Evaluation of Retransmission Strategies in a Local Area Network Environment

Amarnath Mukherjee
Lawrence H. Landweber
John C. Strikwerda

Computer Sciences Technical Report #789

September 1988

Evaluation of Retransmission Strategies in a Local Area Network Environment.

Amarnath Mukherjee
Lawrence H. Landweber
John C. Strikwerda

Department of Computer Sciences
University of Wisconsin
Madison, WI-53706.

Abstract

We present an evaluation of retransmission strategies over local area networks. Expressions are derived for the expectation and the variance of the transmission time of the go-back-n and the selective repeat protocols in the presence of errors. These are compared to the expressions for blast with full retransmission on error (BFRE) derived by Zwaenepoel [Zwa 85]. We conclude that go-back-n performs almost as well as selective repeat and is very much simpler to implement while BFRE is stable only for a limited range of messages sizes and error rates. We also present a variant of BFRE which optimally checkpoints the transmission of a large message. This is shown to overcome the instability of ordinary BFRE. It has a simple state machine and seems to take full advantage of the low error rates of local area networks. We further investigate go-back-n by generalizing the analysis to an upper layer transport protocol, which is likely to encounter among other things, *variable* delays due to protocol overhead, multiple connections, process switches and operating system scheduling priorities.

1. Introduction

With the advent of diskless workstations, network file systems and distributed virtual memory, there is an ever growing set of applications requiring quick response times for *large* data transfers over a local area network. In this paper we present some analytical results of the performance of different protocols over local area networks, characterized by low error rates and high bandwidth.

Degradation of performance could result from a number of factors. It could be caused by flow control (for example, the outstanding window size could be very small), or by the host to network interface, or it could be caused by the choice of retransmission strategy in case of errors. Our focus in this paper is on this last issue. The principle retransmission strategies considered are the *stop-and-wait* protocol, the *blast protocol with full retransmission on error* (henceforth referred to as BFRE), the *go-back-n* protocol and the *selective-repeat* protocol. Zwaenepoel, [Zwa 85], presents an analysis of the stop-and-wait and BFRE. He also presents simulation results for the go-back-n and selective-repeat protocols, which suggest go-back-n as the strategy of choice for local area network environments. Our main contribution is an analytical evaluation of the go-back-n and selective-repeat retransmission strategies in the local area network environment. Our results corroborate those of Zwaenepoel: BFRE becomes unstable much faster with respect to message size than go-back-n or selective-repeat. However, BFRE has a very simple state machine and makes other design issues much simpler and efficient. See for example the network interface design of Kanakia and Cheriton [Kan 88]. It also seems ideally suited for an environment where host processing time is a significant amount of the total time, precisely because the amount of "work" to be done by the host is reduced. This is the motivation for our *optimal blast protocol* which performs well for both large and small message sizes.

Previous analyses of go-back-n and selective-repeat assume low nodal processing times, high error rates and high link delays, see for example [Ana 86], [Bru 86], [Moe 86] and [Moh 87]. The principal focus of those studies is on maximization of channel throughput, given assumptions of packet arrival rates and distributions. While that clearly is a viable goal for some environments, it is not the main focus for file accesses over networks, where response times determine workstation performance. Towsley [Tow 79] provides an interesting analysis of the go-back-n and stop-and-wait retransmission strategies, deriving formulas for *individual* packet delays under general assumptions of the distribution of packet arrivals at the sending site. This analysis is more suitable for the nodes in store and forward networks.

Our study focuses on the statistics of the time to complete a multi-packet message transfer. We address both processing and transmission times. Most related work in this area, with the exception of [Zwa 85], ignore processing time as a negligible component of the delay. Measurements on local networks have shown that this delay is in fact significant.

The rest of the paper is organized as follows. Section 2 presents the model and its assumptions and the protocol definitions. Sections 3 and 4 present the analyses of go-back-n and selective-repeat respectively. Numerical results comparing these protocols are presented in Section 5. We shall see that the performance of BFRE is very sensitive to message size. In Section 6, we propose and evaluate the Optimal Blast Protocol which increases the range of operation of BFRE. Until Section 6, we assume that all processing times are deterministic, similar to [Zwa 85]. This assumption is relaxed in Section 7, where we consider the second order effects of the variation of the processing times on the go-back-n strategy. Section 8 presents our conclusions and Appendices A and B fill in some of details omitted in Sections 3 and 4 respectively.

2. Preliminaries

2.1. The Model

Figure 2.1 represents a typical network interface architecture. To transmit a packet, a station copies the data from host memory to interface memory and then transmits it onto the network. When a packet arrives at a station, it is first put in interface memory from where it is copied to the host's memory. Messages are assumed to be comprised of fixed size data packets. The time to copy a data packet between host memory and interface memory is assumed to be a constant C. The time to transmit a data packet is assumed to be a constant T. The corresponding times for acknowledgement (ACK) packets are Ca and Ta respectively. Propagation delays are assumed to be negligible. C and Ca are limited by the DMA rate of the host bus. T and Ta are limited by the network's speed. In the analyses of Sections 3 and 4, we assume that there is just one send buffer. In case of multiple send buffers, the timing diagrams used in these analyses will change, but the method of analysis and the *relative* performance of the different protocols will not. In fact, we do generalize the analysis of go-back-n in Section 7 to handle arbitrary timing sequences. The focus here is on the relative performance of different retransmission schemes. We feel our analysis should be straightforward to extend to newer, faster interfaces of the future.

Figure 2.2 shows the timing diagram of a simple sliding window protocol. We have assumed that the window size is large enough so that it does not close. The horizontal axis represents time. The upper, middle and lower lines correspond to sending station, network and receiving station activity respectively. In this diagram, we show each packet being separately acknowledged. The sender first copies a packet from its memory to its interface. This takes C time units. The network transmission of this packet takes T time units. The data is then copied at the receiving end taking another C time units. Simultaneously, the sender transmits the next packet. Every packet is separately acknowledged. Copying of the ACK packet to the interface takes Ca time units and its network transmission takes Ta time units. Figure 2.3 shows the corresponding timing diagram of the Blast protocol. Here, the receiver transmits an ACK only at the end. In both these timing diagrams, it is assumed that there is one interface buffer for sending and one for receiving, and that the interface processes one packet at a time. This makes it possible, for example in Figure 2.2, for the the sender's data transmission to overlap with its processing of an acknowledgement, i.e.,

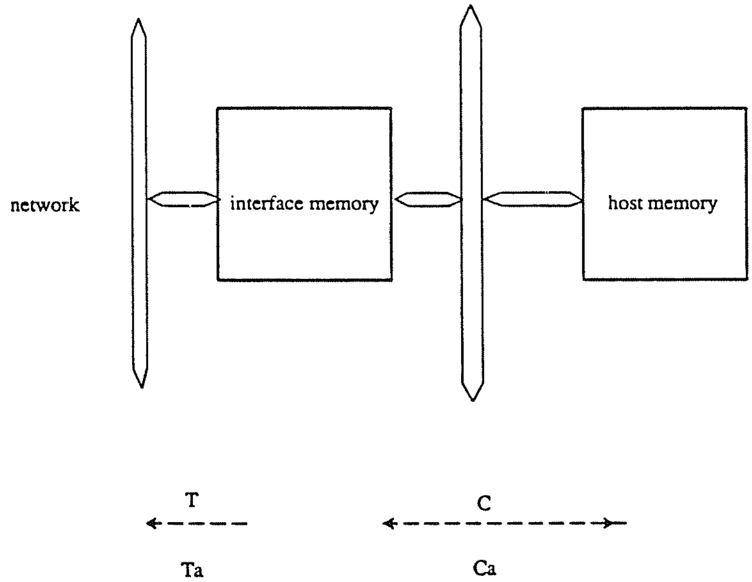

Fig. 2.1 : Network Interface Architecture

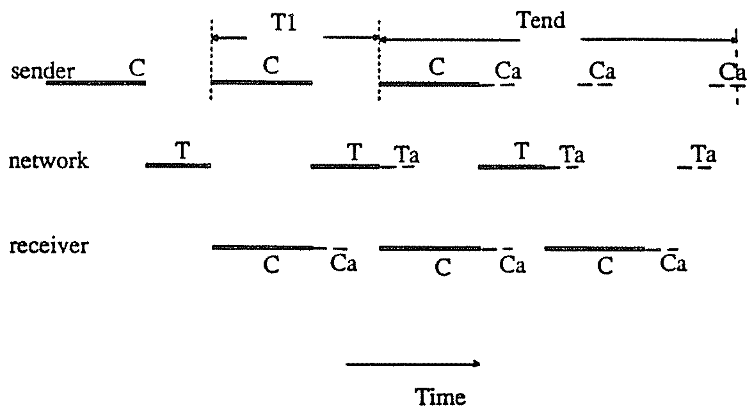

Fig. 2.2 : Timing Diagram of the Sliding Window Protocol - No Errors

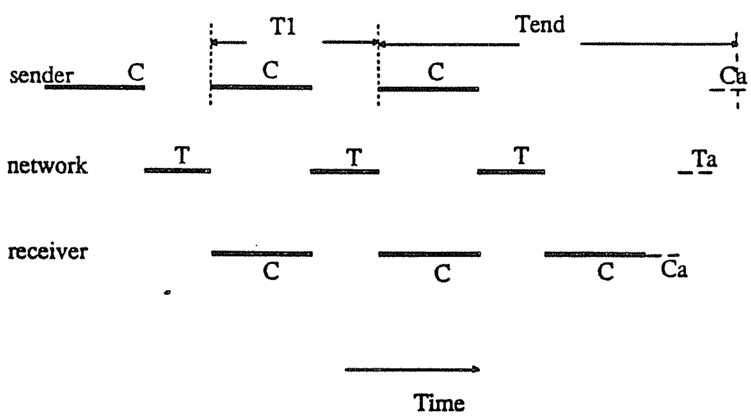

Fig. 2.3 : Timing Diagram of the Blast Protocol - No Errors

data can be transmitted onto the network while an ACK packet is being copied into *host* memory. However, copying of data to the interface from the host cannot be overlapped with transmission of the data onto the network. The actual timing diagram will depend on the implementor's choice of signals and when they are masked off or turned on. It would also depend on the number of send buffers provided. However, the analysis we present in the next section would still remain valid if the time parameters chosen were suitably modified. In fact our analysis can be extended in a straightforward manner to the faster interfaces that are currently being designed [example Son 88, Kan 88].

The next important parameter of the model relates to packet error rates. Error rates in local networks are extremely low. If one out of every n bits are in error due to electrical noise, the probability of a packet of size b bits failing is $1-(1-1/n)^b \; = \; b/n + o(b/n)$. If data is transmitted as packets of 1K bytes each then the probability of a data packet failing is 8K/n. The corresponding packet failure rate for an ACK packet of say 64 bytes, is 512/n. For a bit error rate of one in 10^8 to one in 10^{10} or less, these values are extremely low. We are not aware of any authoritative report on the actual bit error rates on local networks. However, they seem to be sufficiently low, not to warrant any concern for performance degradation just by themselves (as we shall see in Section 5). The advent of optical fibers reduces errors to even lower rates. However, although collisions (in case of random access protocols) are rare, the increased use of remote file servers and other distributed applications are likely to increase their frequency. In addition, various studies, [Son 88, Zwa 85], have reported significant error rates at network interfaces generally resulting from unavailability of buffers. Indeed Zwaenepoel suggests that packet error rates caused by interface errors are in fact somewhere in the range of one in 10^4 to one in 10^5 [Zwa 85]. Since this dominates network errors caused by random noise, we assume in our analysis that all packets have the same probability of failing, irrespective of packet size. This probability, which we denote by p_n, is an important parameter in our model. We further assume that packet errors are statistically independent as in [Zwa 85].

2.2. The Protocols

The protocols we are interested in are essentially retransmission strategies. We distinguish here between *transmission* and *retransmission* strategies. Briefly, the time when the receiver sends an ACK determines the transmission strategy (for example Blast and Sliding-Window are two different transmission strategies). A retransmission strategy, on the other hand, determines which packets are retransmitted in case of errors.

In BFRE, *all* the packets are retransmitted, irrespective of which packets were in error. We have chosen to associate Blast as the transmission strategy along with it. A Sliding-Window version with full retransmission seems to make less sense, because packets which have already been ACKed may then be (unnecessarily) retransmitted.

If the transmission strategy is Sliding-Window, the go-back-n and selective-repeat retransmission strategies work as follows: when a packet successfully reaches the receiver, it is always ACKed if it is "in-sequence". In case of selective-repeat, the receiver may also ACK out of sequence data. In both cases an error is detected at the sender by either a timer interrupt or by a NACK from the receiver. At this point, if the sender backs up to the first packet in error and restarts the transmission, the strategy is referred to as go-back-n [Tan 81]. If, on the other hand, the sender retransmits only those packets which are in error, the strategy is called selective-repeat. In go-back-n, buffering and reassembling of the message at the receiver is much simpler than in selective-repeat, but at the potential cost of retransmitting many more packets.

The mechanisms for go-back-n and selective-repeat are similar if the transmission strategy is Blast. For a N-packet transfer, the first N-1 packets are transmitted unreliably (i.e. with no corresponding ACKs). The last packet is transmitted reliably, i.e. it is retransmitted periodically until an ACK is received. This ACK indicates the first packet in error in case of go-back-n, and all the packets in error in case of selective-repeat. The receiver can also be armed with the NACK capability to flag an error immediately when it detects it (especially in the case of go-back-n retransmission strategy).

3. Go-Back-N Retransmission Strategy

In the go-back-n retransmission strategy, the sender retransmits all packets from the first packet in error. The receiver does not buffer out of sequence data. This simplifies the state machine, but at the potential cost of multiple retransmissions of successful packets. However, as we shall see, more sophisticated protocols cannot really improve on the performance of this protocol for realistic error rates.

3.1. Notation

C : time to copy a data packet between host memory and interface memory

T : time to transmit a data packet onto the network

Ca : time to copy an acknowledgement (ACK) packet between host memory and interface memory

Ta : time to transmit an ACK packet onto the network

T_1 : C + T, time between the initiation of two successive data transmissions

T_{end} : 2C + T + 2Ca + Ta, time taken (as seen by the sender) to transmit the last packet and receive its acknowledgement.

T_{detect} : The time to detect an error at the sender *given that an error has occurred.* In this section, we assume T_{detect} is a constant. We examine it in more detail in Appendix A.

3.2. Analysis

This subsection presents the analysis of the expected time and the variance of the time to transmit N packets in the presence of errors. We assume deterministic processing times (C, Ca) and transmission times (T, Ta) and ignore queueing delays. We also assume that the sender can always send (i.e., if there is a window, it never closes), an assumption justified in light of our previous assumption of deterministic delays and no queueing.

Our analysis assumes a sliding window transmission scheme. A packet transmission fails when either the data packet or its corresponding acknowledgement is lost or is corrupted. Note that the failure of an acknowledgement does not necessarily mean a failed packet transmission, if for instance the acknowledgement for the next packet arrives before the sender times out. So this assumption gives a bound on the performance of go-back-n. As stated in the previous section, we assume that packet failures are independent of their size and are statistically independent. We denote the probability of packet failure by p_n. Given these assumptions, the probability that a packet transmission fails is:

$$p = 1 - (1-p_n)^2$$

If, instead, we use the blast protocol in conjunction with go-back-n, then $p = p_n$. We have determined that this version performs similar to the sliding-window version for practical error rates, and we therefore present only the analysis of the latter.

Now, suppose that the *first* failure occurs after r packets are successfully sent. The time to send the r packets and detect the error at the sender's site is :

$$T_f(r) = rT_1 + T_{detect} \qquad 0 \le r \le N-1$$

where T_f indicates a failed transmission. For simplicity, we denote $q = 1 - p$. In go-back-n, the failure of a packet transmission marks a regeneration point of a stochastic process because all the packets starting from this point onwards have to be retransmitted. The probability of a regeneration occurring after r packets is $q^r p$.

The last packet sequence transmitted will have no errors. We denote the time for this transmission by $T_s(r)$, where r is the number of packets transmitted in this last sequence.

$$T_s(r) = (r-1)T_1 + T_{end} \qquad 1 \le r \le N$$

Its probability distribution is q^r.

Let the total time to successfully transmit N packets with the go-back-n strategy be T_N. If there are k regenerations (retransmission sequences), then the time taken (denoted by $T(N \mid k)$) is :

$$T(N \mid k) = T_f(r_1) + T_f(r_2) + \cdots + T_f(r_k) + T_s(N - \sum_{i=1}^{k} r_i)$$

where r_i is the number of packets transmitted during the i^{th} retransmission. The above equation simplifies to

$$T(N \mid k) = (N-1)T_1 + T_{end} + k\, T_{detect}$$

Let p_k be the probability that there are k regenerations given N packets. Since the last transmission always carries at least one packet successfully, the number of ways in which k regenerations can occur given N packets is $\begin{pmatrix} N+k-1 \\ k \end{pmatrix}$. To see this, note that this problem can be mapped to the problem of finding all possible integer solutions to the equation

$$X_1 + X_2 + \cdots + X_k + X_{k+1} = N$$

where $X_i \ge 0$ for $i = 1, 2, \cdots k$ and $X_{k+1} \ge 1$. Now, let $X_{k+1}' = X_{k+1} - 1$, so that $X_{k+1}' \ge 0$. Then the previous problem is analogous to finding all possible integer solutions to

$$X_1 + X_2 + \cdots + X_k + X_{k+1}' = N - 1$$

which is $\begin{pmatrix} N+k-1 \\ k \end{pmatrix}$. Then p_k is given by

$$p_k = \begin{pmatrix} N+k-1 \\ k \end{pmatrix} p^k q^N \qquad [3.1]$$

The expected time to transmit N packets successfully is now easily obtained:

$$E[T_N] = \sum_{k=0}^{\infty} T(N \mid k)\, p_k$$

$$= \left[(N-1)T_1 + T_{end} \right] \sum_{k=0}^{\infty} \begin{pmatrix} N+k-1 \\ k \end{pmatrix} p^k q^N + T_{detect} \sum_{k=0}^{\infty} k \begin{pmatrix} N+k-1 \\ k \end{pmatrix} p^k q^N$$

Now,

$$\sum_{k=0}^{\infty} \begin{pmatrix} N+k-1 \\ k \end{pmatrix} p^k q^N = (1-p)^{-N}\, q^N = 1$$

and

$$\sum_{k=0}^{\infty} k \begin{pmatrix} N+k-1 \\ k \end{pmatrix} p^k q^N = q^N\, p \sum_{k=0}^{\infty} \frac{\delta}{\delta p} \begin{pmatrix} N+k-1 \\ k \end{pmatrix} p^k$$

$$= q^N\, p\, \frac{\delta}{\delta p} \sum_{k=0}^{\infty} \begin{pmatrix} N+k-1 \\ k \end{pmatrix} p^k$$

$$= q^N\, p\, \frac{\delta}{\delta p} (1-p)^{-N}$$

and noting that $q = 1 - p$, this becomes $N \frac{p}{q}$ Thus $E[T_N]$ is given by

$$E[T_N] = \left[(N-1)T_1 + T_{end} \right] + \left[T_{detect} N \frac{p}{q} \right] \qquad [3.2]$$

Equation 3.2 has an obvious intuitive appeal. If $p = 0$, $E[T_N] = (N-1)T_1 + T_{end}$ is the time for an error free transmission (see Figure 2.2). For every failure, there is a cost of T_{detect} to detect the error. The average number of errors is the expectation of the distribution given by equation 3.1 and is equal to $N\frac{p}{q}$.

We next compute the variance of the transmission time with the go-back-n strategy.

$$var(T_N) = \sum_{k=0}^{\infty} \left(T(N \mid k) - E[T_N] \right)^2 \begin{pmatrix} N+k-1 \\ k \end{pmatrix} p^k q^N$$

$$= T_{detect}{}^2 \sum_{k=0}^{\infty} \left[k - N\frac{p}{q} \right]^2 \begin{pmatrix} N+k-1 \\ k \end{pmatrix} p^k q^N$$

$$= T_{detect}{}^2 \left\{ \sum_{k=0}^{\infty} k^2 \binom{N+k-1}{k} p^k q^N - \left[N \frac{p}{q} \right]^2 \right\}$$

Now, noting that $k^2 = k(k-1) + k$

$$\sum_{k=0}^{\infty} k^2 \binom{N+k-1}{k} p^k q^N = \sum_{k=0}^{\infty} k(k-1) \binom{N+k-1}{k} p^k q^N + \sum_{k=0}^{\infty} k \binom{N+k-1}{k} p^k q^N$$

The first term on the right hand side can be derived in a manner similar to the derivation of equation 3.2, except that we need to work with the second derivative now:

$$\sum_{k=0}^{\infty} k(k-1) \binom{N+k-1}{k} p^k q^N = p^2 q^N \sum_{k=0}^{\infty} \frac{\delta^2}{\delta p^2} \binom{N+k-1}{k} p^k$$

$$= p^2 q^N \frac{\delta^2}{\delta p^2} \left\{ \sum_{k=0}^{\infty} \binom{N+k-1}{k} p^k \right\}$$

$$= \frac{N(N+1)p^2}{q^2}$$

The second term is equal to $N\frac{p}{q}$, as derived before. These finally give

$$var(T_N) = T_{detect}{}^2 \left[\frac{N(N+1)p^2}{q^2} + \frac{Np}{q} - \left[\frac{Np}{q} \right]^2 \right]$$

$$= T_{detect}{}^2 N \frac{p}{q^2} \qquad [3.3]$$

Equation 3.3 shows that the variance of the transmission time is proportional to the variance of the number of regenerations. The proportionality constant, $T_{detect}{}^2$, is (as we shall see from Appendix A) small compared to the entire transmission time. Acknowledging every packet (or at least NACKing packets in error), reduces the time to detect an error. This is the only overhead loss of go-back-n per error. In case of complete retransmission on error, all the packets transmitted so far (possibly successfully) add to the overhead. We postpone further discussion on this until Section 5.

4. Selective Repeat.

In this section we present the analysis of the selective repeat protocol. Several variations of this protocol have been proposed. Most assume that packet error rates are very high. Since this is not true in the LAN environment, we choose the following simple version. The sender transmits all N packets in the first *round.* The receiver sends an acknowledgement at the end of the round with a bit vector indicating the packets in error, which are retransmitted in the next round. This procedure continues until all packets have been successfully transmitted and received.

If there are k packets transmitted in a round, then the time taken is $kT_1 + T_{ohd}$, where $T_1 = C + T$ as before, and T_{ohd} is the overhead per round. We assume in the following analysis that the ACK sent by the receiver is never in error. This assumption is strictly not necessary, but makes the results more intuitive and understandable. The analysis resulting from this simplification should favor selective repeat. Our main motivation in this section is to show that selective repeat cannot do very much better than go-back-n for practical error rates, so we choose to favor intuitive understanding over rigor.

To motivate the analysis, the reader is referred to Figure 4.1. We have broken the time line, as viewed by the sender, into *rounds*. In each round, all outstanding packets are transmitted. Correctly received packets are indicated by a (/) while those requiring retransmission are indicated by an (x). The time line can be seen to consist of the sum of two random variables X and Y, where X is the sum of all the T_1's and Y the sum of all the T_{ohd}'s. The time to complete transmission of N packets is

$$T(N) = X + Y$$

and therefore,

$$E[T_N] = E[X] + E[Y]$$

and

$$var(T_N) = var(X) + var(Y) + 2Cov(X,Y)$$

where $Cov(X,Y)$ is the covariance of X and Y and is given by [Tri 82]

$$Cov(X,Y) = E[XY] - E[X]E[Y]$$

The covariance term is not zero because the number of packet failures and the number of rounds are related (for example, the number of errors is at least equal to one less than the number of rounds).

4.1. Distribution of X

Each packet transmission takes a slot of duration T_1. Let us now consider a possible sequence of correct and erroneous transmissions which take $N + k$ slots (of size T_1 each), $k \geq 0$. Clearly, the $(N+k)^{th}$ slot is always a correct transmission. Hence, the total number of ways of distributing the k errors is

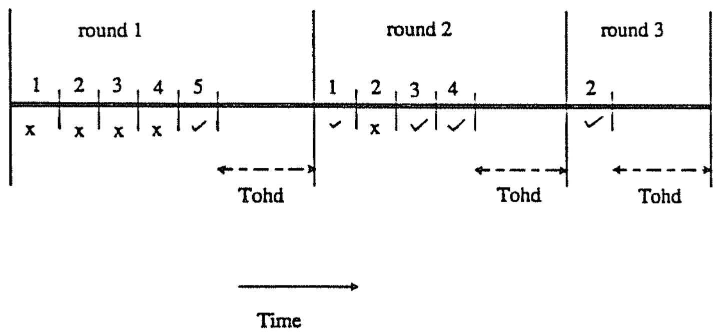

Fig. 4.1 : Selective Repeat, N = 5

$$\binom{N+k-1}{k}.$$

The probability of an error in a slot is $p = p_n$. This gives us

$$Pr[X = (N+k)T_1] = \binom{N+k-1}{k} p^k q^N$$

Therefore, .

$$E[X] = T_1 \sum_{k=0}^{\infty} (N+k) \binom{N+k-1}{k} p^k q^N$$

which simplifies to (as in equation 3.2)

$$= T_1 \left[N + N \frac{p}{q} \right]$$

and using $q = 1-p$, we finally get

$$E[X] = T_1 \frac{N}{q} \qquad [4.1]$$

The variance of X is given by:

$$var(X) = T_1^2 \sum_{k=0}^{\infty} \left[N+k-\frac{N}{q} \right]^2 \binom{N+k-1}{k} p^k q^N$$

$$= T_1^2 \sum_{k=0}^{\infty} \left[k-N\frac{p}{q} \right]^2 \binom{N+k-1}{k} p^k q^N$$

We know the result of this sum from the derivation of equation 3.3:

$$var(X) = T_1^2 \frac{N\,p}{q^2} \qquad [4.2]$$

4.2. Distribution of Y

For every round, we have a fixed overhead T_{ohd}. So $Y = T_{ohd} * \#rounds$. Now, the distribution of the number of rounds is given by

$$Pr[\,\#rounds \le k\,] = \sum_{r_1=0}^{N} \binom{N}{r_1} p^{r_1} q^{N-r_1} \sum_{r_2=0}^{r_1} \binom{r_1}{r_2} p^{r_2} q^{r_1-r_2} \ldots \sum_{r_{k-1}=0}^{r_{k-2}} \binom{r_{k-2}}{r_{k-1}} p^{r_{k-1}} q^{r_{k-2}-r_{k-1}} q^{r_{k-1}}$$

which simplifies to

$$Pr[\,\#rounds \le k\,] = (1 - p^k)^N \qquad [4.3]$$

To see this, note that the $q's$ can be grouped together from the right to the left without much ado and become q^N. Now, let us concentrate on the rightmost $\sum$. We have $\sum_{r_{k-1}=0}^{r_{k-2}} \begin{pmatrix} r_{k-2} \\ r_{k-1} \end{pmatrix} p^{r_{k-1}}$, which becomes $(1+p)^{r_{k-2}}$. This along with the next inner $\sum$ gives $\left(1+p(1+p)\right)^{r_{k-3}}$ and so on. Finally, after the left-most $\sum$ is evaluated, we have

$$q^N (1+p(1+p(\cdots)\cdots)^N$$

$$= q^N (1+p+p^2+\cdots+p^{k-1})^N$$

which simplifies to equation 4.3. Viewed another way, since the total number of rounds is $\le k$, each packet is transmitted successfully in at most k attempts. The probability of this event is $(1-p^k)$. Since there are N packets, all of them encountering errors independently of each other, we get equation 4.3.

The expected cumulative overhead $E[Y]$ is now given by

$$E[Y] = T_{ohd} * E[\#rounds]$$

$$= T_{ohd} * \left[\sum_{k=0}^{\infty} Pr[\#rounds > k] \right]$$

$$= T_{ohd} * \left[\sum_{k=0}^{\infty} \left[1 - (1-p^k)^N \right] \right]$$

For $Np \ll 1$, this last expression can be approximated by

$$T_{ohd} \left(1 + \sum_{k=1}^{\infty} N p^k \right)$$

and this yields

$$E[Y] = T_{ohd} \left(1 + N \frac{p}{q} \right) \qquad [4.4]$$

The variance of Y is given by:

$$var(Y) = T_{ohd}^2 \sum_{k=0}^{\infty} k^2 \left\{ (1-p^k)^N - (1-p^{k-1})^N \right\} - E[Y]^2$$

Alternatively, the variance of Y can be obtained as follows:

$$var(Y) = T_{ohd}^2 \sum_{k=1}^{\infty} \left[k - \left[1 + \frac{Np}{q} \right] \right]^2 \left[\left[1-(1-p^{k-1})^N \right] - \left[1-(1-p^k)^N \right] \right]$$

Now using the formula for summation by parts, and assuming $Np \ll 1$, we can approximate this as follows:

$$\frac{var(Y)}{T_{ohd}^2} \sim \left[\frac{Np}{q}\right]^2 + \sum_{k=1}^{\infty}\left[\left[k+1-\left[1+\frac{Np}{q}\right]\right]^2 - \left[k-\left[1+\frac{Np}{q}\right]\right]^2\right]\left[1-(1-p^k)^N\right]$$

$$\sim \left[\frac{Np}{q}\right]^2 + \sum_{k=1}^{\infty}\left[2k+1-2\left[1+\frac{Np}{q}\right]\right] Np^k$$

and this finally yields

$$var(Y) \sim T_{ohd}^2 \frac{Np}{q}\left[1-\frac{Np}{q}\right] \qquad [4.5]$$

Equations 4.4 and 4.5 along with the covariance term from Appendix B give the variance of the transmission time of selective repeat. The results are presented in the next section.

5. Numerical Results.

This section compares the mean and variance of the transmission times of the go-back-n, selective-repeat and the BFRE protocols. The curves for BFRE are obtained from the analysis of [Zwa 85]. The results for go-back-n and selective-repeat are obtained from the derivations in Sections 3 and 4. We use the measured values of C, Ca, T and Ta reported in [Zwa 85] (Table 5.1). These values are getting progressively smaller with faster networks and interfaces, but we expect the relative times to be the similar at least in the near future.

Parameter	Value
C	1.35 msec
Ca	0.17 msec
T	0.82 msec
Ta	0.05 msec

Table 5.1 : Parameter Values

Figure 5.1 shows the expected time to transfer N packets for the different protocols, for N = 64 and N = 512. For N=64, all three protocols have almost the same expected time for a packet error rate of 10^{-4} to 10^{-5} (the error range that we can expect in a local area network environment). As N increases, BFRE

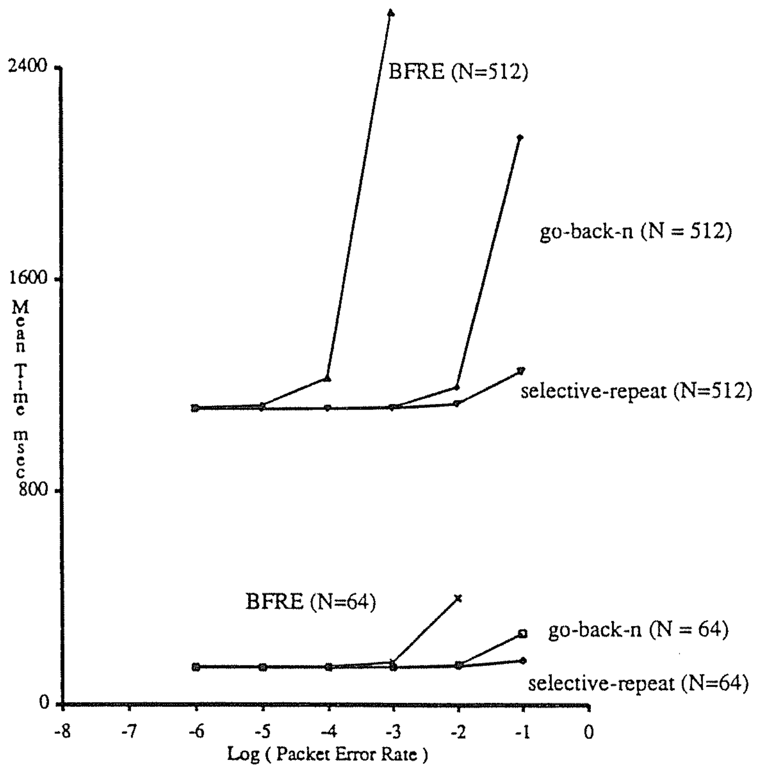

Figure 5.1 : Mean time to transfer N packets versus packet error rate

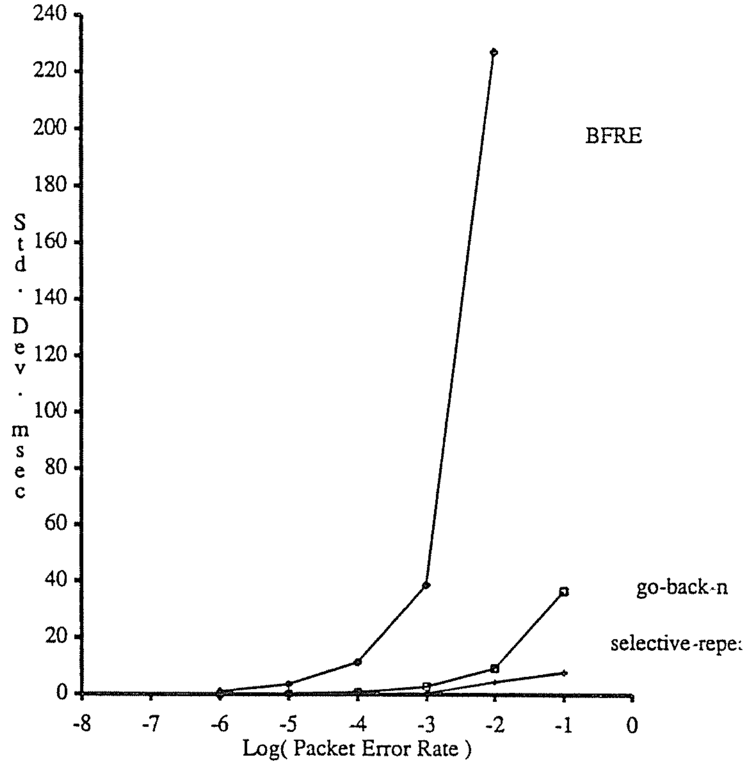

Figure 5.2 : Standard deviation of the time to transfer N = 64 packets versus packet error rate

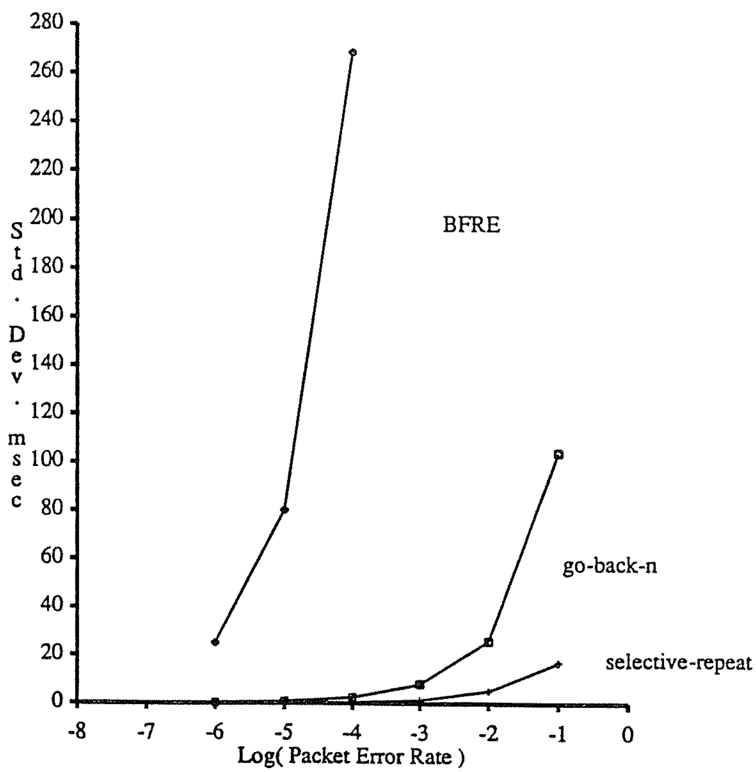

Figure 5.3 : Standard deviation of the time to transfer N = 512 packets versus packet error rate

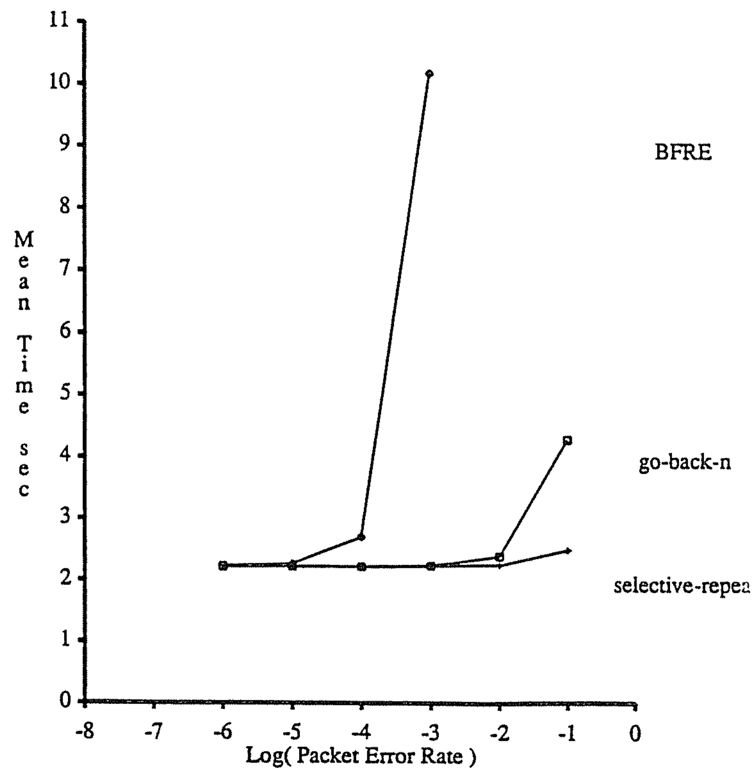

Figure 5.4 : Mean time to transfer N = 1024 packets versus packet error rate

starts performing poorly. Go-back-n however fares almost as well as selective repeat even for N = 512.

An estimate of a parameter could be misleading without an estimate of its error. We therefore plot the standard deviation of the transmission times in Figure 5.2. The curves are for N = 64. The curve for BFRE assumes that the receiver has the NACK capability so that the sender can detect a failed transmission early. Go-back-n can be seen to have almost as low a standard deviation as selective-repeat for the error range of 10^{-4} to 10^{-5}. Selective-repeat does better for error rates of 10^{-2} and higher but that portion of the curve is insignificant from a practical standpoint. The key point here is that go-back-n has a simpler state machine than selective-repeat and performs almost as well.

In Figure 5.3, we have plotted the standard deviation curves for N = 512. This shows that even for large N, go-back-n is still a viable protocol. This figure clearly demonstrates that for large messages, the BFRE protocol, if adopted, should be decomposed into multiple BFRE's. We shall address this point in the next section in more detail. Figure 5.4 gives the expected time to transmit N=1024 packets to show the effect of a 1 Mbyte transfer, assuming 1 Kbyte packet size. Transmitting a 1024x1024 image with 1 byte per pixel will result in a 1 Mbyte transfer. Go-back-n performs very well even for such a large N. These figures indicate that go-back-n is relatively stable for a wide range of message sizes and packet error rates. Since its state machine is considerably simpler than selective repeat, it seems to be a good retransmission strategy for local area network environments. In the next section we show that for large messages, adding a checkpointing mechanism to BFRE at the right places is also a good alternative.

6. Optimal Blast Protocol.

The Blast protocol with full retransmission on error (BFRE) is aesthetically simple and seems to take full advantage of the low error rates and high bandwidth of local area networks. However, its performance, especially the variance of the time to transfer large messages degrades considerably as message sizes increase. To avoid the performance penalties, without sacrificing much of the simplicity of the BFRE protocol, transmission of a large message can be decomposed into multiple BFRE's. The number of packets in each BFRE could be *fixed apriori* or could be *variable*, with the latter enjoying the obvious advantages:

(i) Dynamic adjustability to changes in observed network error rates.
(ii) Tuning according to each individual sender's performance objectives.

The first point is obvious, especially if the error rates fluctuate with time (provided, of course, they can be estimated accurately). The second point emphasizes that the optimization criteria of different communicating pairs need not be the same. In the following discussion, we choose not to minimize the expected time to transmit a message because it is almost equal to the error free transmission time for practical error rates. Instead, we propose to constrain the standard deviation of the time to transmit the packets to some constant times the expected time to transmit the packets successfully. That is, the standard deviation, which we interpret as the error in the estimate of the mean, is constrained by the following equation:

$$\sigma(T_M) \leq rE[T_M] \qquad 0 < r < \infty \tag{6.1}$$

Typically, we would like r to have a very small value. Equation 6.1 says that we are less willing to accept large deviations for smaller messages than for larger messages. Also, we want the standard deviation to be smaller than some constant times the expected time to transmit the entire message. We shall see the consequence of this criterion shortly.

To achieve this desired standard deviation, for an M-packet-transfer, we propose to "checkpoint" the (blast) transmission by requiring a mandatory ACK from the receiver after every N packets, where N is chosen such that equation 6.1 is satisfied. This means that we have approximately M/N BFRE's in series, each of N packets. We call N the *optimal blast size*.

Let each BFRE be of size at most N packets. Let $n = \frac{M}{N}$. Then, ignoring the end effects of truncation and assuming that successive BFRE's are statistically independent, we have

$$var(T_M) = n\ var(T_N) \tag{6.2}$$

and

$$E[T_M] = n\ E[T_N] \tag{6.3}$$

The constraint in equation 6.1 can then be rewritten as

$$n\ var(T_N) \leq r^2 E[T_M]^2 = r^2 n^2 E[T_N]^2 \tag{6.4}$$

Now, if the receiver NACKs on errors, [Zwa 85] shows that the variance of the time to transmit N packets using BFRE is

$$var(T_N) = t_0(N)^2 \, \frac{p\ (1+p\ )}{q^2} \tag{6.5}$$

where $t_0(N)$ represents the time to transmit N packets with no errors, p is the probability of a BFRE failing and $q = 1-p$. The expected time to transmit the N packets is

$$E\,[T_N] = \frac{t_0(N)}{q} \tag{6.6}$$

From equations 6.4, 6.5 and 6.6 we get

$$n \geq \frac{p(\,1+p\ )}{r^2}$$

and since n ≈ M/N, we have

$$N \leq \frac{Mr^2}{p\ (\ 1+p\ )} \tag{6.7}$$

The probability of a BFRE failing, p, is of course dependent upon N. It is the probability that at least one of the N packets that are transmitted fail, and is given by

$$p = 1 - (1-p_n)^{N+1} \tag{6.8}$$

Given M, r and p_n, we can obtain N by solving equations 6.7 and 6.8 iteratively to obtain the optimal blast size which satisfies equation 6.1. In Figures 6.1 through 6.3, we show the optimal blast size for error rates between 10^{-6} and 10^{-4}, for different message sizes, M. Both the axes are in units of number of packets. It is interesting to see how the optimal blast size drops *rapidly* with increasing r and p. In figures 6.4 and 6.5, we show a comparative performance of the optimal blast protocol and the normal BFRE protocol. The optimal blast protocol in these figures uses the optimal blast size for any particular M, r and p. In Fig. 6.4, we have plotted the ratio of the expected times of the optimal blast protocol and BFRE. This value is close to unity. However, in Fig. 6.5, we see the very sharp improvement in the standard deviation of the time, which essentially means that we have increased the confidence in the estimate of the mean almost for free. The reason is that the expected time is almost equal to the error free transmission time for practical error rates, but the standard deviation can still be large for large message sizes. We however see one problem with the optimal blast protocol: for small M, the ratio of the two expected times is greater than unity, especially as r gets smaller. This is because in our optimal blast, the sender waits for an ACK of the previous packet group before it starts transmitting the next packet group. This causes the pipeline to empty out and

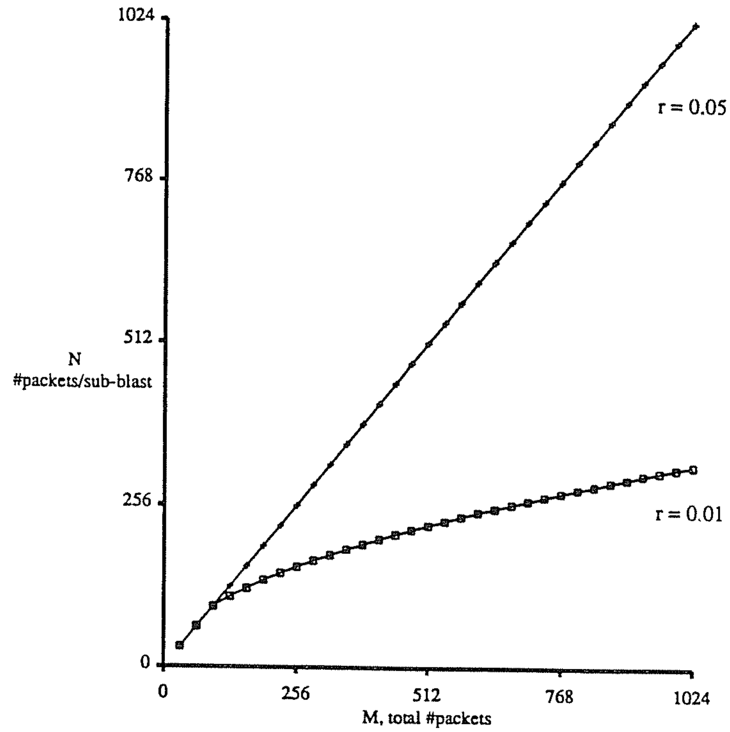

Fig 6.1 : Optimal #packets per blast for $p_n = 10^{-6}$

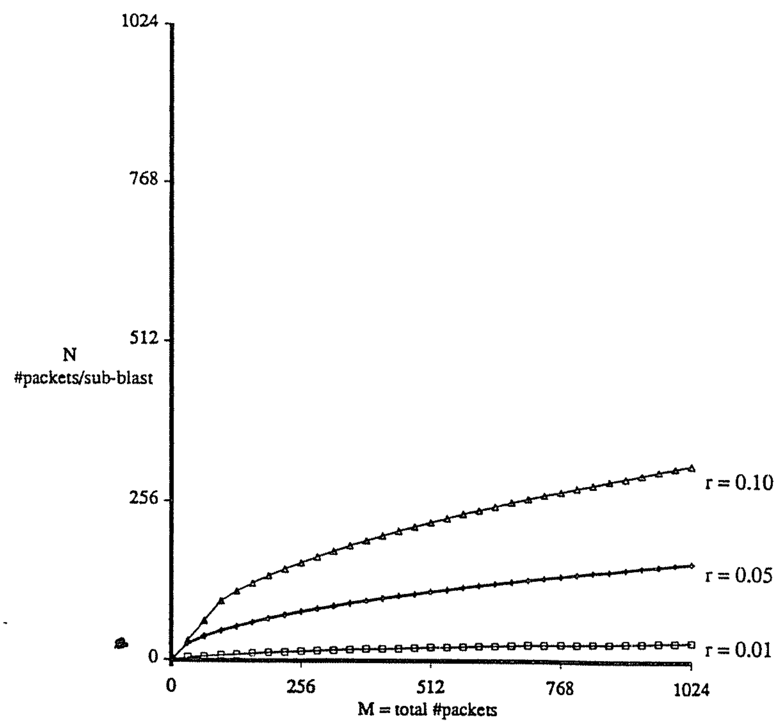

Fig 6.3 : Optimal #packets per blast for $p_n = 10^{-4}$

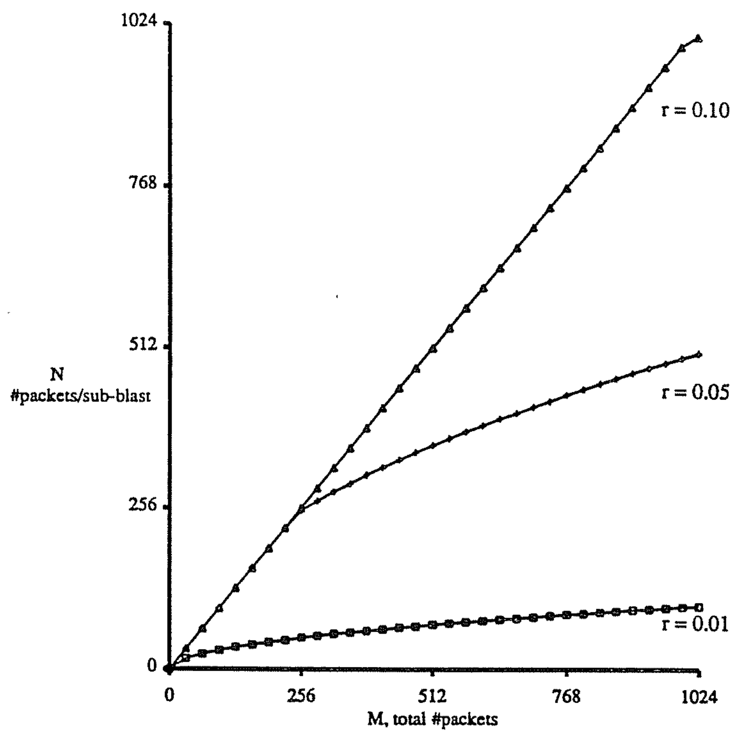

Fig 6.2 : Optimal #packets per blast for $p_n = 10^{-5}$

fill up again for each sub-blast. The delay resulting from this dominates over the expected time of a simple BFRE for smaller message sizes because the probability of a retransmission is extremely low. Smaller values of r increase the number of sub-blasts (see figure 6.3) exacerbating the problem. However, as M increases, one of the properties of constraint 6.1 is that it increases the sub-blast size even though p_n and r are the same. The pipeline does not empty out as often as before. In addition, the probability of a retransmission increases for the simple BFRE. These factors pull the ratio of the expected times below unity as the total number of packets, M, increases. The standard deviation to the transmission time improves for all M, though it is more pronounced for large M.

To prevent the degradation in the expected transmission time for small M, we propose the following modification to the protocol:

(i) The sender determines the optimal blast size, N, for the given message.

(ii) It then transmits packets 1 through N-1 in the current BFRE without requesting an ACK from the receiver.

(iii) It transmits packet N with the REQUEST_FOR_ACK bit set.

(iv) Without waiting for the ACK, it continues with the next blast using steps (ii) and (iii).

(v) The receiver ACKs the packets which have their REQUEST_FOR_ACK bit set, provided it has received all the packets with sequence numbers greater than the previously ACKed packet and less than the current one. It can also NACK packets in error. Dropped packets however will have to be detected by the sender's timeout mechanism.

(vi) In case of an error (either a NACK or a timeout), the sender retransmits the whole "window" of outstanding BFRE's not yet ACKed. This leads to a go-back-n retransmission across sub-blasts, although each smaller sub-blast is still fully retransmitted!

We note that the sender does *not* have to negotiate the sub-blast size with the receiver in advance. In a window based flow control scheme, there has to be space for the packet when it arrives at the receiver, but flow control and error control are orthogonal functions here. One bit in the packet could serve as REQUEST_FOR_ACK/ NO_ACK, and could be set whenever the sender wants an ACK. Thus the size of a sub-blast could change with time even between the same communicating pairs. This could happen, for instance, if the sender's effective window size drops because it senses congestion. [Jai 86] and [Jac 88]

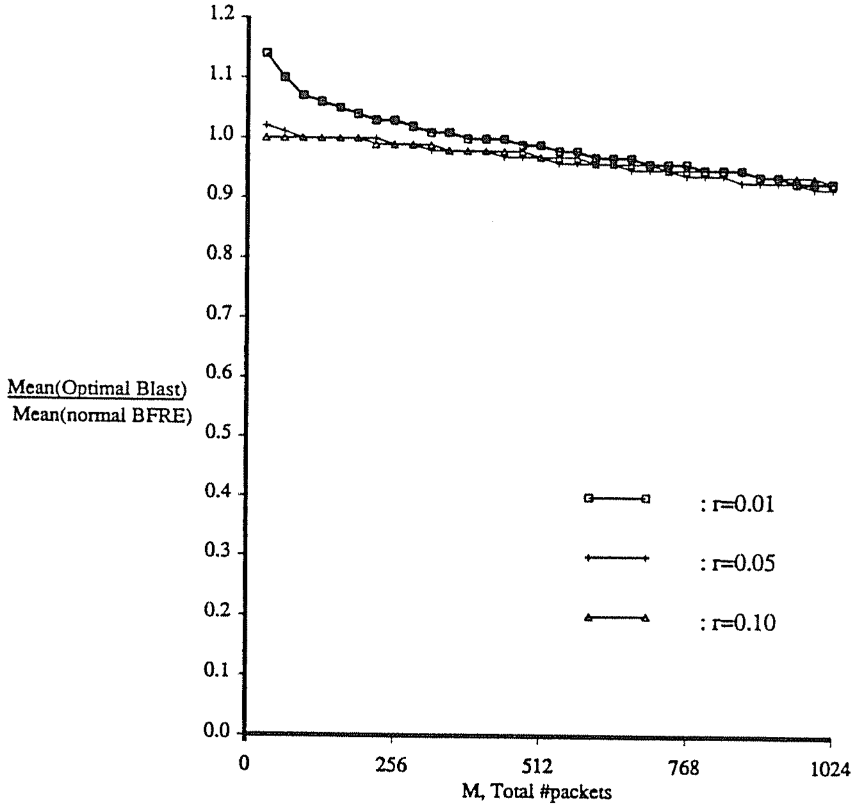

Fig 6.4 : Ratio of expected time to transmit with optimal number of packets per blast to ordinary BFRE.
$p_n=10^{-4}$

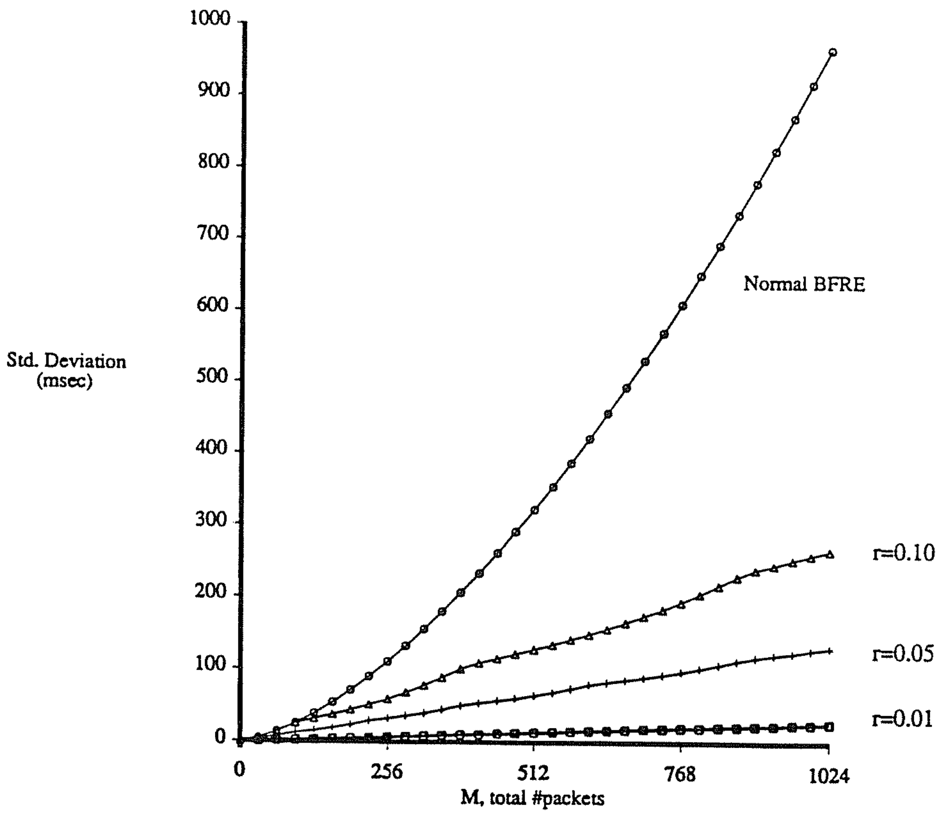

Fig 6.5 : Std. Deviations of transmission time with optimal number of packets per sub-blast.
$p_n=10^{-4}$

claim that packet errors are a good indicator of congestion, and their congestion control protocol shrinks the effective window size to deal with it. The window is slowly increased after that. Their scheme fits in harmoniously with the sender's choosing the optimal blast size independent of the receiver. All that the sender has to do is to set the sub-blast size as min{N, congestion_window, flow_window}, where N is the optimal blast size from equations 6.7 and 6.8.

7. Generalized Analysis of Go-back-n

We now generalize the go-back-n results by removing the deterministic time constraints under which the results were obtained in Section 3. We begin with some notation and definitions. Denote the time from the beginning of the transmission of packet i to the beginning of the transmission of packet $i+1$ by the random variable X_i, if the packet transmission was successful, i.e. both the data packet and its ACK were successful. This corresponds to T_1 in Section 3. The time corresponding to T_{end} is denoted as X_{end}. Thus X_{end} is a random variable denoting the time from the beginning of the transmission of the last packet until its ACK is received, given that the transmission is successful. Similarly let $T_{detect,i}$ be the time to detect the ith error if one occurs. It is easy to see that the time to transmit N packets given that k regenerations have occurred is

$$T(N \mid k) = \sum_{i=1}^{N-1} X_i + X_{end} + \sum_{i=1}^{k} T_{detect,i} \qquad k = 0, 1, 2, \cdots$$

We assume that the X_i's are independent and identically distributed random variables with mean $E[X]$ and second moment $E[X^2]$. Also let their common Laplace transform be $X(s)$. Likewise we assume that $E[X_{end}]$, $E[X_{end}^2]$ and $X_{end}(s)$ are the mean, second moment and Laplace transform of X_{end}, and $E[T_{detect}]$, $E[T_{detect}^2]$ and $T_{detect}(s)$ are the mean, second moment and Laplace transform of T_{detect} respectively (of course, we are assuming the $T_{detect,i}$ to be i.i.d. random variables too). Then the Laplace transform of T(N) which we denote by $T(s)$ is given by

$$T(s) = \sum_{k=0}^{\infty} \binom{N+k-1}{k} p^k q^N X(s)^{N-1} X_{end}(s)\, T_{detect}(s)^k$$

$$= \frac{q^N X(s)^{N-1} X_{end}(s)}{[\ 1 - p\ T_{detect}(s)\ ]^N} \qquad \textbf{[7.1]}$$

Taking the natural logarithms of both sides of equation 7.1, we get

$$\ln(\ T(s)\ ) = N\ \ln(\ q\ ) + (N-1)\ln(\ X(s)\ ) + \ln(\ X_{end}(s)\ ) - N\ \ln(\ 1-p\ T_{detect}(s)\ ) \qquad [7.2]$$

Now, we note that $E[X] = -\frac{d}{ds}X(s)\,|_{s=0}$ and $E[X^2] = \frac{d^2}{ds^2}X(s)\,|_{s=0}$, and similarly for the other random variables. Thus differentiating the left hand side of equation 7.2 once and putting s = 0 gives $E[T_N]$ and differentiating it twice and evaluating it at s = 0 yields $var(T_N)$. The resultant equations are :

$$E[T_N] = (N-1)E[X] + E[X_{end}] + E[T_{detect}]\,N\frac{p}{q} \qquad [7.3]$$

and

$$var(T_N) = (N-1)\ var(X) + var(X_{end}) + E[T_{detect}]^2\ \frac{N\,p}{q^2} + N\ \frac{p}{q}\ var(T_{detect}) \qquad [7.4]$$

For the deterministic case in Section 3, $E[X] = T_1$, $E[X_{end}] = T_{end}$, $E[T_{detect}] = T_{detect}$ and $var(X) = var(X_{end}) = var(T_{detect}) = 0$. As one would expect, the equations are then the same as equations 3.2 and 3.3. Equations 7.3 and 7.4 are independent of the actual distribution of the X_i's and $T_{detect,i}$'s, but depends only on their mean and variance. It is clear that the variance of the time to successfully transmit N packets will increase linearly with the variance of the protocol processing and transmission times and the time to detect errors. Also, equations 7.3 and 7.4 are more general in the sense that they factor in various unaccounted for "random delays."

We do not have any real-life data on the variance of packet processing times and transmission times. In real implementations, there is likely to be a variation in packet processing times by the two stations. The variance of the transmission times could also be caused by network load, which, although usually low, can occasionally be quite high[Gus 87]. It is our surmise that packet processing and transmission times will be normally distributed about their mean, but this needs empirical verification. Equation 7.4 is valid only if the random variables X_i, X_{end} and the $T_{detect,i}$'s are independent of each other. It should apply to protocols implemented at the transport level or below, where correlations among consecutive packet transmission times are likely to be weak. The results of this section provide a means of isolating the communication of a pair of nodes from all other traffic. To some extent we have an expression for the mean and the variance of the delay for a bulk data transfer under a multiple-sender/multiple-receiver assumption. The results also apply to multiple hop transmissions, provided that windows never close at intermediate stations. The main problem that remains is to determine the mean and the variance of the $X_i's$ and X_{end}. The latter is likely to be more important as the number of hops increase and/or load from the other connections increase.

8. Conclusions and Future Work.

We have presented analytical results for the expectation and the variance of transmission times for different retransmission strategies over local area networks. For small messages (i.e., small number of packets per message), BFRE, go-back-n and selective-repeat, all perform well. However, as the message size increases, BFRE shows larger mean and variance than go-back-n while the latter does almost as well as selective repeat. These conclusions are based on an estimate of the packet error rate between 10^{-4} and 10^{-5}. More reliable network interfaces will likely reduce error rates on local area networks. Under such conditions, BFRE will perform almost as well as the others, and given its simplicity, will be a more attractive protocol. For error rates which we observe today, go-back-n and the optimal blast protocol will be more viable alternatives since any protocol has to deal with a wide range of message sizes.

The analysis of the selective repeat protocol shows that it is only marginally better than go-back-n for practical error rates. The derivation of the joint distribution of the number of errors and the number of rounds in Appendix B is interesting in its own right. We see the application of a number of simple mathematical tools to capture a rather complex phenomena.

We have also extended the analysis of go-back-n to handle the second order effects of variable processing and transmission times. We assumed a general distribution of delays, instead of a deterministic one and showed how they affect the expected time and the variance of the transmission time of large messages. Possible application of this model will be datagram oriented transport protocols with associated protocol processing overhead, variable delays due to multiple connections, and variable transmission times due to network load. We showed that for go-back-n the variance of a message transmission time increases linearly with the variance of individual packet transmissions in addition to the that contributed by erroneous transmissions.

This study can be extended in many directions. We are currently investigating the incorporation of windows. Characterizing how workload affects the distribution of the X_i's in Section 8 (example: are they normally distributed? If so, what is their mean and variance ?) is also a very interesting problem.

Appendix A

This section presents the analysis of T_{detect}. Given M+1 packets, of which the *first* packet has failed, we are interested in the time the sender takes to detect the error. We are assuming that errors due to electrical noise are much lower than errors due to packet losses at the interface. In a single hop LAN, all packets are received in the order sent. Therefore the receiver can detect a dropped packet with sequence number s if it receives any packet with sequence number of $s+1$ or greater. It then NACKs sequence number s. If the NACK gets through successfully, the error is detected at the sender; otherwise, a NACK from a future packet is needed for error detection. Ultimately, if there are no packets left (i.e., all M packets or their NACKs failed), the sender times out after $T_{timeout}$ time units. So error detection at the sender is upper bounded by $T_{timeout}$.

Now let

$q_n = (1-p_n)$, the probability that a packet does not fail.

$u = 1 - q_n^2$, the probability that a packet exchange fails

Then,

$$Pr[\, i \text{ failures to detect}\,] = \begin{cases} (1-u)u^i & 0 \le i \le M-1 \\ u^M & i=M \end{cases} \qquad \text{[A1]}$$

Distribution of time to detect

In the following discussion, C and T are defined as before. T_n is the time to transmit a NACK packet, and Cn is the time to copy it from (to) the interface memory to (from) the host memory. We assume that $Cn \le T$. Let

$T_{start} = C + T$

$T_{end}^{(1)} = (C+T) + (C+Cn)$

$T_{end}^{(2)} = (C+T) + (Cn+T_n)$

$T_{end}^{(3)} = (C+Cn) + (Cn+T_n)$

Then the time to detect the error after exactly i failures, T_i is

$$T_i = \begin{cases} T_{start} + T_{end}^{(1)} + (i+1)(C+T) & 0 \le i \le M-3 \\ T_{start} + T_{end}^{(2)} + (i+1)(C+T) & i=M-2 \\ T_{start} + T_{end}^{(3)} + (i+1)(C+T) & i=M-1 \\ T_{timeout} & i=M \end{cases} \qquad \textbf{[A2]}$$

The interested reader is urged to verify these equations by drawing the appropriate timing diagrams.

The mean time to detect the error given M+1 packets is now easily obtained from equations A1 and A2:

$$T_{detect}(M+1) = T_{start} \sum_{i=0}^{M-1} (1-u)u^i + T_{end}^{(1)} \sum_{i=0}^{M-3} (1-u)u^i + T_{end}^{(2)} (1-u)u^{M-2} + T_{end}^{(3)} (1-u)u^{M-1} +$$

$$(C+T) \sum_{i=0}^{M-1} (1-u)u^i (i+1) + T_{timeout}\, u^M$$

which simplifies to

$$T_{detect}(M+1) = T_{start}(1-u^M) + T_{end}^{(1)}(1-u^{M-2}) + T_{end}^{(2)}(1-u)u^{M-2} + T_{end}^{(3)}(1-u)u^{M-1} +$$

$$(C+T)\left[\frac{1-(M+1)u^M + Mu^{M+1}}{1-u} \right] + T_{timeout}\, u^M \qquad [A3]$$

Equation A3 gives the mean time to detect an error if the receiver NACKs an erroneous packet. We have found that the time to detect an error is small and is almost a "constant". Low packet loss rates make it extremely unlikely that consecutive errors will occur. Most of the time, a NACK will arrive almost immediately. Thus *recovery* of go-back-n in case of an error is quick if M is large. It is almost independent of the timeout $T_{timeout}$, because of the *feedback control* provided by the NACK. Blast protocols with NACK and complete retransmission on error have also been shown to be independent of $T_{timeout}$ [Zwa 85]. Tuning $T_{timeout}$ to a very low value to reduce the time to recovery is another possible solution, but it is a *feedforward control* and can lead to needless retransmissions.

Appendix B

In this appendix, our goal is to compute the covaraince of the random variables X and Y, where X and Y represent the total number of errors and the total number of rounds respectively to complete the transmission of N packets using selective-repeat. Since we have to compute E[XY], we are interested in the joint distribution of the random variables X and Y. If $Y=R+1$, $R \ge 0$ and $X=k$, then the k errors are distributed as $k_1, k_2, \cdots, k_R$, such that

$$N \geq k_1 \geq k_2 \geq \cdots \geq k_R > 0 \qquad \text{(B.1)}$$

Note that the last (strict) inequality stresses the fact that all the $k_i's$ are greater than zero. The joint probability distribution of X and Y is given by

$$Pr[X=k, Y=R+1] = \sum_{\{k_1+k_2+\cdots+k_R=k\}} \binom{N}{k_1} \binom{k_1}{k_2} \cdots \binom{k_{R-1}}{k_R} p^k q^N \qquad \text{(B.2)}$$

where the $k_i's$ satisfy the constraint in equation B.1.

Claim:

$$\sum_{\{k_1+k_2+\cdots+k_R=k\}} \binom{N}{k_1} \binom{k_1}{k_2} \cdots \binom{k_{R-1}}{k_R}$$

is equal to the coefficient of x^k in $(1 + x + x^2 + \cdots x^R)^N$ provided $N \geq k_1 \geq k_2 \geq \cdots \geq k_R \geq 0$ (note that we are allowing the $k_i's$ to be zero here).

Proof: By the binomial theorem,

$$(1+x)^N = \sum_{k=0}^{N} \binom{N}{k} x^k$$

Substituting $x_1(1+x_2)$ for x in the above equation, we get

$$(1+x_1(1+x_2))^N = \sum_{k_1=0}^{N} \sum_{k_2=0}^{k_1} \binom{N}{k_1} \binom{k_1}{k_2} x_1^{k_1} x_2^{k_2} \qquad \text{(B.3)}$$

and putting $x_1 = x_2 = x$ we have

$$(1 + x(1+x))^N = \sum_{k_1=0}^{N} \sum_{k_2=0}^{k_1} \binom{N}{k_1} \binom{k_1}{k_2} x^{k_1+k_2}$$

The left hand side of equation B.4 equals $(1+x+x^2)^N$. Continuing this way, we can expand x_2 in equation B.3 to $x_2(1+x_3)$ and so on. This proves our claim.

In the above derivation, we have allowed the $k_i's$ to be zero. This gives us $Pr[X=k, Y \leq R+1]$. Let us define $A(k,R) = Pr[X=k, Y \leq R+1]$ and $P(k,R) = Pr[X=k, Y=R+1]$. Then

$$P(k, R) = A(k, R) - A(k, R-1) \qquad R = 0, 1, 2, \cdots, k \qquad \text{(B.5)}$$

Now,

$$(1+x+x^2+\cdots+x^R)^N = \frac{\left[1-x^{R+1}\right]^N}{(1-x)^N}$$

$$= \sum_{j=0}^{N}(-1)^j \binom{N}{j} (x^{R+1})^j \sum_{l=0}^{\infty} \binom{N+l-1}{N-1} x^l$$

$$= \sum_{j=0}^{N}(-1)^j \binom{N}{j} \sum_{l=0}^{\infty} \binom{N+l-1}{N-1} x^{l+(R+1)j} \qquad \text{(B.6)}$$

Now putting $k=l+(R+1)j$ and interchanging the order of summation, we have

$$(1+x+x^2+\cdots+x^R)^N = \sum_{k=0}^{\infty} \sum_{j=0}^{min(N,[k/(R+1)]_-)} (-1)^j \binom{N+k-(R+1)j-1}{N-1} \binom{N}{j} x^k \qquad \text{(B.7)}$$

where $[k/(R+1)]_-$ is the largest integer less than or equal to $k/(R+1)$. If we denote the coefficient of x^k in equation B.6 as $C(k,R)$, then

$$A(k,R) = C(k,R)p^k q^N \qquad \text{(B.8)}$$

Equations B.5 and B.8 finally give the probability of exactly R+1 rounds and k errors. Now we can compute

$$E[XY] = \sum_{R=0}^{\infty} \sum_{k=0}^{\infty} (R+1)k\, P(k\ ,R) \qquad \text{(B.9)}$$

The covariance of X and Y is given by

$$Cov(X,Y) = E[XY]-E[X]E[Y] \qquad \text{(B.10)}$$

E[X] and E[Y] have already been computed in section 4. Equation B.9 can be simplified as follows. We define $Q(XY, z, R)$ as follows:

$$Q(XY,z,R) = \sum_{j=0}^{N}(-1)^j \binom{N}{j} \sum_{l=0}^{\infty} \binom{N+l-1}{N-1} (pz)^{l+(R+1)j} \ -$$

$$\sum_{j=0}^{N}(-1)^j \binom{N}{j} \sum_{l=0}^{\infty} \binom{N+l-1}{N-1} (pz)^{l+Rj} \qquad \text{(B.11)}$$

Comparing equation B.11 with equation B.6 we can see that

$$Q(XY,z,R) = (\,1+(pz)+(pz)^2+\cdots+(pz)^R\,)^N \;-\; (\,1+(pz)+(pz)^2+\cdots+(pz)^{R-1}\,)^N$$

$$= \frac{1}{(1-pz)^N}\left[\left[1-(pz)^{R+1}\right]^N - \left[1-(pz)^R\right]^N\right] \qquad \text{(B.12)}$$

On the other hand, from the definition of $P(k, R)$, and from equations B.6, B.7, B.9 and B.11 we can see by inspection that

$$E[XY] = q^N \sum_{R=0}^{\infty} (R+1) \frac{\delta}{\delta z} Q(XY, z, R) \Big|_{z=1}$$

$$= q^N \frac{\delta}{\delta z} \sum_{R=0}^{\infty} (R+1)\, Q(XY, z, R) \Big|_{z=1} \tag{B.13}$$

Using equation B.12, the inner sum on the right hand side becomes

$$\frac{1}{(1-pz)^N} \sum_{R=0}^{\infty} (R+1) \left[\left(1 - \left[1 - (pz)^{R+1}\right]^N\right) - \left(1 - \left[1 - (pz)^R\right]^N\right) \right]$$

Applying the formula for summation by parts to this expression, we get

$$\frac{1}{(1-pz)^N} \sum_{R=0}^{\infty} \left(1 - \left[1 - (pz)^R\right]^N\right)$$

Thus equation B.13 simplifies to

$$q^N \frac{\delta}{\delta z} \left[\frac{1}{(1-pz)^N} \sum_{R=0}^{\infty} \left(1 - \left[1 - (pz)^R\right]^N\right) \right]$$

$$= \frac{Np}{q} \sum_{R=0}^{\infty} \left(1 - \left[1 - p^R\right]^N\right) + N \sum_{R=0}^{\infty} R \left[1 - p^R\right]^{N-1} p^R \tag{B.14}$$

The first term in equation B.14 can be seen from section 4 to be equal to $E[X]E[Y]$. Hence, from equations B.14 and B.10, we have

$$Cov(X,Y) = N \sum_{R=0}^{\infty} R \left[1 - p^R\right]^{N-1} p^R \tag{B.15}$$

For $Np \ll 1$, we can approximate this as

$$Cov(X,Y) \sim N\, (1-p)^{N-1} p$$

and finally, putting $q = 1-p$ we get

$$Cov(X,Y) \sim N\, q^{N-1} p \tag{B.16}$$

Thus, as N gets large, $Cov(X,Y) \rightarrow 0$ because $q<1$. Intuitively, this means that as the number of packets become very large, the relationship between the total number of errors and the total number of rounds becomes weak for any fixed q.

References.

[Ana 86] : Anagnostou M.E. and E. N. Protonotarios, "Performance analysis of the selective repeat ARQ protocol, " *IEEE Trans. Commun.,* vol COM-34, pp. 127-135, Feb. 1986.

[Bru 86] : Bruneel H. and M. Moeneclaey, "On the throughput analysis of some continuous ARQ strategies with repeated transmissions, " *IEEE Trans. Commun.*, vol. COM-34, pp. 244-249, March 1986.

[Gus 87] : Gusella R. , "The Analysis of Diskless Workstation Traffic on an Ethernet," Tech. Rep. UCB/CSD 87/379, University of California, Berkeley, November 1987.

[Jai 86] : Jain R. , "A Timeout-Based Congestion Control Scheme for Window Flow-Controlled Networks, " *IEEE Journal on Selected Areas in Communications*, SAC-4, pp. 1162-1167, October 1986.

[Kan 88] : Kanakia H. and D.R. Cheriton, "The VMTP Adapter Board (NAB): High-Performance Network Communication for Multiprocessors ," SIGCOMM 1988, to appear.

[Moe 86] : Moeneclaey M., "Throughput optimization for a generalized stop-and-wait ARQ scheme, " *IEEE Trans. Commun.*, vol COM-34, pp. 205-207, Feb. 1986.

[Moh 87] : Mohan S. , J. Qian and N.L. Rao, "Efficient Point-To-Point and Point-To-Multipoint Selective-Repeat ARQ Schemes with Multiple Retransmissions : A Throughput Analysis, " *Proceedings of the ACM SIGCOMM '87 Workshop,* Stowe, Vermont, August 11-13, 1987, pp 49-57.

[Tan 81] : Tanenbaum A.S. , *Computer Networks,* Prentice-Hall Inc. , Englewood Cliffs, NJ 07632, 1981.

[Tow 79] : Towsley D. and J.K. Wolf, "On the Statistical Analysis of Queue Lengths and Waiting Times for Statistical Multiplexors with ARQ Retransmission Schemes," *IEEE Trans. Commun,* vol COM-27 April 1979, pp 693-702.

[Tri 82] : Trivedi K.S. , *Probability and Statistics with Reliability, Queuing, and Computer Science Applications,* Prentice-Hall Inc., Englewood Cliffs, NJ 07632, 1982.

[Son 88] : Song C. and L.H. Landweber, "Optimizing Bulk Data Transfer: A Packet Train Approach," SIGCOMM 1988, to appear.

[Zwa 85] : Zwaenepoel W., "Protocols for Large Data Transfers on Local Networks," Tech. Rep. TR-85-23, Department of Computer Science, Rice University.