate and of the amount of time diversity. An application to mobile radio, where the successes and failures on the fading channel can be accurately modeled as a Markov chain, is studied as well, and the accuracy of the Markov model is assessed.

Further directions of research involve the application of the presented technique to more elaborate ARQ protocols, and possibly to hybrid techniques. Also, a delay analysis, which would require considering queuing and re-ordering delays, would be very useful. Third, the specific features of the mobile radio environment suggest that some better protocols (e.g., as to power efficiency) could be devised.

REFERENCES