Fast recovery selective repeat- ARQ protocol

Sayed Abd El Hadi Nouh
Computer and Systems Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt

In order to ensure high reliability for data communications systems, it is necessary to resort to an error control method that eliminates the transmission errors caused by the channel noise so that error-free data can be delivered to user. One of the most popular methods for error control is the Automatic Repeat reQuest (ARQ). It is widely used in data communications systems because it is simple and provides high reliability. This paper presents a new procedure for handling retransmission in a selective- repeat ARQ strategy named Fast-Recovery Selective- Repeat (FRSR) which provides a very fast error recovery mechanism. Unlike other selective repeat strategies, the proposed procedure ensures that a correctly received packet will not be retransmitted. In addition, this protocol is simple enough that it places little computational load on the transmit and receive processors. Analysis of this protocol indicates that it yields a throughput close to that one of an ideal selective-repeat protocol if its parameters are adjusted properly. The analysis is performed by means of analytical methods. Analytical results are used to study the performance measures through numerical examples. The results provide insight into the interaction between the protocol parameters and identify the operational characteristics.

لضمان الاعتماد على نظم الاتصالات الرقمية بشكل امن في نقل البيانات، فانه من الضروري من وجود طريقة للتحكم في إزالة الأخطاء التي تصيب حزم البيانات أثناء مرورها في خطوط النقل وذلك حتى نضمن سلامة وصحة وصول البيانات إلى المستخدم النهائي. يقدم هذا البحث بروتوكول جديد يعمل على ضمان وصول حزم البيانات، من خلال شبكات الاتصالات الرقمية، إلى المستخدم النهائي بشكل صحيح وامن. هذا ويتميز البروتوكول المقترح بالاتي: يضمن هذا البروتوكول إعادة إرسال حزم البيانات الخاطئة فقط من قبل المرسل، ولا يسمح بإعادة إرسال حزم بيانات تم أستقبلها صحيحة بدون أخطاء. يقوم البروتوكول المقترح باكتشاف وإعادة إرسال حزم البيانات الخاطئة بطريقة سريعة مما يؤدى إلى تحسين كفاءة نقل البيانات على الشبكة. مراعاة البساطة عند تصميم البروتوكول لدرجة أن حجم المهام المطلوب إنجازها من قبل المرسل والمستقبل صغير ويتطلب زمن قليل. لقد تم تحليل وتقييم أداء البروتوكول المقترح بالطرق التحليلية، وكذلك تم استخدام نتائج التحليل في دراسة مقاييس الأداء للبروتوكول وذلك من خلال أمثلة قياسية تم توضيحها بالرسم. كذلك تم عمل مقارنة أداء البروتوكول بأداء البروتوكولين المثاليين. ولقد وجد أن أداء البروتوكول المقترح يماثل إلي حد كبير جدا أداء البروتوكول المثالي (ISR).

Keywords: ARQ, Selective repeat protocol, Performance, Throughput, Delay.

1. Introduction

As long as the ideal error- free channel has not been realized, data communication systems are obliged to possess the capability of handling transmission errors. The corresponding techniques which have been proposed fall in one of two major categories, namely automatic repeat request (ARQ) and forward error control (FEC) schemes [1-5]. Hybrid strategies are also known [6- 8]. In ARQ systems, erroneous data are retransmitted. In FEC systems, forward error correcting codes are used.

Various forms of ARQ schemes are used for the control of errors over a noisy channel in computer networks. Basically they can be classified into two categories; the stop and wait (SAW) scheme for which the channel is idle from time to time, and the continuous scheme for which message blocks are sent from the transmitter to the receiver continuously.

In the stop and wait scheme, the transmitter sends a packet to the receiver and waits for an acknowledgment from the receiver. If the packet is received correctly, the receiver sends a positive acknowledgment (ACK) and the transmitter sends the next packet. If a negative acknowledgment (NAK) is received, the sender retransmits the packet and again waits for an acknowledgment. Retransmissions continue until the transmitter receives an ACK. This scheme is quite simple but inherently inefficient because of the idle time spent in waiting for an

Alexandria Engineering Journal, Vol. 39 (2000), No. 6, 857-869

acknowledgment of each transmitted packet. Several studies show that, for data communications systems with high data rates and large round- trip delays, the throughput performance of SAW scheme becomes unacceptable [9-11].

The widespread use of satellites for data communications, coupled with the declining cost of digital hardware encouraged the use of continuous ARQ strategies to replace the SAW procedure. There are two basic approaches for handling errors in the presence of the continuous ARQ scheme. The first one is called the go back N (GBN) protocol, and the other is the selective- repeat (SR) protocol.

In the go back N scheme, the transmitter continuously transmits packets in order and then stores them pending receipt of ACK/NAK for each. Whenever the transmitter receives a NAK indicating that a particular packet was received in error, it stops transmitting new packets. Then it goes back to the negatively acknowledged packet and proceeds to retransmit that packet and all the packets following it. This scheme is more efficient than the SAW protocol when the transmission error rate is not too high and the link propagation delay is small. On the other hand, the GBN scheme becomes quite ineffective for communications systems with high data rate and large round-trip delays. This ineffectiveness is caused by the retransmission of error- free packets following a packet detected in error [12-14]. This can be overcome by using the selective-repeat (SR)- ARQ protocol.

In the SR- ARQ protocol, packets are also transmitted continuously. However, the transmitter only resends those packets that are negatively acknowledged (NAK'ed). After re-sending a NAK'ed packet, the transmitter continues transmitting new packets in the transmitter buffer. Therefor, this is more efficient than other ARQ schemes. With this scheme, a buffer must be provided at the receiver to store the error- free packets following a received packet detected in error, because, ordinarily packets must be delivered to the end user in correct order.

Several modifications of the basic selective repeat strategy have been proposed in the literature [15-20]. However, these schemes either lead to slow error recovery or they do not provide adequate mechanism to ensure that correctly received packets will not be retransmitted. In this paper, a novel procedure for handling retransmissions in selective- repeat ARQ system, named Fast-Recovery Selective-Repeat (FRSR) is presented and analyzed. The proposed approach provides a very fast error recovery mechanism and ensures that, correctly received packets will not be retransmitted (i.e., it remedies the problem of duplication). Compared with the earlier ARQ techniques, this strategy has superior performance. In addition it is simple enough that it places little computational load on the transmit and receive processors.

The organization of the remaining part of the paper is as follows: Section 2 describes the new SR- ARQ strategy. Analysis of the throughput for the new approach is presented in Section 3. In section 4, a study of the performance measures of the proposed scheme through numerical examples is presented. An expression for the delay performance of the new scheme is obtained in Section 5. Finally, Section 6 concludes the paper.

2. The FRSR strategy

The protocol entities under consideration consists of a primary process (P), at the transmitter, and a secondary process (S), at the receiver, that can communicate by exchanging messages on an asynchronous channel.

For handling reliable information flow, the concerned protocol includes two types of frames, collectively called the protocol data units (PDUs): Information frames (I- frames) and Estate- Record frames (ER- frames).

I- frames are used to convey user data and ER- frames are used to report the corrupted or lost I- frames, and carry no data. Fig. 1 shows the frame format for both the I- frame and the ER- frame. A description of each field in both frames is as follow:

- Flag Indicates beginning or end of frame,
- Control information such as the frame type, the address, etc.,
- S(N) Indicates the send sequence number of the transmitted I-frame. $0 \le S(N) \le M-1$,

where *M -1* is the maximum sequence number and S(N) is modulo *M*,

- ERR It is a one - bit Estate Record Request. It is set to 1 if the transmitter requests an ER- frame; otherwise it is set to 0,
- DATA User data,
- FCS Frame check sequence,
- ES(N) Expected sequence number of the next I- frame to be received,
- L_i Sequence number of the lost I- frames, where L_1 is the oldest lost I- frame and L_n is the most recent one.

In FRSR scheme, the loss of a transmitted I- frame is detected through a gap in the sequence number. That is, loss of an I- frame is detected when an I- frame is correctly received whose S(N) is greater than the S(N), by more than one, of the last new I- frame correctly received.

Flag	Control	S(N)	ERR	DATA	FCS	Flag

I-Frame Format

Flag	Control	ES(N)	$L_1, L_2, \ldots, L_n$	FCS	Flag

ER-Frame Format

Fig. 1. Protocol data units format.

It is worthwhile to mention that, in FRSR strategy, the two control Frames (ACK & NAK frames) which are used in the earlier SR-schemes, are replaced by one control frame (ER- frame). Moreover, instead of acknowledging each I- frame individually (as in the earlier SR- schemes), the ER- frame acknowledges multiple I- frames and reports the erroneous frames in the same time. As we shall see, this modification enhances the efficiency of the throughput performance, overcomes the problem of duplication and provides fast error recovery mechanism.

Let us now explain how ER- frame acknowledges multiple I- frames and record the erroneous frames in the same time. Reception of an ER- frame at the sending station acknowledges the receipt of all I-frames whose sequence numbers are older than ES(N), if no gap is reported. Otherwise, it acknowledges the receipt of all I- frames whose sequence numbers are older than L_1, between L_i and L_{i+1}, and between L_n and ES(N).

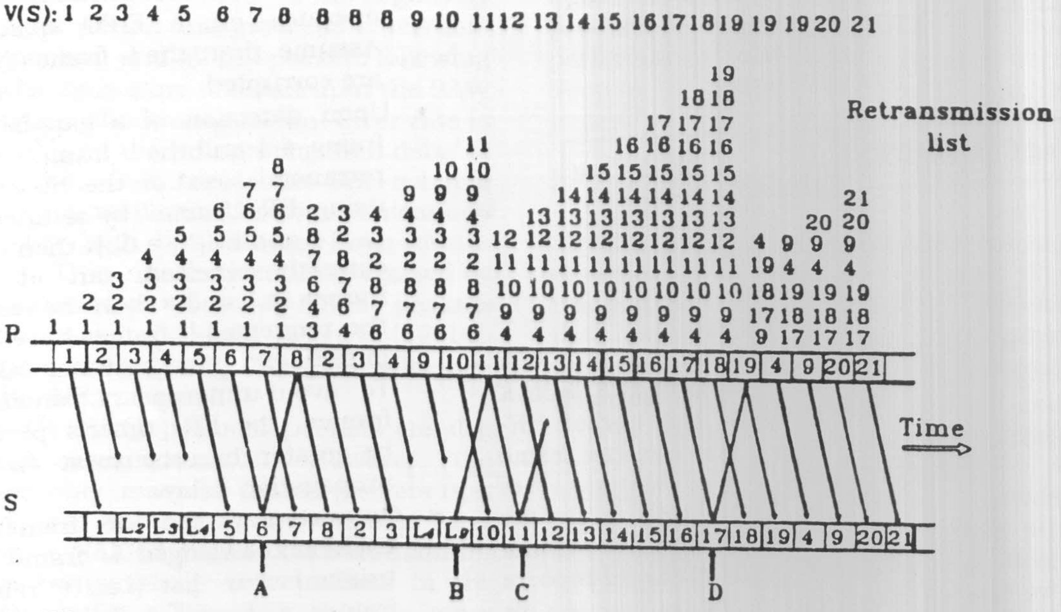
Since each protocol entity is essentially a sequential system. Then the Primary P process must retain a send sequence variable V(S) which indicates the send sequence number N(S) to be allocated to the next I-frame to be transmitted and the Secondary S process must retain a receive sequence variable V(R) which indicates the sequence number of the next I- frame expected to arrive.

The operation of the FRSR protocol can be explained through the frame sequence diagram, shown in Fig. 2, as follows:

- P sends I- frames continuously without waiting for an ER- frame to be returned. Since more than one I- frame have been sent, but have not yet been acknowledged, P must retain a copy of each I- frame transmitted in a retransmission list that operates on a FIFO queue discipline. Assume that the I- frames No. 2, 3, and 4 are corrupted.
- Upon detection of a gap between the I-frame # 1 and the I- frame # 5, S requests retransmission of the missing I- frames using ER- frame: by setting $L_1 = 2$, $L_2 = 3$, $L_3 = 4$ and ES(N) = 6. It then stores the two correctly received, out of sequence, I-frames (# 1 and # 5) in the receive list until the requested I- frames have been retransmitted and starts a timer called ER timer. To avoid unnecessary transmission of ER-frames, the ER- timer's period, t_{ER}, must be greater than the worst- case round- trip propagation delay.
- On receipt of the ER- frame, P removes each acknowledged I- frame from the retransmission list (i.e., P removes the I-frames # 1 and # 5). In addition, P will determine which acknowledged I- frame was transmitted most recently, and will retransmit each lost I- frame that was transmitted prior to this most - recently acknowledged I- frame. That is, a negatively acknowledged I- frame is not retransmitted unless a transmission following it has been acknowledged. Otherwise, the request of retransmission of I- frame is ignored, since the S may have not yet received the previous transmission of this

I- frame. Hence, the FRSR protocol ensures that, a correctly received I- frame will not be retransmitted. Accordingly, since the I- frame # 5 is the most - recently acknowledged, then P retransmits the I- frames # 2, # 3 and # 4 and resumes the transmission of new I- frames. Assume that, the retransmitted I- frame # 4 and the I- frame # 9 are corrupted.

- When the timer expires and still there is a gap in the receive list, it is restarted and a new ER- frame is sent, reporting this gap (i.e. L_1 = 4 and ES(N) = 9).
- On receipt of the ER- frame in which the retransmission of the I- frame # 4 is requested, P ignores this request, because the transmission following the I- frame # 4 (i.e., I- frame #9) have not yet been acknowledged, then S continuous in transmission of new I- frames.
- Upon detection of a new gap, S sends an ER- frame reporting this gap and the previous unrecovered gap(s) and starts the ER timer, if it is not already running. If it is already running, it is reset and restarted. Accordingly, S sends the ER- frame with L_1 = 4, L_2 = 9 and EN(S) =11.
- If the ER- frame is corrupted, S does not detect any new gap(s) and the ER timer is expired. S resends the same previous ER- frame and again restarts the ER- timer. The ER- timer is not stopped unless all lost I- frames are recovered.



A: *ER- timer started, ER- frame(lost:2,3,4; EN(S)=6)sent*

B: *ER- timer timeout, ER- frame(lost:4; EN(S)=9)sent*

C: *ER- timer restarted, ER- frame(lost:4,9; EN(S)=11)sent*

D: *ER- timer timeout, ER- frame(lost:4,9; EN(S)=17)sent*

Fig. 2.Error recovery in FRSR protocol.

In the FRSR protocol, the receive list at the receiving station is essentially a resequencing buffer. The arrived, out of sequence, I- frames are stored until the gap(s), (i.e. lost I- frame(s)), through them is completed and the receiving station can forward ordered I- frames to the end user. Since the resequencing buffer has a finite size. Then, if the transmitter sends

I- frames faster than the receiver can process them, a buffer overrun could occur and the system goes into deadlock. Also, at the sending station, since the retransmission list has a finite size, a limit must be set on the number of the I- frames that can be awaiting acknowledgment. A simple solution is to provide a flow control mechanism through the sliding window protocol, in which a maximum limit is set on the number of I- frames, in the retransmission list, that can be awaiting acknowledgment. This limit is called the send window size (SWS). Also, the maximum number of, out of sequence, I- frames that can be stored in the receive list is called the receive window size (RWS). Like the various versions of the SR- ARQ scheme, in the FRSR protocol, SWS = RSW = W, where W is called the window size. Factors, such as the maximum frame length, the available buffer storage, the link propagation delay and the transmission rate, must be taken into consideration when selecting W. SWS is implemented by the use of two variables:

Upper Window Edge (UWE) and Lower Window Edge (LWE). Where the UWE is incremented by one each time a frame is transmitted and the LWE is incremented by one each time a frame is acknowledged. The difference between the UWE and LWE should never exceed the window size, W. Then, the sender must not transmit any new I- frames if the receiver has not acknowledged the receipt of I- frame whose sequence number is W older than the next new I- frame to be transmitted. This scheme ensures that the receiving station will never receive more than W - 1 out- of-sequence I- frames.

The sender can request an ER- frame from the receiver by setting the ERR bit in the transmitted I- frame. The ERR bit must be set to one, by the sender, in the following cases:

- When the number of unacknowledged I-frames, in the retransmission list, reaches W, the ERR bit is set in the Wth I- frame and the ER- timer is started. The timer will be stopped if the sender receives an ER-frame. If the ER- timer expires, the Wth I-frame, with the ERR bit is set to one, is retransmitted and again the ER- timer is restarted.
- The sending station must also set the ERR bit on the last I- frame to be transmitted. In this case, the ER- timer is also started. The timer will be stopped if the returned ER- frame acknowledges the receipt of all transmitted I- frames. If the timer expires, the last unacknowledged I- frame with ERR bit set to one is retransmitted and the ER- timer is restarted.

3. FRSR throughput

For a tractable mathematical analysis, it is convenient to define the throughput as the number of correctly received user data bits per second. This quantity is often normalized to the channel capacity. The throughput performance of the FRSR protocol is influenced by many factors. That must be taken into consideration through the analysis, such as: the I- frame length, the number of overhead bits in each I- frame, the characteristic of the transmission errors and the behavior of the mechanism that it uses to send I- frames and retransmit lost I- frames.

The model under consideration consists of two stations *A* and *B*, which communicate through a bi-directional link, Fig. 3. Each station operates as a transmitter and a receiver. It is assumed that, Station *A* has an infinite number of I- frames to send to Station *B*.

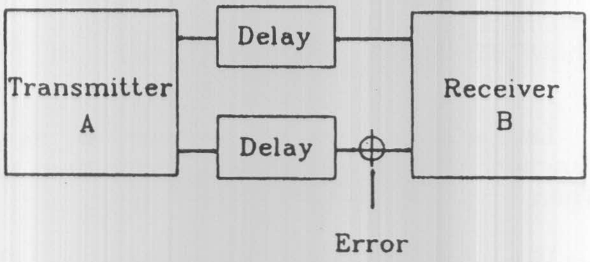


Fig. 3. Communication system model.

The expression of the throughput will be derived based on the concept of average transmission time per I- frame [21], This quantity comprises both the actual transmission time of an I-frame and the duration of recovery actions in case of transmission errors. The essence of the throughput analysis is to consider a typical I-frame called " test frame" (T) and we find the average transmission time for this frame. Since the test frame can arrive at the receiving station in error. Hence, for the test frame to be received correctly, it may have to be transmitted more than once. Assuming that the probability of a frame in error lies between zero and one, then the average number of retransmissions is between zero and infinity.

The notation used to define the protocol parameters is summarized below:

d is the Number of data bits per I- frame,
h is the Number of overhead bits per I-frame,
p_b is the Probability of a bit being in error (independent bit errors); $0 < p_b <? 1$,
C is the Channel capacity in bps,
W is the Window size,
t_p is the Propagation delay in second, (one way link delay),
t_{ER} is the ER-timer's period.

For analysis, several assumptions are made throughout this paper. A noiseless feedback channel is available, i.e., the probability that the ER- frame is in error is neglected and the length of the ER- frame is negligible compared to that of the I- frame. Also, it is assumed that I- frames are of fixed length. In the following we will derive the expression of the throughput achievable by the FRSR technique.

Let t_{ix} denote the transmission time of an I- I. frame. Where,

$$t_{ix} = \frac{h+d}{C}. \tag{1}$$

Let ω denote the window size, W, as a function of the I- frame transmission time, t_{ix}. Then,

$$\omega = W\, t_{ix}. \tag{2}$$

Let p_f denote the probability that an I-frame is in error. Where,

$$p_f = 1 - (1 - p_b)^{h-d}.$$

Before proceeding in the throughput analysis, we note that, the expression of the throughput will be derived through the derivation of the average transmission time of the test I-frame before and after the window is closed.

$$p_f = 1 - (1 - p_b)^{h-d}. \tag{3}$$

If the test frame arrives in error the first time it is transmitted, it requires a second transmission. Let T_1 be the average time that it takes from the first transmission of the test frame until its first retransmission given that the first transmission was unsuccessful, Fig. 4. This average time, as it is shown in Fig. 4, includes the transmission time of the test frame, t_{ix}. The remaining window size is (W - 1) after the test frame is sent. According to FRSR scheme, the time at which the first retransmission of the test frame starts, depends on the arrival time of the ER- frame , at the sending station, which reports the loss of the first transmission of the test frame. On the other hand, the time at which this ER-frame is sent, by the receiver, depends on the number of transmitted I- frames, following the first transmission of the test frame, which are needed before an I-frame arrives at the receiver correctly. Let this needed number is denoted by i , (where i ranges from 1 to W- 1). Then, the probability of this needed number, i, is given by $(1 - p_f)\, p_f^{\,i-1}$. Let $T_1^{(o)}$ denote the average time from the first transmission of the test frame until its first retransmission before the window is closed. This quantity can easily be found using Fig. 5:

$$T_1^{(o)} = (2t_p + t_{ix})(1 - p_f^{W-1}) + t_{ix}\left(\sum_{i=1}^{W-1} i(1-p_f)p_f^{i-1}\right). \tag{4}$$

Also, $T_1^{(c)}$ is the average time from the first transmission of the test frame until its first retransmission after the window is closed. According to FRSR protocol, this quantity is given by:

$$T_1^{(c)} = (\omega + t_{ER})\, p_f^{W-1}. \tag{5}$$

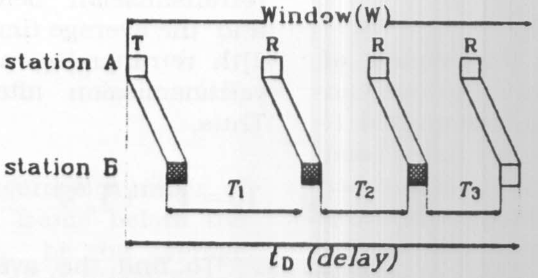


Fig. 4. Average time between successive transmissions and delay of the test packet..

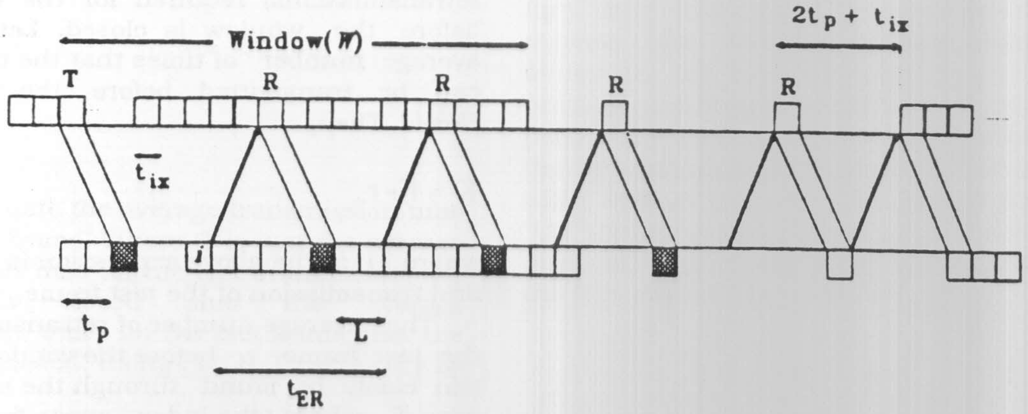


Fig. 5. Timing diagram for throughput analysis.

But T_1 is just the sum of $T_1^{(o)}$ and $T_1^{(c)}$. Thus,

$$
\begin{aligned}
T_1 &= \left(2t_p + t_{ix}\right)\left(1 - p_f^{W-1}\right) \\
&\quad + t_{ix}\left(\sum_{i=1}^{W-1} i\left(1-p_f\right)p_f^{i-1}\right) + \left(\omega + t_{ER}\right)p_f^{W-1} \\
&= 2t_p - 2t_p p_f^{W-1} + t_{ix}\left(\frac{2 - p_f - p_f^{W-1}}{1 - p_f}\right) \\
&\quad + t_{ER}\, p_f^{W-1}.
\end{aligned}
\tag{6}
$$

Let T_n, where $n \geq 2$, be the average time that it takes from the (n -1)th retransmission of the test frame to the nth retransmission given that the (n-1)th retransmission is not successful. To find T_n, we proceed as follows:

Let Y_n denote the number of intervals of length t_{ER} after the (n - 1)th retransmission of the test frame and before the window is closed. Then,

$$
Y_n = \mathrm{Max}\left[0, \left\lceil \frac{\omega - \sum_{i=1}^{n-1} T_i}{t_{ER}} \right\rceil\right]. \tag{7}
$$

Where, $\lceil x \rceil$ is the smallest integer $\geq$ x.

Let R denote the number of I- frames that can be transmitted in a t_{ER} period, where $t_{ER} \geq 2t_p + t_{ix}$. Then,

$$R = \frac{t_{ER}}{t_{ix}} . \tag{8}$$

In FRSR protocol, since the request of retransmission of an erroneous I- frame can not be carried out unless a transmitted I-frame following it in time has been acknowledged. Then, let L denote the number of I- frames that can be sent within the remaining time of a t_{ER} period following a transmission of an ER- frame and a retransmission of the test frame before the window is closed, see Fig. 5. Where

$$L = \frac{t_{ER} - \left(2t_p + t_{ix}\right)}{t_{ix}} . \tag{9}$$

Let $T_n^{(o)}$ denote the average time that it takes from the (n -1)th retransmission of the test frame to the nth retransmission before the window is closed, given that the (n - 1)th retransmission is not successful. From Fig. 5, this quantity can be given by:

$$T_n^{(o)} = \begin{cases} \left(1 - p_f^L\right) t_{ER} + p_f^L \left(1 - p_f^R\right) \\ \times \left(\sum_{i=2}^{Yn} i \left(p_f^R\right)^{i-2} t_{ER} \right) \\ - t_{ix} \left(1 - p_f^L \left(p_f^R\right)^{Yn-1}\right), \text{ if } Y_n \geq 1, \\ 0, \text{ otherwise.} \end{cases} \tag{10}$$

Let $T_n^{(c)}$ denote the average time that it takes from the (n -1)th retransmission of the test frame to the nth retransmission after the window is closed, given that the (n - 1)th retransmission is not successful. From Fig. 5, this quantity can be given by:

$$T_n^{(c)} = \begin{cases} p_f^L \left(p_f^R\right)^{R_n - 1} \left(t_{ER}\left(R_n + 1\right) - t_{ix}\right), \\ \quad \text{if } R_n \geq 1, \\ t_{ER} - t_{ix}, \text{ otherwise.} \end{cases} \tag{11}$$

But T_n, where $n \geq 2$, is just the sum of the average time that it takes from the (n-1)th retransmission of the test frame to the nth retransmission before the window is closed and the average time that it takes from the (n-1)th retransmission of the test frame to nth retransmission after the window is closed. Thus,

$$T_n = T_n^{(o)} + T_n^{(c)} . \tag{12}$$

To find the average transmission time for the test frame, we need to introduce the expected number *K*, which represents the average total number of transmissions (including the first transmission and all retransmissions) required for the test frame before the window is closed. Let *r* be the average number of times that the test frames can be transmitted before the window is closed. Then,

$$K = 1 + r , \tag{13}$$

where '1' in the above expression is due to the first transmission of the test frame.

The average number of retransmissions of the test frame, r, before the window is filled, can easily be found through the summation over T_i, where the index i starts from 1 up to the maximum upper limit at which the total sum is $\leq$ the window size in time ω. This maximum upper limit is r, then,

$$\operatorname*{Max}_{r} \left[\sum_{i=1}^{r} T_i \right] \leq \omega \cdot \tag{14}$$

The probability that K or less than K transmissions of the test frame is needed is given by,

$$\sum_{i=1}^{K} \left(1 - p_f\right) p_f^{i-1} = 1 - p_f^K , \tag{15}$$

where $(1- p_f)\ p_f^{\,i-1}$ is the probability that exactly i transmissions are required.

Hence, the probability that more than K transmissions of the test frame are needed is given by,

$$1-\left(1-p_f^K\right)=p_f^K. \tag{16}$$

The average number of transmissions of the test frame is given by,

$$\sum_{i=1}^{\infty} i\left(1-p_f\right)p_f^{i-1}=\frac{1}{1-p_f}. \tag{17}$$

Now, let β_o be the average number of transmissions of the test frame before the window is closed and β_c be the average number of transmissions of the test frame after the window is closed. Where,

$$\beta_o=\frac{1-p_f^K}{1-p_f}, \tag{18}$$

$$\beta_c=\frac{p_f^K}{1-p_f}. \tag{19}$$

Let T denote the average transmission time for the test frame. Where, T is just the average transmission time for the test frame before the window is closed plus the average transmission time for the test frame after the window is closed. Using Fig. 5, T can easily be given by:

$$T=t_{IX}\beta_o+t_{ER}\beta_c+\left(2t_p+t_{IX}\right)p_f^K. \tag{20}$$

Hence, the normalized throughput γ of the FRSR scheme is given by:

$$\gamma=\frac{d}{CT}. \tag{21}$$

It is worthwhile to show that, the throughput of FRSR scheme with infinite window size is the same as the ideal selective repeat protocol. As $W\to\infty$, so does K, and hence both β_c and p^k_f tend to zero and Eq. (20) will be:

$$T=\frac{t_{IX}}{1-P_f}. \tag{22}$$

Consequently, the normalized throughput γ is given by:

$$\begin{aligned}\gamma&=\left(\frac{d}{C\,t_{IX}}\right)\left(1-p_f\right)\\&=\left(\frac{d}{d+h}\right)\left(1-p_f\right).\end{aligned} \tag{23}$$

Eq. (23) is identical to that obtained for the throughput of the ideal selective repeat protocol in [21].

4. Numerical results

We present in this section numerical result for the throughput efficiencies of FRSR scheme for various protocol parameters. Moreover, the superior performance of FRSR protocol can be indicated through comparative performance measures. So we will compare the through- put performance of FRSR scheme with that of both the ideal selective repeat "ISR" protocol and the ideal go- back N "IGB" protocol. The expression of the throughput of the IGB protocol is given, in [21], by:

$$\gamma=\left(\frac{d}{d+h}\right)\left(\frac{1-p_f}{1+2p_f\left(\frac{t_p+t_{IX}}{t_{IX}}\right)}\right). \tag{24}$$

Figure 6 depicts the throughput performance as a function of bit error rate, for a terrestrial T1 link with the propagation delay of 10 ms is used. We note that, as the period of the ER- timer decreases, the performance of FRSR approaches that of ISR. The effect of the ER- timer's period on the throughput becomes less significant as the bit error rate decreases.

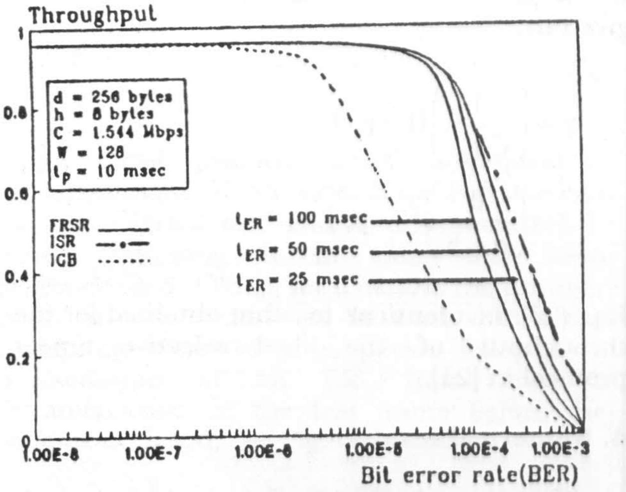


Fig. 6. Throughput versus data length.

Also, it is noted that for a long range of bit error rate FRSR performs as good as ISR.

In Fig. 7, we compare the throughput efficiencies, as a function of the data length, for the three protocols, FRSR, ISR and IGB, when a terrestrial T1 link with t_p of 20 ms is used. The curves indicate that as the bit error rate decreases, the optimum data length becomes larger. It is obvious that FRSR performs like ISR when the optimum data length is used. Even when the data length is not optimum, for a large range of data length, FRSR performs as well as ISR. This figure also shows that the optimum data lengths for FRSR and ISR are the same.

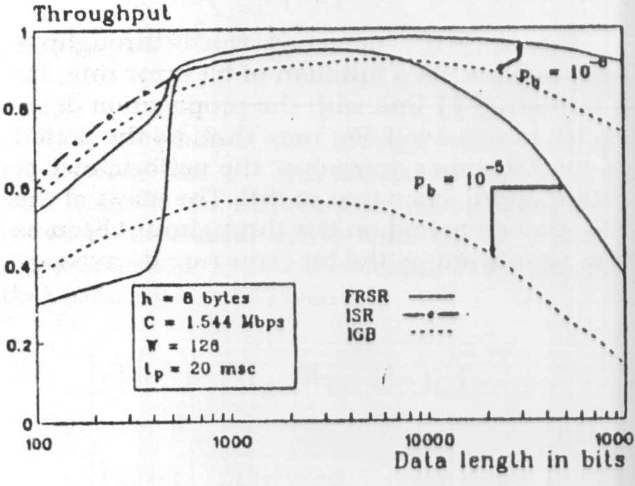


Fig. 7. Throughput versus data length.

The effect of the channel capacity on the performance of FRSR is considered in Fig. 8. The propagation delay for these curves is considered to be 10 ms. From these curves we note that, as the channel capacity increases the throughput is slightly affected when the bit error rate is not high. When the error rate is high the performance can be significantly improved by adjusting the data length to the optimum size for a particular capacity.

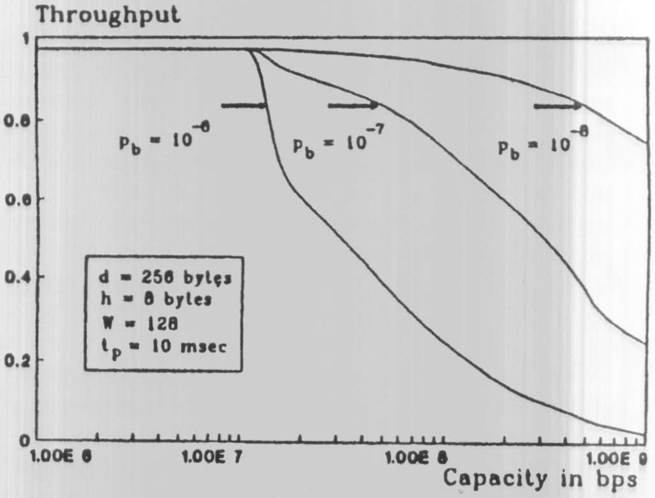


Fig. 8. Throughput versus channel capacity.

Figure 9 shows the impact of the window size on the throughput performance of FRSR for the case of T3 link with 1 ms propagation delay. It is clear that, at high bit error rate, the throughput performance is more sensitive to the window size. We also note that large window sizes have very little effect on the performance of FRSR. This is even more pronounced when the error rate is not very high. Large window size only results in a significant increase in the buffer size (i.e. the retransmission list size) with no noticeable improvement in the performance.

The effect of the propagation delay on the throughput performance of FRSR, when a terrestrial T1link is used, is considered in Fig. 10. From the curves in this figure, we note that the throughput is quite insensitive to the variation in the propagation delay when the bit error rate is not high. When the bit error rate is high and the propagation delay is large, a larger window size would improve the performance.

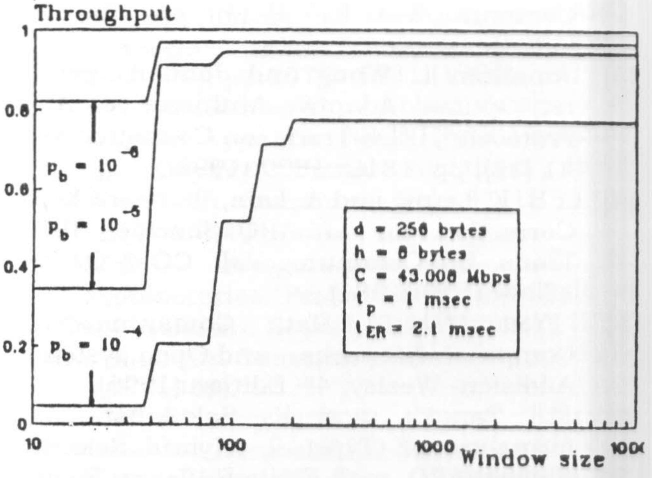


Fig. 9. Throughput versus window size.

5. FRSR delay performance

The concept of delay is defined as the time interval elapsed from the instant of the transmission of the I- frame, at the transmitter, until it is delivered to the end user, at the receiver. In fact, this delay consists of two parts: (1) the transmission and propagation time of the I- frame; and (2) the possible delay resulting from retransmission of that I- frame. This delay is referred to as ARQ delay.

In FRSR scheme, to find the delay performance, we need to find the average time that it takes for a typical I- frame from the instant of its transmission, at the transmitter, until it is correctly received at the receiver. Fig. 4 shows the delay time (t_D) experienced by the test I- frame. As it is noticeable from this figure, t_D can be found using the expressions that we have already obtained for the throughput performance of FRSR scheme.

Let N denotes the average number of re-transmissions of the test I- frame. Since the average number of transmissions of the test I-frame can be found using Eq. (17), then N can be found by subtracting 1 from Eq. (17). Thus,

$$N = \frac{p_r}{1-p_r}. \tag{25}$$

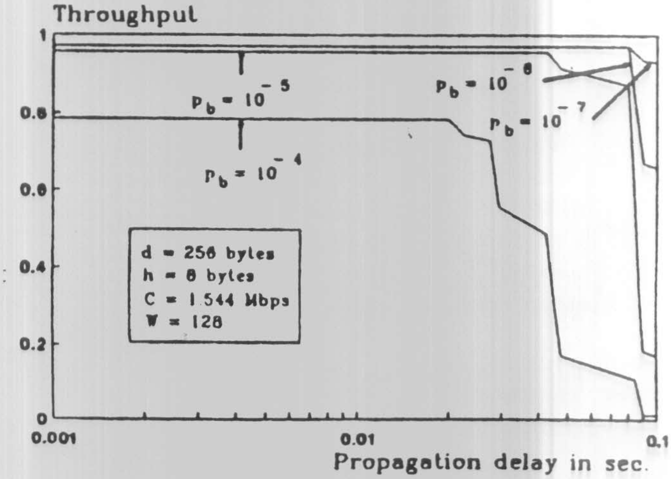


Fig. 10. Throughput versus propagation delay.

Then, t_D, the FRSR delay performance can easily be found using Fig. 4:

$$t_D = \begin{cases} t_p + t_{ix} N + \left(\sum_{n=1}^{\lfloor N \rfloor} T_n \right) + (N - \lfloor N \rfloor) T_{\lceil N \rceil}, \\ \qquad \text{if } N > 1, \\ t_p + t_{ix} + N T_1, \qquad \text{if } N \le 1. \end{cases} \tag{26}$$

Where, $\lfloor x \rfloor$ is the largest integer $\le$ x and $\lceil x \rceil$ is the smallest integer $\ge$ x.

The effect of the bit error rate on the delay performance of FRSR scheme, for T1 and T3 links with the propagation delay 10 ms, is shown in Fig. 11. The curves shown are obtained by numerical evaluation for the above expressions of the delay performance. From these curves we note that, as the channel capacity increases the delay performance becomes less sensitive to the change in the bit error rate.

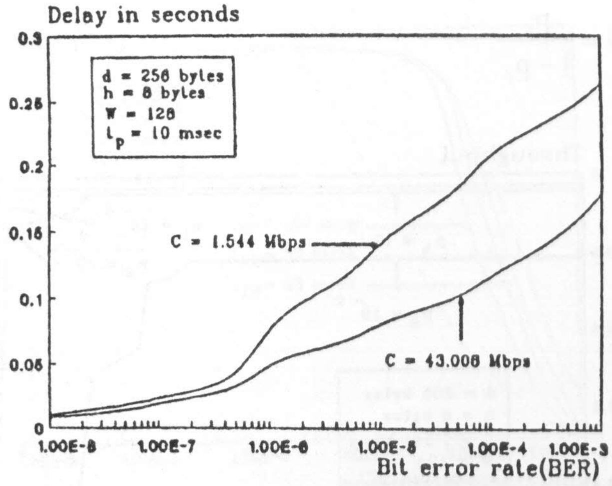


Fig. 11. Delay versus BER.

6. Conclusions

The paper has presented a new scheme, for handling retransmission in a selective-repeat ARQ protocol, named Fast Recovery Selective Repeat (FRSR) which provides a very fast error recovery mechanism. Unlike other selective repeat strategies, the proposed procedure ensures that a correctly received packet will not be retransmitted. The expressions for the throughput of FRSR are developed and the performance measures in high- speed environments, (T1and T3 links), are studied by means of numerical examples. We also obtained an expression for the delay performance of FRSR. Comparative studies between the throughput of FRSR and those of ISR and IGB are made. The comparison indicated that FRSR performs close to ISR provided that the protocol parameters are adjusted properly. FRSR is simple enough that it places little computational load on the transmit and receive processes. This is an essential requirement for a protocol that is supposed to operate in a high - speed environment.

References

[1] S. Kallel, R. Link and S. Bakhtiyari, "Throughput Performance of Memory ARQ Scheme", IEEE Trans. Vech. Tech., Vol. 48 (3), pp. 891- 899 (1999).

[2] C.X. Chen, M. Komatsu and K. Kinoshita,"Throughput Analysis of ARQ Schemes in Dialogue Communication over Half- Duplex Line", IEICE Trans. Commun., Vol. E77-B (4), pp. 485- 493 (1994).

[3] Jonathan L. Wang and John A. Silvester,"Optimal Adaptive Multireceiver ARQ Protocols", IEEE Trans. on Commun., Vol. 41 (12), pp. 1816- 1829 (1993).

[4] C. S. K. Leung and A. Lam, "Forward Error Correction for an ARQ Scheme", IEEE Trans. on Commun., Vol. COM- 29, pp. 1514- 1519 (1981).

[5] Fred Halsall, Data Communication, Computer Networks, and Open systems, Addision- Wesley, 4th Edition (1996).

[6] H. Tanaka, and K. Sakakibara,"Performance of Type -I Hybrid Selective Repeat ARQ with Finite Buffer on Fading Channel", IEICE Trans. Commun., Vol. E 80- B(1), pp.59- 66 (1997).

[7] R. H. Deng,"Hybrid ARQ Schemes Employing Coded Modulation and Sequence Combining", IEEE Trans. on Commun., Vol. 42(6), pp. 2239- 2245 (1994).

[8] R. H. Deng and M. L. Lin,"A Type- I Hybrid ARQ System with Adaptive Code Rates", IEEE Trans. on Commun., Vol. 43 (4), pp. 733- 737 (1995).

[9] S. Hara, A. Ogino, M Araki M. Okadaand N. Morinaga, "Throughput Performanc of SAW- ARQ Protocol with Adaptive Packet Length in Mobile Packet Data Transmission", IEEE Trans. Vech. Tech., Vol. 45 (3), pp. 561- 569 (1996).

[10] N. Sugimachi, Y. Hayashida and Y. Yoshida,"Delay Performance of Tandem Stop and Wait ARQ scheme", Elec. and Commun. in Japan, Vol. 73(11), pp. 65-72 (1990).

[11] M. Komatsu, Y. Hayashida and K. Kinoshita,"Throughput Performance of ARQ Protocols Operating over Generalized Two- State Markov Error Channel", IEICE Trans. Commun., Vol. E 77-B(1), pp. 35- 42 (1994).

[12] A. Annamalai, V. K. Bhargava and W. S. Lu,"On Adaptive Go Back N ARQ Protocol for Variable Error Rate

Channel", IEEE Trans. on Commun., Vol.46 (11), pp. 1405- 1408 (1998).

[13] M. Zorzi and R.R. Rao,"On the Use of Renewal Theory in the Analysis of ARQ Protocols", IEEE Trans. on Commun., Vol.44 (9), pp.1077- 1081 (1996).

[14] Y. Hayashida,"Throughput Analysis of Tandem Type Go Back N ARQ Scheme for Satellite Commu- nications", IEEE Trans. on Commun., Vol. 41 (10), pp. 1517- 1524 (1993).

[15] M.E. Anagonostou and E. N. Protonotarios,"Performance Analysis of the Selective Repeat ARQ Protocol", IEEE Trans. on Commun., COM- 34, pp. 127- 135 (1986).

[16] J.J. Metzner and D. Chang, "Efficient Selective Repeat ARQ Strategies for Very Noisy Channel", Proc. Globecom'83, pp. 35.2.1- 35.2.8 (1983).

[17] R. Fantacci,"Performance Evaluation of A Finite Buffer Generalized Selective Repeat ARQ Scheme for Satellite Communications", IEEE Trans. on Commun., Vol. 45 (2), pp. 140- 143 (1997).

[18] R. Fantacci,"Queueing Analysis of the Selective Repeat Automatic Repeat Request Protocol Wireless Packet Networks", IEEE Trans. Veh. Tech., Vol.45 (2), pp. 258- 264 (1996).

[19] J.M. Spinelli,"Self- Stabilizing Sliding Window ARQ Protocols", IEEE/ACM Trans. Network, Vol.5 (2), pp.245- 254 (1997).

[20] R.Cam and C. Leung, "Throughput Analysis of Some ARQ Protocols in the Presence of Feedback Error", IEEE Trans. on Commun., Vol.45 (1), pp. 35- 44 (1997).

[21] M. Schwartz, Telecommunication Networks: Protocols, Modeling and Analysis, Addison-Wesley, Reading, Massachusetts (1987).

Received May 7, 2000
Accepted September 25, 2000