Gary Stroebel

Message Reassembly Times in a Packet Network

This paper addresses the problem of computing the reassembly time of a multipacket message. All packets from a single message are assumed to flow in sequence along the same physical path. The analysis includes the effects of contention between messages in the network on the delay time at each station along the path and its impact on message reassembly time.

Introduction

A common procedure in the transmission of a message from a source node to a destination node through a communications network is to divide the message created at the source into several smaller frames or packets. Certain header information, such as destination address and sequence number, is then attached to each packet. In this way, each packet can flow through the network as an independent flow unit.

In some communications networks (such as ARPANET [1]), the packets from a single message may flow from source to destination via several different physical paths. With this scheme, the packets may arrive at the destination "out of order." With explicit routing (such as the current Systems Network Architecture [2] implementation), all packets from a single message flow in sequence along the same physical path.

In this paper, only the case of explicit routing is addressed. A typical routing path is depicted in Fig. 1. In order to travel from the source to the destination, each packet must pass through N intermediate stations. These stations correspond to the various control units and transmission lines that physically connect the source to the destination node.

Suppose a message with m packets arrives at the first station. This station spends some time processing the first packet. Upon completion of this processing, this packet moves to the second station. Then, while the second station processes the first packet, the first station processes the second packet of the message, and so on. Thus, there is considerable overlap (or pipelining) associated with the message transmission. However, there is not complete overlap because station processing times are not all equal.

Sometime after the arrival of the message at the first station, the leading packet reaches the destination node. This time is easily calculated as the sum of the N station processing times. A more difficult problem, however, is to determine the message reassembly time, *i.e.*, that time between the arrival of the first packet at the destination node and the final transmission of the mth packet. This is the problem that this paper addresses.

This problem has received attention in the literature (*e.g.*, Kleinrock [1] and Miyahara *et al.* [3]). However, these studies have introduced a number of approximations and simplifications. By means of a completely different approach, we attempt to remove some of these limitations. Examples of these limitations include the requirement that all lines have the same speed, all packets have the same mean length, and the arrival process at a service station is Poisson.

Multipacket message transmission delays

In Fig. 1, we consider an m-packet message created by a particular source node. In the transmission of this message from the source to a destination node, each packet must pass through a series of N stations. The packet response time R_i at station i is a random variable with known mean. This response time involves the packet service time plus the waiting time for packets from other messages. This response time *does not* include any waiting that a given packet may experience due to other packets from the same message.

The average response times, as defined above, can be computed from a mean value analysis of a queueing network similar to that described in Bard [4]. This analysis is particularly useful since it separates the total mean response time into three components: namely, the mean service time, mean waiting time for jobs (packets) in other chains, and mean waiting time for jobs in the same chain. As the packet response time is defined in this section, only the first two components are used. The reason for not including the waiting time for packets from the same message will become apparent later.

The average end-to-end message transmission time D consists of two parts. The first is the average time for the first packet to flow from the source to the destination. This is simply

$$D_p = \sum_{i=1}^{N} \bar{R}_i . \tag{1}$$

The second part is the average message reassembly time D_m, *i.e.*, the time for the remaining $(m - 1)$ packets of the message to arrive at the destination node. The remainder of this paper is directed towards a method for estimating D_m.

We first define a packet pair. A packet pair consists of two adjacent packets of the same message. We introduce the notion of a leading packet and a trailing packet. In the transmission of a packet pair, the trailing packet is never allowed to pass the leading packet; *i.e.*, the packet pair is assumed to arrive in the same sequence that it was sent. We also note that an m-packet message consists of $(m - 1)$ packet pairs.

We now find an expression for the average reassembly time of a packet pair. This is simply the time between the arrival of the leading packet and the arrival of the trailing packet at the destination node. Once this expression is determined, the average message reassembly time is readily calculated.

Again we refer to Fig. 1 and define the "state" of a packet pair by the 2-tuple (n, b) where

n = the station location of the leading packet, and

b = the number of stations back from n for the location of the trailing packet.

State transitions are observed only at those time instants (epochs) that the leading packet moves to the next

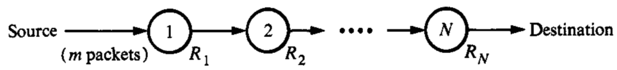

Figure 1 Flow of an m-packet message.

station. Thus, state transitions are of the form

$$(n, b) \rightarrow (n + 1, b + 1 - j) ,$$

where j is the number of stations that the trailing packet has traversed while the leading packet has moved from station n to $n + 1$. We assume that the sequence of states is a Markov chain.

The initial state could be defined as (1, 0). However, since this state flows to the state (2, 1) with probability 1, the initial state of our analysis is taken as (2, 1). We define the destination node as station $N + 1$.

The following bounds then apply to the state descriptors defined above:

$$2 \le n \le N + 1 ,$$

$$1 \le b \le N ,$$

$$0 \le j \le b .$$

We define

$q_j(n, b)$ = probability that state transition from (n, b) is to state $(n + 1, b + 1 - j)$, where $0 \le j \le b$.

For the time being, we assume the $q_j(n, b)$ are known for all valid states (n, b). From these values, we can determine the state transition probability matrix $\mathbf{A}$.

We define

$p^{(i)}(n, b)$ = probability of the packet pair being in the state (n, b) after i state changes.

We note that

$$p^{(0)}(n, b) = 0$$

except for

$$p^{(0)}(2, 1) = 1 .$$

It is easily established that

$$\mathbf{p}^{(i+1)} = \mathbf{p}^{(i)} \mathbf{A} = \mathbf{p}^{(0)} (\mathbf{A})^{i+1} , \tag{2}$$

where $\mathbf{p}^{(i)}$ is a vector of state probabilities after i state changes.

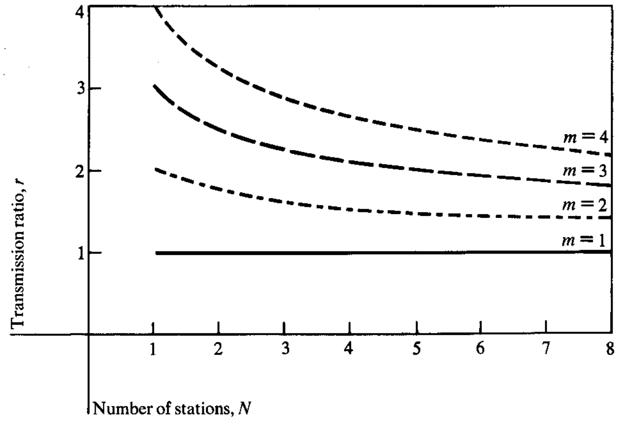

Figure 2 All stations with same mean response time.

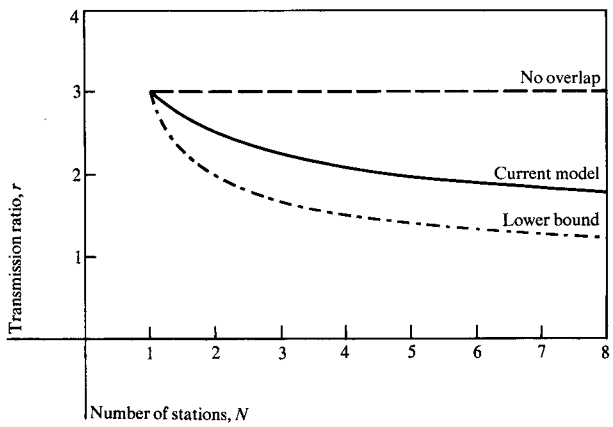

Figure 3 All stations with same response time and $m = 3$.

After $(N - 1)$ transitions from the state (2, 1), the leading packet will arrive at the destination node (station $N + 1$). We determine the probability that the trailing packet is b stations behind at this point.

$$p_b = p^{(N-1)}(N + 1, b), \qquad 1 \le b \le N . \tag{3}$$

Now if the trailing packet is b stations behind when the leading packet arrives at the destination node, the packet pair average assembly time is

$$T_b = \sum_{i=N-b+1}^{N} \bar{R}_i , \tag{4}$$

so that the packet pair average assembly time becomes

$$T = \sum_{b=1}^{N} p_b T_b . \tag{5}$$

Then for an m-packet message, there are $(m - 1)$ packet pair assemblies. Therefore, using Eqs. (1) and (5), the average end-to-end message transmission time becomes

$$D = D_p + (m - 1)T. \tag{6}$$

The calculation of p_b (and, therefore, T) is based upon the state transition matrix **A** through Eq. (2). The matrix **A**, in turn, is based upon the previously defined $q_j(n, b)$. In the following, we derive expressions for $q_j(n, b)$.

We recall our definition of R_i as the packet response time at station i. It includes the packet service time plus waiting time for packets from other messages. It does not include any waiting time that a particular packet may experience due to other packets from the same message.

As such, we assume that both the leading and trailing packets draw their random response time R_i at station i from the same distribution. Thus, we need not distinguish between a R_i for the leading and trailing packets. Note that this would not be the case if we had included waiting for packets of the same message.

Given a state (n, b) with transitions to states $(n + 1, b + 1 - j)$ for $0 \le j \le b$ we define

$x_j(n, b)$ = probability that the leading packet moves from station n to $n + 1$ before the trailing packet completes j hops.

As such

$$x_j(n, b) = \text{Prob}\left[R_n \le \sum_{i=0}^{j} R_{n-b+i}\right] \qquad (0 \le j \le b - 1) \tag{7}$$

and

$$x_b(n, b) = 1 , \tag{8}$$

since we do not allow passing of packets.

If we assume the R_i are independent and exponentially distributed with mean $\bar{R}_i$, it is relatively easy using the convolution theorem to show that

$$x_j(n, b) = 1 - \prod_{i=0}^{j} \frac{\bar{R}_n}{\bar{R}_n + \bar{R}_{n-b+i}} \tag{9}$$

The $q_j(n, b)$ are then related to the $x_j(n, b)$ by

$$q_j(n, b) = x_j(n, b) - x_{j-1}(n, b) \tag{10}$$

for $1 \le j \le b$, where $x_{-1} = 0$.

From Eqs. (9) and (10), the terms of the state transition matrix are readily calculated.

An APL program has been written to do the calculations associated with this analysis. In the following, we provide two examples to get a feeling for the type and the reasonableness of the results of this analysis.

For the first example, we suppose that all N stations have the same average packet response time $\bar{R}_i$. We define a transmission ratio

$$r = \frac{D}{\sum_{i=1}^{N} \bar{R}_i}, \tag{11}$$

where D is given by Eq. (6).

In Fig. 2, this ratio is plotted as a function of the number of stations between the source and destination for various message sizes. We see that the ratio of message transmission time to packet transmission time decreases as the number of service stations increases. This is expected since more overlap occurs as more stations are added. Furthermore, this ratio decreases faster with N for larger messages, again because of a greater degree of overlap.

An absolute lower bound on the end-to-end message transmission time can be obtained with the assumption that each station response time is a constant. In this way, the packets move in perfect sequence from source to destination and achieve a maximum degree of overlap. This lower bound result is shown in Fig. 3 and compared with the results of the current model for the case of a three-packet message.

For the second example, we assume that one station response time is 10 times that for all other stations. With our model, it does not make any difference which of the N stations has the long response time. Results for this case are shown in Fig. 4. Again, a lower bound can be found by assuming that all station response times are constants. This lower bound, in comparison with the current model results, is shown in Fig. 5.

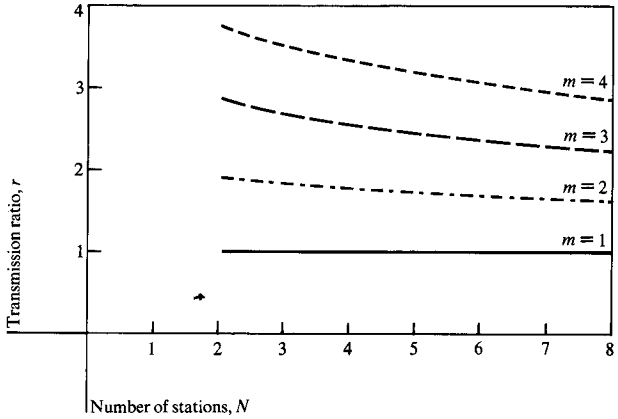

Figure 4 One station with long response time.

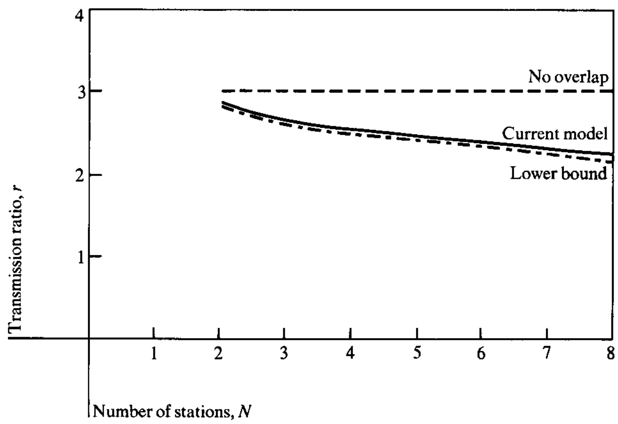

Figure 5 One station with long response time and $m = 3$.

References

1. L. Kleinrock, *Queueing Systems, Volume II: Computer Applications*, John Wiley & Sons, Inc., New York, 1976.
2. J. H. McFadyen, "Systems Network Architecture: An Overview," *IBM Syst. J.* **15**, 4–23 (1976).
3. M. Miyahara, Y. Teshigawara, and T. Hasegawa, "Delay and Throughput Evaluation of Switching Methods in Computer Communication Networks," *IEEE Trans. Commun.* **COM-26**, 337–344 (1978).
4. Y. Bard, "Some Extensions to Multiclass Queueing Network Analysis," *Performance of Computer Systems*, M. Arato, A. Butrimenko, and E. Gelenbe, Eds., North-Holland Publishing Company, Amsterdam, 1979, pp. 51–61.

Received April 14, 1981; revised June 19, 1981

The author is located at the System Products Division laboratory, 3605 Highway 52, North, Rochester, Minnesota 55901.