A SURVEY OF END-TO-END RETRANSMISSION TECHNIQUES

S.W. EDGE & A.J. HINCHLEY

DEPARTMENT OF STATISTICS & COMPUTER SCIENCE
UNIVERSITY COLLEGE LONDON

ABSTRACT

This paper compares two alternative schemes for end-to-end retransmission: use of a sender timeout coupled to positive acknowledgement of data, which is well researched, and use of negative acknowledgement, which is not. The former scheme is the more common; we summarise its characteristics in relation to a set of defined retransmission objectives, and present an algorithm to achieve low retransmission delay. We then discuss the major properties of negative acknowledgement retransmission and show, using a simulation model of a simple end-to-end connection, that for a high rate of packet loss, this is the superior scheme. From the results we obtain, we can then specify the best scheme to use under a number of well defined classes of end-to-end path reliability.

1. Introduction

Retransmission of lost or damaged messages from a sender to a receiver is a basic ingredient of computer network protocols. It can occur at many levels from a simple point-to-point line level to an end-to-end connection across a number of levels of data path, where several mechanisms combine to ensure reliable data transfer.

Specific instances of retransmission are found at line level in protocols such as HDLC (CCITT 76), and between packet switches in a network (McQuillan 77). In a multiple network environment, retransmission might be employed between gateways operating at the edges of the individual networks (Sunshine 77a). Finally, a reliable delivery end-to end protocol may support process-process communication across a one - or many - network path, and such a protocol may also require a retransmission capability. Examples of the latter are: TCP (Cerf 78a), INWG 96 (Cerf 78b), the EIN end-to-end protocol (EIN 76), and the CYCLADES end-to-end protocol (CYCLADES 73).

Other instances of retransmission occur in packet switching over a broadcast medium, and specialised retransmission schemes have evolved for use in broadcast satellite operation (Binder 75, packet radio (Kunzelman 78), and with Ethernet (Metcalfe 76).

This paper concentrates on the requirements of end-to-end protocols. In our terminology, a "transport station" will denote the physical implementation of an end-to-end protocol at a particular site, and end-to-end communication will be between pairs of "processes".

2. Aims of End-to-End Retransmission

The aims of a retransmission mechanism are:

- ensuring reliability
- minimising delay
- minimising redundant duplicate retransmissions
- simple operation

These aims relate to any level of protocol, but particularly at the end-to-end level, where both the delay and variation in delay are scaled up to such a degree that meeting the criteria mentioned above becomes rather more of a critical matter.

The minimisation of delay is important in packet switched networks, where nodal switching delays add up to a significant amount compared to say the delay in a pre-established digital circuit. For example, the transit delay of existing X25 networks is typically one sixth to half a second (Erskine 77, Guilbert 77). However, retransmissions, which will normally be keyed to some message (or lack of it) from the receiver, are likely to be delayed by at least two such intervals.

Minimisation of retransmission overhead is an obvious criteria related to efficient use of resources - both communication resources to carry the unnecessary additional messages, and processor resources in generating and interpreting the messages. In public networks, packet costs will be incurred for such messages, and in large private datagram networks, runaway retransmissions may cause network congestion, which is not easily recoverable. We note that efficient resource utilisation and minimisation of delay are also the goals of flow control (Pouzin 76).

Finally, simplicity is important, both to allow unambiguous definition and to reduce the size and complexity of transport stations.

The diversity of these aims argues against a single scheme for all situations. By broadly classifying different end-to-end reliability levels it can be shown, however, that schemes can be introduced to satisfy the different requirements.

3. Positive Acknowledgement Retransmission

Positive acknowledgement retransmission, using a timeout at the sender, is the most common retransmission scheme adopted. Its main advantage is simplicity: data made available to a receiving process is positively acknowledged by the receiving station; data which times out at a sending station is retransmitted. Reliability is virtually guaranteed in all circumstances short of system crashes, which might permanently remove the necessary state information to maintain the connection (Sunshine 75). In addition, however, careful re-use of either message sequence numbers (Tomlinson 74, Dalal 74) or connection identifiers (Reed 77) will be necessary, in order to reliably detect message arrivals from previous (recent) incarnations of a connection.

Analysis (Sunshine 75) has shown the delay incurred by the need to rely, on occasions, on retransmitted packets, is reduced by using a smaller retransmission timeout. Minimum retransmissions, however, are achieved by employing a timeout equal to twice the maximum packet lifetime for the communication path (if known), thus ensuring that retransmissions are only triggered when a packet is definitely lost or damaged. Since the retransmission delay with such a timeout could be excessive, a smaller "tuned" timeout might be used, where the probability of retransmitting a packet, about to be acknowledged, is kept very small. The value of such a tuned timeout will always exceed the average round trip delay of the transit medium - to avoid runaway retransmission - but the excess could be small (Sunshine 75).

The most serious drawback of positive acknowledgement retransmission is that, under certain conditions, a very high level of redundant duplicate retransmission is unavoidable (McKenzie 74). This is possible whenever two or more data carrying packets are pipelined (i.e. simultaneously outstanding) at a sender. Because these must be acknowledged in sequence (following data delivery to a receive process) the loss of a single packet delays the acknowledgement of all subsequent packets and unnecessarily induces their retransmission. The pipelining of a large number of packets may thus lead to a large ratio of retransmitted packets to packets actually lost. Further, in schemes where a large unit of acknowledgement is used - such as a letter in INWG 96 - unnecessary retransmission will be even higher, since in addition to succeeding packets, those packets preceeding a damaged packet, but belonging to the same acknowledgement unit, will also require retransmission.

The seriousness of the above obviously relates to the level of end-to-end packet loss. For very low loss rates, even when acknowledgement is per-letter, the absolute magnitude of retransmission is likely to be low (Day 75). However, for high loss rates, alternative retransmission schemes may be needed, which reduce the level of redundant retransmission.

3.1 Modifications of Positive Acknowledgement Retransmission

We explore several modifications to positive acknowledgement retransmission, which reduce the number of redundant retransmissions.

Several authors (McKenzie 74, Sunshine 75) have suggested retransmitting only the first packet timed out (repeatedly if necessary) on the assumption that for a low end-to-end loss rate, this will probably be the only packet lost. The main disadvantage of this scheme is that it would have a very high recovery delay whenever a large number of packets were discarded at a receiving station - the occurrence of which can be an important reason for requiring an end-to-end retransmission capability (Cerf 74). We therefore conclude that this is not a good general scheme.

In another scheme, which has been implemented with TCP (Mathis 77), the retransmission interval for each packet commences at some base value and increases linearly or exponentially following each retransmission of the packet. The main purpose is to avoid flooding the subnet and receiver with retransmissions, in the event that packets have to be discarded at the receiver (or in the subnet) and flow control alone is inadequate to quench this flow. We expect this scheme will be most useful with transport stations operating over datagram networks which implement minimal congestion control. It is interesting to note that an increasing retransmission interval may also be used in broadcast networks for a similar purpose (Metcalfe 73).

3.2 Selecting a Retransmission Timeout

As we noted above, the retransmission timeout used with positive acknowledgement is subject to two constraints: it should be large enough to minimise, ot at least considerably reduce, the probability of premature retransmission; and it should avoid unnecessarily high retransmission delays. The importance of the latter is obviously related to the frequency with which retransmission is required. Thus for extremely reliable end-to-end paths, such as occur across the Arpanet or across X25 nets, a single large timeout might be used at a transport station, which can be pre-set to minimise retransmission on all connections. For less reliable paths, such a large timeout would be unsuitable for those connections with a low transit delay. Instead, connections could be partitioned into categories - e.g. satellite, multinet, single net, etc. - based upon their order of round trip time, with a suitable timeout for each category. Finally, for very unreliable paths, the retransmission timeout might have to be tuned individually to each connection, in order to minimise retransmission delay. The following algorithm, which continuously re-evaluates the retransmission timeout from round trip delay measurements, is one method of achieving this.

Each time that a new packet is acknowledged at a sending transport station, the round trip delay t since it was first transmitted is determined, using timestamping information associated with the packet retransmission queue. Then, provided the packet was not retransmitted - when t could be misleading - the average round trip delay estimate T, for the connection, is updated as follows:

$$T := (m.T + t)/(m+1) \qquad m>=o \qquad (1)$$

The value of T obtained thus is affected by all the delays inherent to the connection, and is consequently more useful than knowledge of the delay of the transit medium alone. Appendix 1 shows that the weight m has two features: reduced m reduces the delay in correcting T when the connection round trip delay changes; but increased m reduces statistical fluctuation of T in steady state.

The retransmission interval can be periodically updated from the latest value of T:

$$\text{Retransmission Timeout} := n.T$$

The multiplier n could be determined empirically at each transport station, to reduce premature retransmissions to a reasonable level. Since retransmission overhead will be high anyway for an unreliable connection, a low value for n (e.g. 2 or 3) might not be unreasonable.

4. Negative Acknowledgement Retransmission

Retransmission induced by explicit negative acknowledgement of lost packets by a receiver is far less common, at all levels, than positive acknowledgement retransmission. One example where it is used is in HDLC (CCITT 76), which uses two types of reject command to prompt retransmission (Gelenbe 78). In the end-to-end case, there is a proposal to incorporate negative acknowledgement in the INWG 96 protocol (Cerf 78b).

The major disadvantage of using negative acknowledgement in end-to-end protocols is the complexity this would add. For example, a receiving transport station would have to detect packets lost en route, possibly by timing-out missing message fragments. Furthermore, positive acknowledgement retransmission, with a suitably large timeout (Pouzin 73), would still be required to ensure reliability, in the event that negative acknowledgements or retransmissions were lost. Positive acknowledgement retransmission would probably also be required for interactive traffic, where loss of an isolated packet-size message might go undetected at a receiving transport station. In the latter case, use of a small sender timeout to hasten positive acknowledgement retransmission could be integrated with negative acknowledgement by restarting the timeout of a packet, whenever it or a packet with lower sequence than itself was negatively acknowledged.

The main advantage of negative acknowledgement retransmission is to reduce redundant retransmission. For example, negative acknowledgement of missing "letter" fragments (see Section 5) might be used in INWG 96, in addition to positive acknowledgement of whole letters. With pipelined traffic, this additional means of re-supplying lost packets would substantially reduce the likelihood (discussed in Section 3) of unnecessarily retransmitting successors to - or packets in the same letter as - a damaged packet. As an alternative, the unit of positive acknowledgement in INWG 96 could be reduced - e.g. to letter fragments - to avoid retransmitting a whole letter whenever a letter portion was lost, but this would not avoid retransmitting packet successors when traffic was pipelined.

5. Comparison of Positive and Negative Acknowledgement Retransmission

We illustrate the points made above about each basic retransmission scheme with some simulation and analytical results for an example model of an end-to-end connection. The model, we will describe, comprises a single process-to-process connection, maintained by a pair of transport stations (TS's), where data flow is one way from a Send Process (source) to a Receive Process (sink). The data transport mechanism, under review, is a simple one common to many end-to-end protocols, such as TCP(Cerf 77) and INWG 96 (Cerf 78b). Figure 1 illustrates the connection.

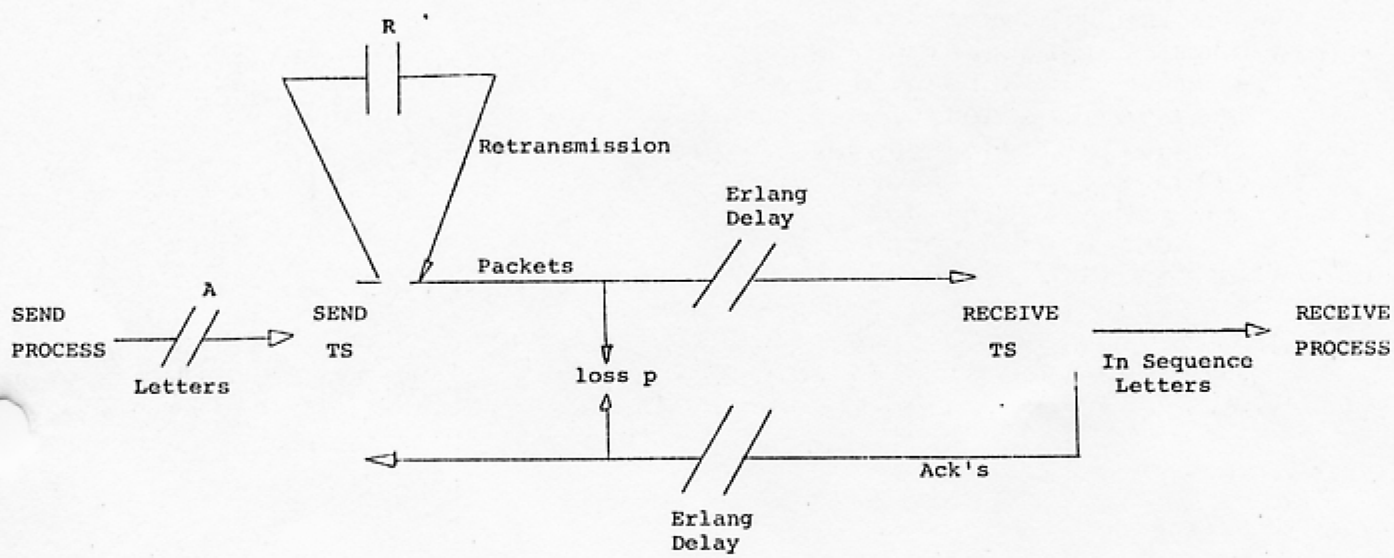

Figure 1. End-to-End Connection Model

An infinite stream of fixed size "letters" is passed one at a time and at constant time intervals from the Send Process to the Sending TS. Each letter is encapsulated in a single packet, which is assigned a monotonically increasing letter sequence number, and transmitted immediately. There is no restriction on flow, corresponding to an infinite window size and unlimited internal buffering in the case of TCP and INWG 96. Further, packets are not fragmented (e.g. at gateways) during transit. The transit delay between the two TS's has an Erlang distribution with parameter k and mean 1/u (which we set to unity). The probability density function f for a delay x is:

$$f(x) = (k.u)^{k}.x^{k-1}.e^{-kux}/(k-1)! \qquad x>=0 \qquad (2)$$

This models a wide range of delay distributions, from exponential (k = 1) to constant (k approaches infinity). The coefficient of variation of delay is $k^{-\frac{1}{2}}$. We will use two values of k, k=25 and k=4, to model low and high delay variability respectively. Loss of packets in the transit medium (or at the receiving TS) is represented by applying a fixed loss probability to each packet in transit independently. The delay distribution and loss are identical in either direction. The receiving TS buffers out of order letter arrivals, and whenever possible, passes letters in sequence to the receive process - which can always accept them. Further, a single acknowledgement packet is returned immediately to the sending TS, for all letters delivered at the same time (this also re-acknowledges all earlier letter deliveries) and carries the sequence number of the next letter to be delivered. Such an acknowledgement is also returned for each duplicate letter arrival; the latter then being discarded. At the Send TS, unacknowledged letters are retransmitted after a constant timeout interval, since their previous (re)transmission, and they are discarded upon acknowledgement. Internal TS processing delays are ignored, but may be regarded as comprising part of the transit medium delay.

The unit of time is the mean TS-TS transit delay (1/u in eq.2). The model then has the following parameters:

- k : parameter of Erlang delay distribution
- A : inter-letter arrival interval at the send TS
- p : probability of packet loss in the transit medium
- R : send TS retransmission interval

The model described so far is used to illustrate positive acknowledgement retransmission. We illustrate negative acknowledgement with the following additional mechanism, which has been chosen for its simplicity. The Receive TS buffers and delivers arriving letters as before. However, each arriving letter commences a timeout period if its immediate sequential predecessor has not yet arrived. If the latter has not arrived on timeout, a negative acknowledgement is returned to the Send TS for all the immediately preceeding non-arrivals - up to but not including the highest sequence preceeding letter which has arrived. In practice, only the highest and lowest sequence numbers of these would be physically carried. At the Send TS, negatively acknowledged letters are immediately retransmitted, independently of positive acknowledgement retransmission (which operates as before). When implemented, this scheme requires a new parameter:

Tnack = Timeout at the receive TS

The first example we illustrate with this model concerns the average number of times letters are retransmitted with positive acknowledgement retransmission operating only, for different values of the retransmission timeout R. Simulation results for this are shown in Fig. 2. The first point to notice is that for a large enough R, retransmission is minimised for each case considered, as we stated earlier. Tuned R - discussed in Section 3 - is the minimum value of R giving this minimisation, and we will call this value Rtun.. It clearly exceeds the average normalised round trip delay (which is 2) in every case, and is increased by higher delay variability (reduced k). R less than Rtun causes premature letter retransmission, and the rise of this (with reduced R) becomes more stepwise as acknowledgement delay approaches a constant (k approaches infinity, p approaches zero). Excessive retransmission for pipelined traffic - discussed in Section 3 - is illustrated for the case k=25, p=.1, where the minimum level of retransmission at .54 substantially exceeds the minimum requirement of .11, when only lost packets are retransmitted.

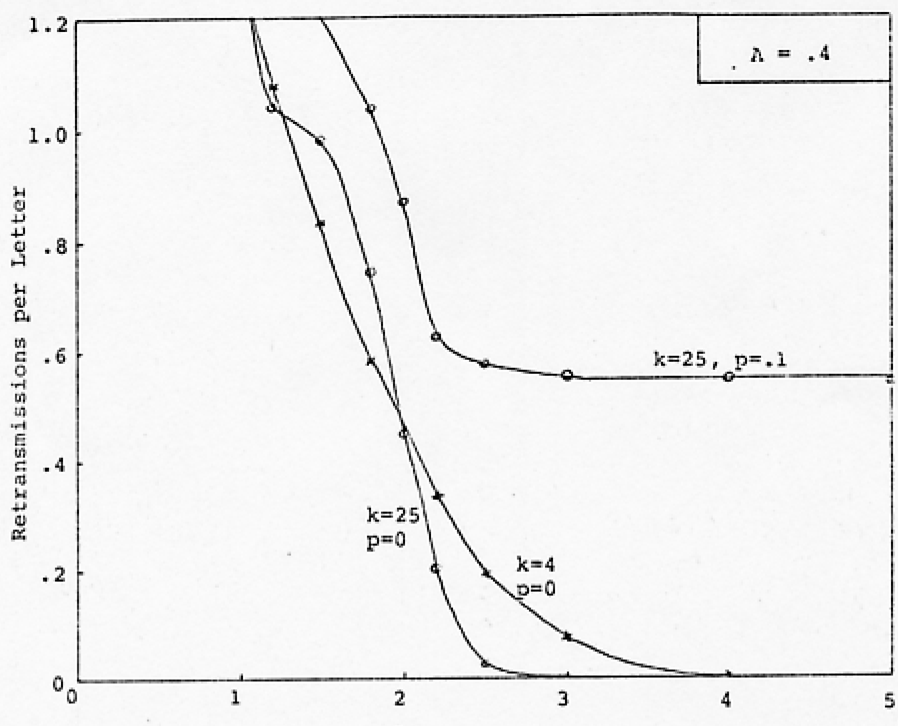

Figure 2. Average Retransmissions per Letter versus Sender Timeout

Next we look at this situation from the point of view of negative acknowledgement. Figure 3 shows simulation results relating the value of the timeout Tnack used at the receiver to the extent of redundant retransmission when there is no packet loss (p=O). The results are analagous to those in Fig. 2. A low value of Tnack increases the likelihood of prematurely negatively acknowledging letters, which have not arrived, thereby producing redundant retransmissions. Conversely, for large enough Tnack, redundant retransmission ceases. We can thus determine a tuned Tnack value to reduce the retransmission delay when packets are lost, in the same way that Rtun was found above. Figure 3 shows that its value depends on the variability of the transit medium delay (k) and the traffic rate (A), each of which affects the extent to which letters can arrive out of order.

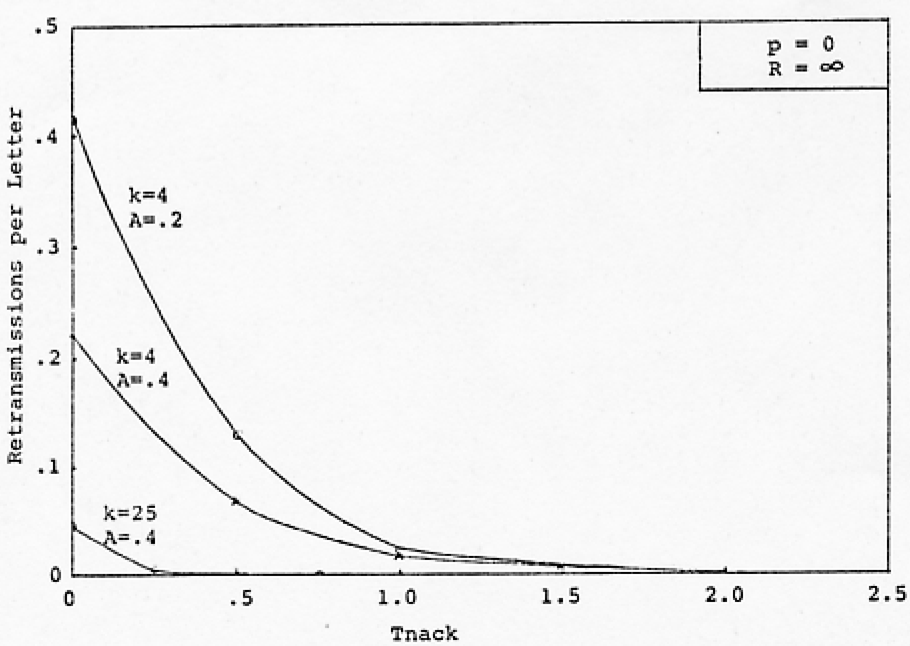

Figure 3. Average Retransmissions per Letter versus Receiver Timeout

The preceeding results allow comparison of the minimum (tuned) retransmission delay inherent in either scheme. For positive acknowledgement, this is simply Rtun. For negative acknowledgement, an approximate retransmission delay is obtained by assuming that the immediate sequential successor to a lost letter arrives safely at the receiver, and after timeout and return of a negative acknowledgement, induces the retransmission (i.e. p is small, k is large). Thus:

$$\text{av. retransmission delay} = \text{av. round trip delay} + \text{Tnack} + \text{T}$$

$$= 2 + \text{Tnack} + \text{T}$$

For the case p=0, k=25, A=.4, Rtun is 3 (Fig. 2), giving equality of retransmission delay for tuned Tnack =.6 (Fig. 3). The case p=0, k=4, A=.4 is similar, providing near equality.

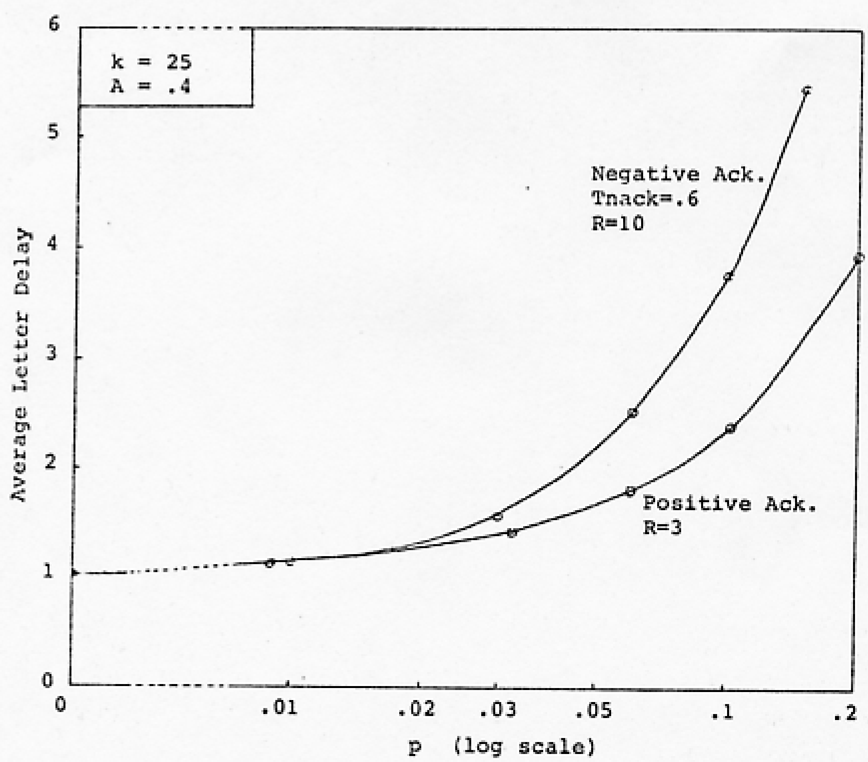

Figure 4. Average Letter Delay versus Packet Loss

The above comparison implies that delays in either scheme should be similar. We investigate this in Fig.4, which shows the average letter delay, from the Send to the Receive Process, against packet loss. Simulation results are shown for negative acknowledgement, whereas for positive acknowledgement we use analytical results, from Appendix 2, whose accuracy slightly exceeds that obtained from simulation. We re-use the tuned timeouts quoted above, so it is not surprising to find in Fig.4 almost identical delays for a low loss rate. At higher loss rates, the delay for negative acknowledgement becomes much larger, because substantially more lost packets require positive acknowledgement retransmission, following the loss of a negative acknowledgement or retransmission, and this uses a large untuned timeout (R=10). The latter setting reflects the reduced significance of the sender time-out with negative acknowledgement retransmission, although we would obviously expect to reduce delay by employing a smaller sender timeout.

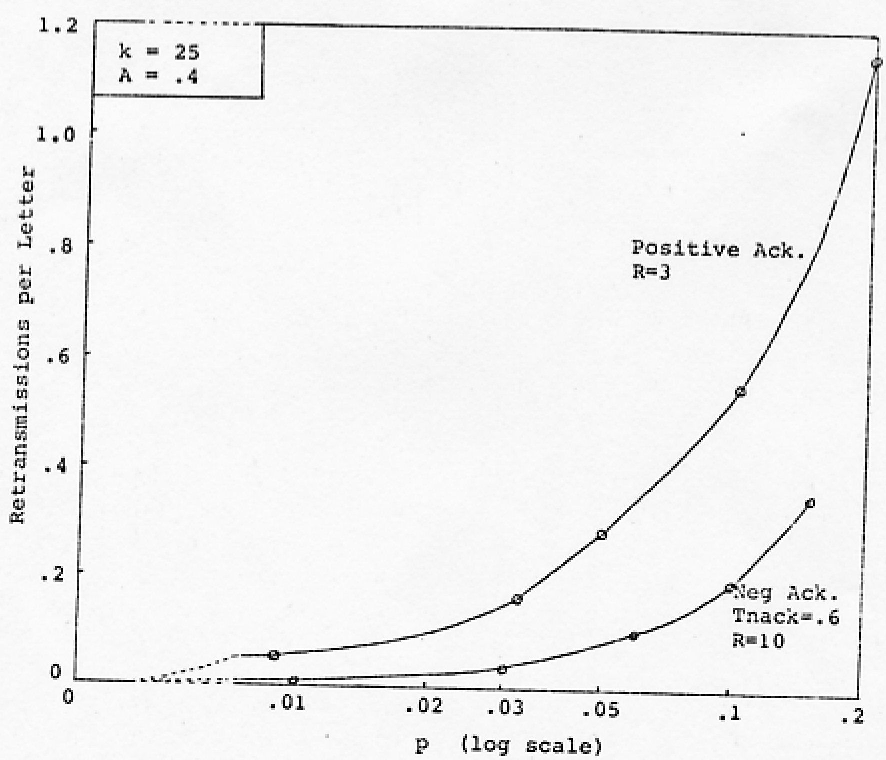

Figure 5. Average Retransmissions per Letter versus Packet Loss

In Fig. 5, we compare the retransmission overhead of each scheme - obtained by simulation - over the same range of loss rates used in Fig. 4. For low loss rates, retransmission in either scheme is low. However, it is clear that negative acknowledgement produces far fewer retransmissions than positive acknowledgement, and the magnitude of the difference becomes more significant for increased packet loss.

6. Selection of a Retransmission Scheme

We evaluate here the two basic schemes we have been comparing, in the context of three levels of end-to-end reliability.

6.1. Highly Reliable End-to-End Path

In this case we assume that lower level mechanisms provide reasonably reliable communication. An X25 virtual call network is an obvious example, where the bit error probability can be as low as 10^{-10} (Danet 76). Such a loss is acceptable for most network use, but where a guaranteed process-to-process reliability several orders of magnitude better is required, we may expect to superimpose an additional reliable end-to-end protocol. As an example, a version of the INWG 96 protocol adopted specifically for use above X25 networks is currently being produced by an IFIP working group (Cerf 78b).

It is clear that the frequency of necessary retransmission is sufficiently low that positive acknowledgement retransmission using a long timeout, will be adequate.

6.2. Moderately Reliable End-to-End Path

We reserve this case for end-to-end packet loss probabilities lying between 10^{-2} and 10^{-6}, the range likely to be found in most datagram networks, or where a receiving transport station may occasionally discard packets due to flow control (Cerf 74). Here retransmission will be required too infrequently to make the reduction in retransmission overhead, possible with negative acknowledgement, worthwhile. Positive acknowledgement will thus be suitable, and the value of the timeout may be set by categorising connections, as discussed in Section 3.2

6.3. Unreliable End-to-End Path

This case is for packet loss probabilities less than 10^{-2}. Such a high loss rate is obviously atypical, because in nearly all practical situations, lower level mechanisms reduce the loss rate seen at the end-to-end level. However, it may occur in special circumstances, such as where a receiving station uses inadequate buffering and must frequently discard arriving packets (Edge 77).

Provided a high level of retransmission overhead is acceptable, positive acknowledgement retransmission could be used here, and the retransmission timeout would probably require tuning - as described in Section 3.2 - to reduce delay. Alternatively, the simulation results discussed in Section 5 show that a negative acknowledgement scheme could substantially reduce the level of unnecessary retransmission at the cost of somewhat larger delay.

7. Conclusions

We conclude that positive acknowledgement retransmission - as employed in many existing end-to-end protocols - is most suitable for the majority of end-to-end connections, namely those with moderate or high reliability. For unreliable connections, negative acknowledgement retransmission may be preferable, because this greatly reduces the extent of redundant retransmission, which can be large with just positive acknowledgement. The drawbacks to negative acknowledgement are added complexity and probable higher delay. Optimal performance in either scheme requires careful selection of timeout values, and for unreliable connections, these should be tuned to the individual connection.

8. Acknowledgements

We are grateful to Dr. Carl Sunshine and Professor P.T. Kirstein for some helpful comments during the preparation of this paper.

REFERENCES

Binder 75 : "The Atlantic Satellite Broadcast Experiments" - R.Retburg, D.Walden - BBN Report 3056, April 1975.

CCITT 76 : "Draft Recommendation X-25" -AP V1-No. 55-E, CCITT, 1976.

Cerf 74 : "A Protocol for Packet Network Intercommunication" - V. Cerf, R.Kahn - IEEE Trans on Communications, p.637, May 1974.

Cerf 78a : "Specification of Internet Transmission Control Program - TCP Version 3" - V.Cerf, J.Postel - Information Sciences Institute, Univ. of Southern California, January 1978.

Cerf 78b : "Proposal for an Internetwork End-to-End Transport Protocol" - V.Cerf, A.McKenzie, R.Scantlebury, A.Zimmermann -Computer Network Protocols Symposium, Liege, Belgium, p.H-5, February 1978.

CYCLADES 73 : "Specifications Functionelles des Stations de Transport du Reseau Cyclades Protocols ST-ST" - Reseau Cyclades SCH.502.3, Institut de Recherche D'Informatique et D'Automatique, Rocquencourt, France, May 1973.

Dalal 74 : "More on Selecting Sequence Numbers" - Y.Dalal - INWG Protocol Note 4, October 1974.

Danet 76 : "The French Public Packet Switching Service: The Transpac Network" - A.Danet, R.Despres, A.Le Rest, G.Pichon, S.Ritzenhaler - ICCC Proceedings, Toronto, p.251, 1976.

Day 75 : "A Note on Some Unresolved Issues for an End-to-End Protocol" - J.Day - INWG General Note 98, August 1975.

Edge 77 : " Buffer Management Strategies for Communications Network Host Protocols with particular Reference to TCP" - S.W.Edge, A.J.Hinchley - Indra Note 611, Dept. of Statistics & Computer Science, UCL, March 1977.

EIN 76 : "End-to-End Protocol" - EIN Publication EIN/76/003.

Erskine 77 : "Datapac: A Packet Switching Network for Canada" - S.B.Erskine - Online Conference Proceedings, Uxbridge, England, p.25, May 1977.

Gelenbe 78 : "Performance Evaluation of The Protocol HDLC" - E.Gelenbe, J.Labetoulle, G.Pujolle - Computer Network Protocols Symposium, Liege, Belgium, p.G-3, February 1978.

Guilbert 77 : "The Transpac Network" - J.F.Guilbert - Online Conference Proceedings, Uxbridge, England, p.15, May 1977.

Kunzelman 78 : "Overview of the Arpa Packet Radio Experimental Network" - R.C.Kunzelman - IEEE CompCon, San Francisco, p.157, Spring 1978.

Mathis 77 : "Single Connection TCP Specification" - J.Mathis - Packet Radio Network Development, Quarterly Technical Report, Appendix B, February-April 1977.

McKenzie 74 : "Internetwork Host-to-Host Protocol" - A.McKenzie - INWG General Note 74, December 1974.

McQuillan 77 : "The ARPA Network Design Decisions" - J.M.McQuillan, D.C.Walden - Computer Networks, Vol 1, No. 5, p. 243, August 1977.

Metcalfe 73 : "Packet Communication" - R.M.Metcalfe - MAC TR-114, MIT, Project MAC,(Ph.D. Thesis), December 1973.

Metcalfe 76 : "Ethernet: Distributed Packet Switching for Local Computer Networks" - R.M.Metcalfe, D.R.Boggs - Communications of the ACM, vol 19, No. 7, p.395, July 1976.

Pouzin 73 : "Efficiency of Full-Duplex Synchronous Data Link Procedures" - L.Pouzin - INWG General Note 35, 1973.

Pouzin 76 : "Flow Control in Data Networks - Methods and Tools" - L.Pouzin - ICCC Proceedings, Toronto, p.467, 1976.

Reed 77 : "The Initial Connection Mechanism in DSP" - D.P.Reed - Local Network Note 10, MIT Laboratory for Computer Science, August 1977.

Sunshine 75 : "Interprocess Communication Protocols for Computer Networks" - C.A.Sunshine - TR-105, Digital Systems Lab, Stanford University, (Ph.D. Thesis), December 1975.

Sunshine 77a : "Interconnection of Computer Networks" - C.A. Sunshine - Computer Networks Vol 1, No. 3, p.175, January 1977.

Sunshine 77b : "Efficiency of Interprocess Communication Protocols for Computer Networks" - C.A.Sunshine - IEEE Trans on Communications, p.287, February 1977.

Tomlinson 74 : "Selecting Sequence Numbers" (DRAFT) - R.Tomlinson - INWG Protocol Note 3, August 1974.

APPENDIX 1 - Analysis of an Algorithm to Estimate Round Trip Time

The algorithm updates the current round trip delay estimate T_n, for a connection, with successive round trip measurements t_n according to either of the following equivalent expressions:

$$T_{n+1} := (m.T_n + t_n)/(m+1) \qquad m>= o \quad (3)$$

$$= T_n + (t_n - T_n)/(m+1) \qquad (4)$$

If the actual round trip delay changes suddenly, and the latest measurement t_n is a better estimate than T_n, then Eq.4 shows T moves closer to the more correct value for smaller m. Conversely, in steady state, after a sequence $t_O, t_1, \ldots, t_n$ of updates, T_{n+1} is given by (using Eq.3):

$$T_{n+1} = (m+1)^{-1}.(t_n + m(m+1)^{-1}t_{n-1} + m^2(m+1)^2 t_{n-2} + \ldots + m^n(m+1)^{-n}t_O + m^{n+1}(m+1)^{-(n+1)}T_O)$$

If we regard each t_i as a random variable, and make the approximation that they are independent and identically distributed (whence the steady state assumption), then simple expressions for the expectation (E) and variance (var) of T_{n+1} can be obtained for the case n approaches infinity:

$$E(T_{n+1}) = E(t_i) \qquad \text{for each } i$$

$$\text{var}\ (T_{n+1}) = (m+1)^{-2}.(\text{var}(t_n) + m^2(m+1)^{-2}\text{var}(t_{n-1}) + m^4(m+1)^{-4}\text{var}(t_{n-2}) + ..)$$

$$= \text{var}(t_i)/(2m+1) \qquad \text{for each } i$$

The last expression above shows that statistical fluctuation of T_n is reduced for larger m.

APPENDIX 2 - Average Letter Delay with Positive Acknowledgement Retransmission

We use previous analytical study (Sunshine 75 and Sunshine 77b) to calculate the average letter delay for the connection model defined in Section 5. The density function f for letter transit delay (x) is, from Eq.2, Section 5:

$$f(x) = (k.u)^k.x^{k-1}.e^{-kux}/(k-1)! \qquad x>=0$$

The probability distribution F is:

$$F(x) = 1 - e^{-kux}.\sum_{j=0}^{k-1}(kux)^j/j! \qquad x>=0$$

We now add in the effects, successively, of the subnet loss probability p, the retransmission interval R, and the requirement of sequenced letter delivery. The probability F′ of letter arrival after time x, with loss, is:

$$F'(x) = (1-p).F(x)$$

The probability G of arrival, with retransmissions, is:

$$G(x) = 1 - \prod_{j=0}^{n}(1 - F'(x-jR)) \qquad ;\ n=\lfloor x/R \rfloor$$

The probability H of arrival with all predecessors present is:

$$H(x) = \prod_{j=0}^{\infty} G(x+jA) \qquad ;\ A = \text{inter letter arrival time}$$

Average letter delay is:

$$\text{average letter delay} = \int_0^{\infty}(1-H(x)).dx$$