To be presented at the National Telecommunications Conference, Dallas, Texas, November 1976

ON THE EFFECT OF PERIODIC ROUTING UPDATES IN PACKET-SWITCHED NETWORKS*

#184

William E. Naylor and Leonard Kleinrock
Computer Science Department
University of California, Los Angeles

Abstract

In a packet switched network, some form of adaptive routing procedure is desirable so that packets may be routed around line and node failures and possibly around congestion in the network. It is clear that there must be some overhead associated with any form of adaptive routing (i.e., the channel time and processor time required to transmit, process and generate the routing information). Clearly, one would hope that the cost for such adaptive routing does not exceed the benefits derived therefrom. Since adaptive routing is considered to be necessary in practice, its overhead has received only partial consideration by most authors [4,6,12,15,17]. In this paper, we study some unusual phenomena caused by the interference of routing updates. Specifically, we consider the cost (in terms of message delay) of the current ARPANET routing update procedure. We begin by presenting some results of a set of measurement experiments which prompted an analysis of the effects of the routing procedure on message delay. A simplified model of the system is then discussed and analyzed exactly. This exact solution is a bit unwieldy for highly detailed models and so in this case, we resort to simulation to show the performance and to demonstrate the effect of modified routing schemes. The simulation results indicate a rather high cost associated with the use of periodic routing updates in networks of the size of the ARPANET. This suggests the use of a "passive" routing scheme with catastrophe-triggered updates.

Measurement

We set out to determine by what means and how accurately one could predict round-trip network delay, for a stream of messages with fixed interarrival times, based on previous delay samples in the ARPANET. This traffic pattern is exhibited by fixed data rate sources such as speech [8]. These measurement experiments were conducted with no intention of considering the effects of the periodic update scheme used in the ARPANET. However we observed a much lower than expected correlation between successive delays. In experiments sending data as fast as possible, successive delays display higher correlation [11]. A closer look (suggested by D. Cohen of the Information Sciences Institute, University of Southern California) revealed some interesting phenomena regarding the effect of periodic routing udpate procedures.

The experiments took place on Friday evening December 12, 1975, between the hours of 9 and 11 p.m. PST. (A light network load and therefore nearly constant delay was expected during this time period.) Full single-packet messages were sent from the UCLA PDP 11/45 to a "discard fake HOST" (a portion of the ARPANET IMP software which mimics a "real HOST" acting as a sink for a message stream, see [1]) over (minimum hop path) distances of 1, 2, 5 and 10 hops at fixed interdeparture times of 124, 165, and 248 msec. The round-trip delay (i.e., the delay from the time the message is ready to be sent until the RFNM is returned) was measured by the PDP 11/45 and recorded for subsequent study.

Figures 1 through 4 show some of the round-trip delay measurements plotted against message sequence number (i.e., time). Here we show network delay as a function of (message arrival) time. These particular samples are representative of the collection of experiments.

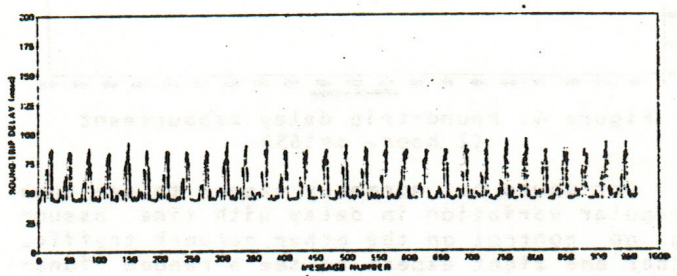

Figure 1. Round-trip delay measurement (1 hop, s=124)

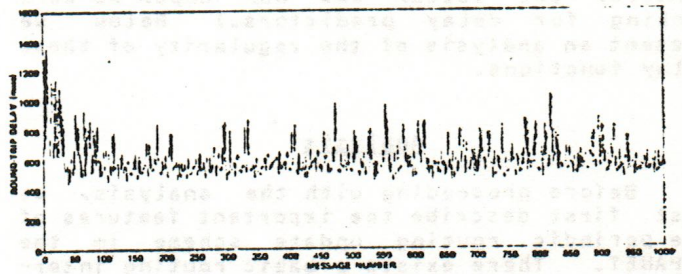

Figure 2. Round-trip delay measurement (10 hops, s=124)

Notice the unusual increases in delay at regular intervals (of about 30 messages) in Figure 1. In each interval there is a group of three dominant peaks separated by two regularly spaced points where the delay is near the minimum. Notice also the regular decrease in the first peak in each group with time until it is replaced with a full sized peak at intervals of five groups. This curve is in fact a periodic function (with some "noise" due to the "other" background data traffic) with a period of about 150 messages!

This periodic behavior is less noticeable at longer network distances. There is a pattern of climbing to a local maximum and suddenly dropping and starting the climb again in Figures 2 and 3. Generally speaking increases in delay are gradual while decreases are immediate (though the opposite condition occasionally occurs as well). The curve shown in Figure 4 varies quite severely. There is a pattern however, and very close examination

---

* This work was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC-15-73-C-0368.

reveals a periodic function (again with some noise) with a period of approximately 130 messages. Plots of other samples show that the shape of the curves is more related to the data rate than to network distance.

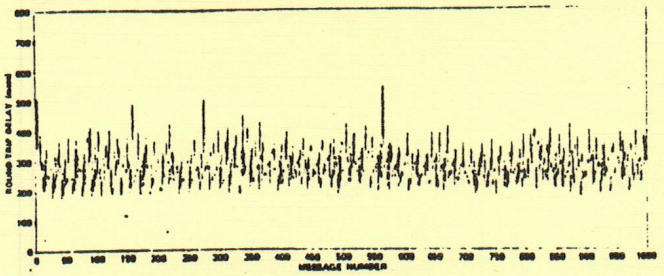

Figure 3. Round-trip delay measurement (5 hops, s=248)

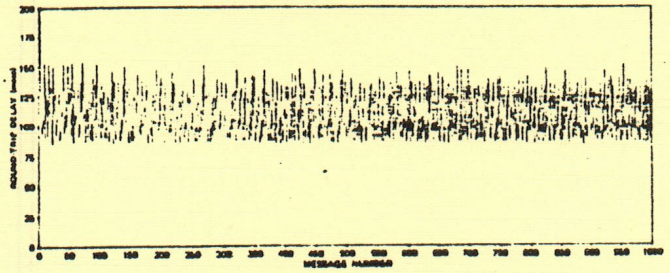

Figure 4. Round-trip delay measurement (2 hops, s=165)

One would not normally expect to see such a regular variation in delay with time assuming no control on the other network traffic. Rather one might expect to see a random function of time, possibly with a slowly varying average which changes with network load [5]. (Indeed, the latter was our hope; we were looking for delay predictors.) Below we present an analysis of the regularity of these delay functions.

Analysis

Before proceeding with the analysis, we must first describe the important features of the periodic routing update scheme in the ARPANET. There exists a basic routing interval of 640 msec. The beginning time for the basic period is chosen essentially at random for each half-duplex channel in the network. It was noted, by the network builders, that as the network grew in size, routing information was not being propagated in a timely fashion with only one update per basic period [2]. Therefore provision was made to send up to five updates in one basic period. During a basic period the line utilization (including that for updates) is measured to determine the number of updates to be sent during the next basic period. For each additional 20% of line utilization, one of the possible five updates is dropped. So, for example, at 65% line utilization only two updates are sent in the next basic period. It is important to note that the updates are not necessarily evenly spaced within the basic period. Rather this period is divided into five equal segments. Routing updates are sent only at segment boundaries (i.e., every 128 msec). For the 20 to 40% range, for example, updates are sent at 0, 128, 384, an 512 msec into the basic period (rather than 0, 160, 320, 480 msec for evenly spaced updates). Routing update packets are 1160 bits in length, requiring 23.2 msec to transmit on the 50 kbps channels used in the ARPANET. Additionally, approximately 12000 machine cycles (based on a measurement reported in [3]) are required by an IMP (the ARPANET switching computer) to process an incoming routing update (i.e., 11.52 msec for a 516 IMP and 19.2 msec for a 316 IMP).

Our analysis uses a queueing system with some "background" traffic (i.e., routing and ambient data traffic) to which we add a stream of deterministically generated traffic. The inclusion of the ambient data traffic is to model the interference caused by other packet sources in the system. We wish to study the system time of this added stream traffic (i.e., the round-trip delay as shown in Figures 1 through 4). We first examine the waiting time on a single channel. The model for the single channel is pictured in Figure 5. There are three classes of customers arriving to a single queue; routing update packets (R), ambient data packets (A), and stream packets (S). All arrivals are served in first-come-first-served (FCFS) fashion (though in the actual system, routing update packets take precedence over both priority data packets and control packets, all of which take precedence over non-priority data packets).

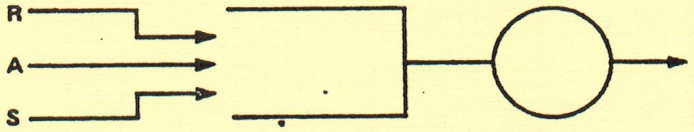

Figure 5. Single channel model

Let w(t) represent the waiting time for a stream traffic packet arriving at time t. Suppose U(t) is the unfinished work, at time t, in the background system (i.e., the system with the stream traffic removed). As long as the stream arrivals have no priority over other customers, an arrival must wait until all work in the system has completed before receiving any service of its own. The amount of work in the system found by an arrival at time t is at least U(t). Hence

$$w(t) \geq U(t)$$

Equality is achieved for an arrival when it is the first stream traffic arrival in a busy period.

For simplicity of the following analysis, we assume that the ambient data traffic has zero intensity. U(t) for a system void of any data traffic is shown in Figure 6. At the arrival time of a routing update packet, the amount of work in the system jumps up by 23.2 msec (the service time of a routing packet). With no other packets in the system, the routing packet is immediately served at a rate of one second per second, and exits the channel after 23.2 msec. After the departure there is no work in the system until the next arrival.

A more interesting measure of performance is w(n*s), the waiting time for the nth message, where the constant interarrival time of

the stream traffic is s. Figure 7 is a plot of U(n*s) for s = 248, 165, and 124 msec (i.e., the periods used in the measurement experiments). For these s (and the message size used in the measurement experiments) the line utilization is in the 20 to 40% range so that every fifth update is dropped. These single channel curves nicely display some of the characteristics of delay shown in Figures 1 through 4. The major shape of Figure 1 is nearly the same as U(n*124). U(n*165) and U(n*248) display the relative variation of the previous corresponding figures. That is U(n*165) varies more rapidly than does U(n*248) which in turn varies more rapidly than U(n*124); and correspondingly Figure 4 varies more rapidly than do Figures 2 and 3 which in turn vary more rapidly than Figure 1. One can clearly identify the period of each U(n*s) curve. Indeed, let P(s) represent the period of U(n*s). Then

$$P(s) = LCM(s,r)/s$$

where LCM(x,y) = the least common multiple of x and y, and r = the routing period (i.e., the minimum time r such that the pattern of routing arrivals is the same in the intervals (0,r), (r,2*r), (2*r,3*r), ...). For these examples, r=640 msec. P(s) has the values 80, 128, 160 for s = 248, 165, and 124 respectively.

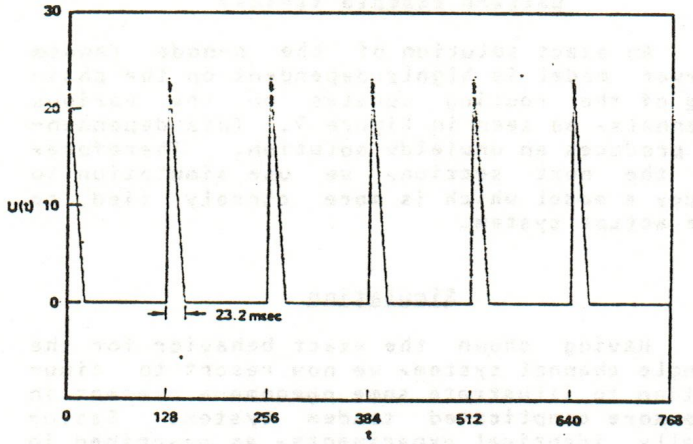

Figure 6. Unfinished work (routing only)

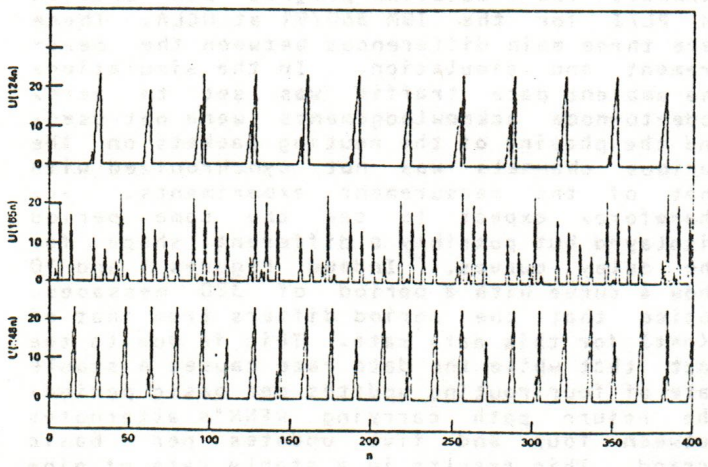

Figure 7. Unfinished work at stream arrivals

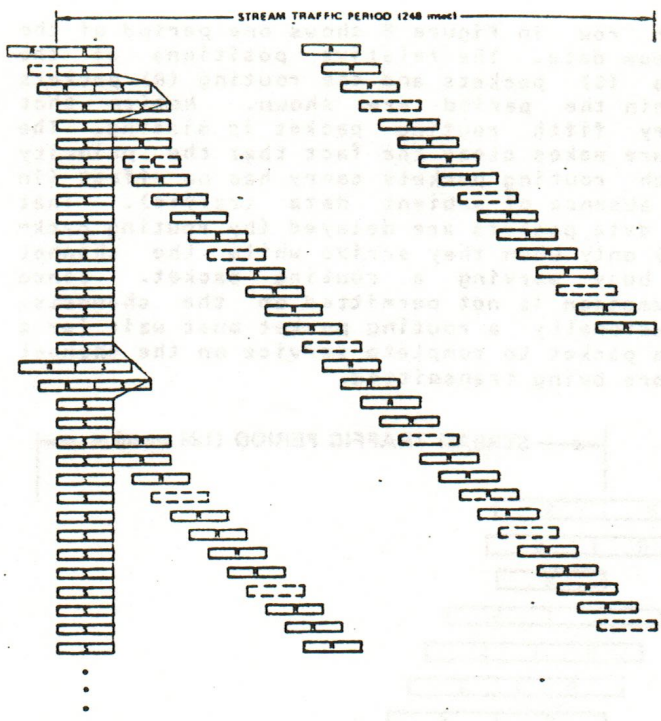

Figure 8a. Precession of interference pattern (s=248)

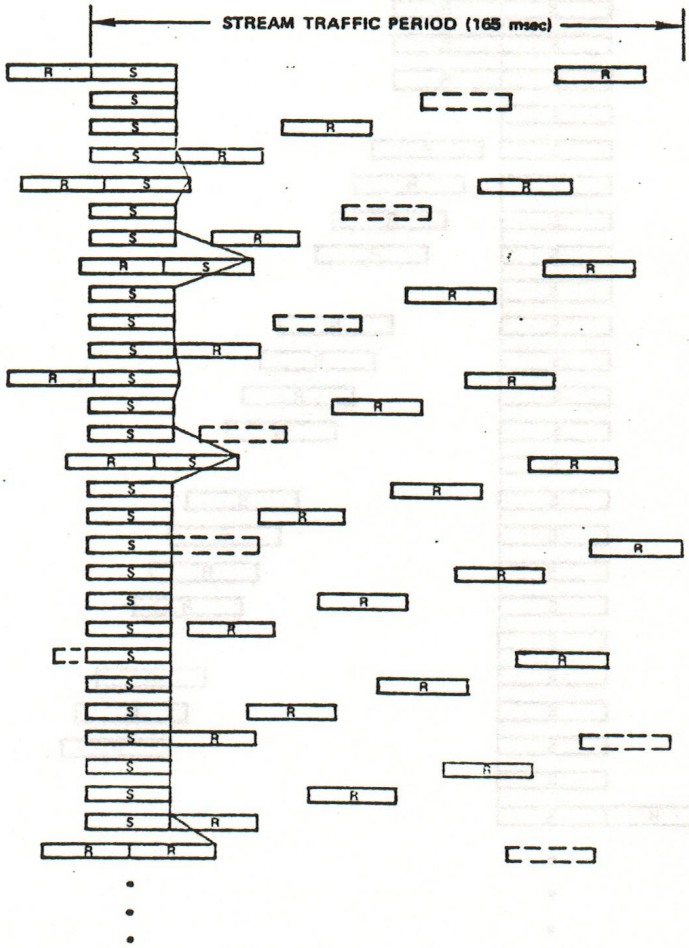

Figure 8b. Precession of interference pattern (s=165)

Figure 8 shows the detailed precession of the periods of the routing and stream traffic.

Each row in Figure 8 shows one period of the stream data. The relative positions of the data (S) packets and the routing (R) packets within the period are shown. Notice that every fifth routing packet is missing. The figure makes clear the fact that the priority which routing packets carry has no effect (in the absence of ambient data traffic). That is, data packets are delayed (by routing packets) only when they arrive while the channel is busy serving a routing packet. Since preemption is not permitted on the channels, occasionally a routing packet must wait for a data packet to complete service on the channel before being transmitted.

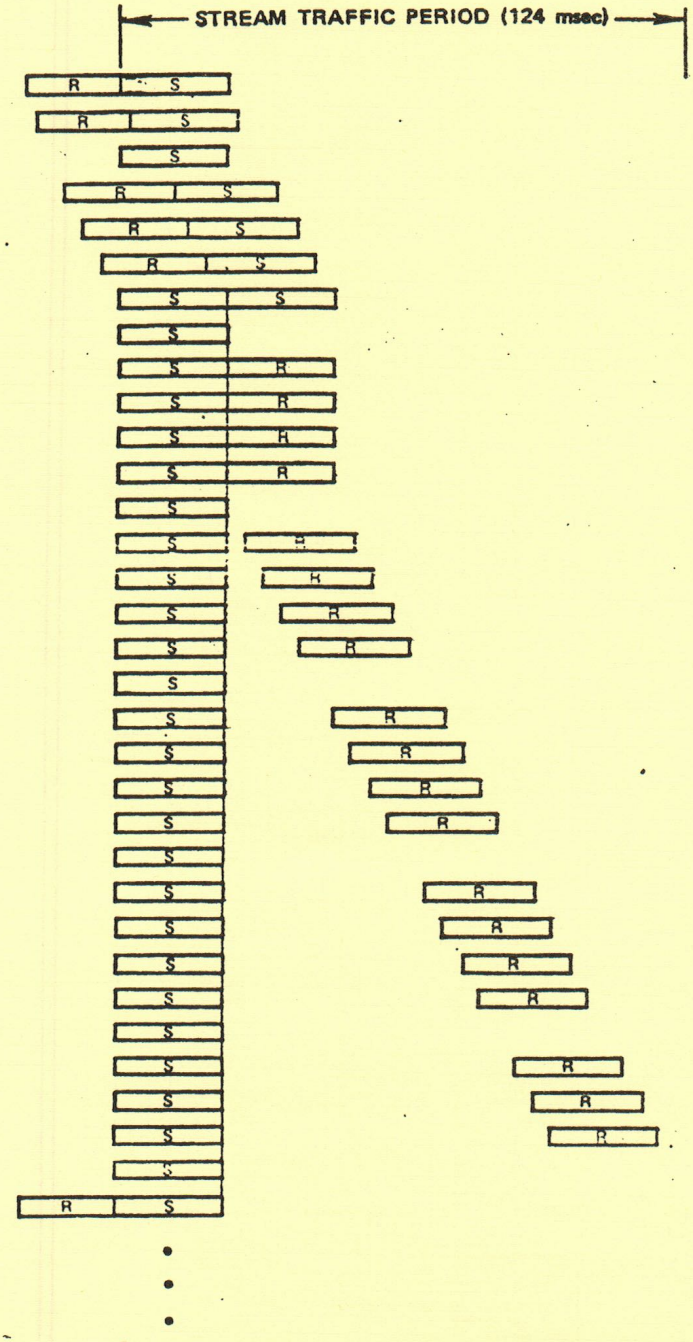

Figure 8c. Precession of interference pattern (s=124)

When packets pass through more than one tandem channel, a more complicated pattern arises. An example pattern for a two channel system is shown in Figure 9. Once the packets are delayed at channel 1, their arrival rate at channel 2 then corresponds to that of the routing packets. During this time we have two deterministic streams arriving at a constant offset. Hence the waiting time at channel 2 remains constant (except for routing drop outs) until the data packets are no longer delayed at channel 1. This example illustrates the climbing-dropping phenomenon shown in Figures 2 and 3.

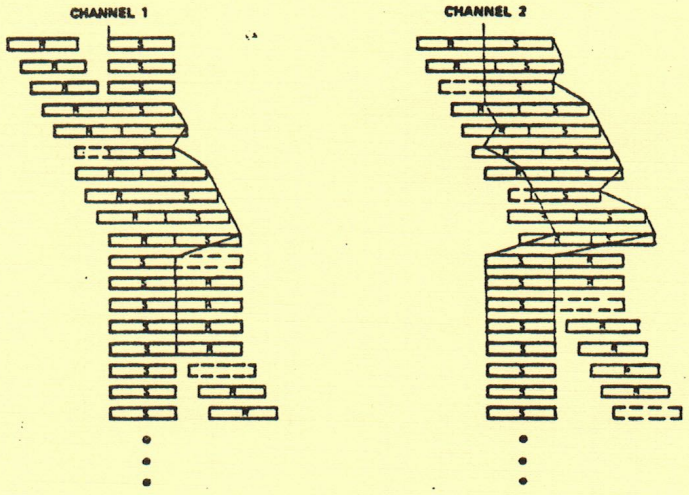

Figure 9. Two tandem channel interference pattern example (s=124)

An exact solution of the n-node tandem server model is highly dependent on the phasing of the routing updates on the various channels, as seen in Figure 9. This dependency produces an unwieldy solution. Therefore, in the next section, we use simulation to study a model which is more closely tied to the actual system.

Simulation

Having shown the exact behavior for the single channel system, we now resort to simulation to illustrate some phenomena present in the more complicated tandem system. Essentially identical experiments, as described in the above measurement section, were performed using a rather detailed simulation of the ARPANET. The simulation program was written in PL/I for the IBM 360/91 at UCLA. There were three main differences between the measurement and simulation. In the simulation, the ambient data traffic was set to zero, node-to-node acknowledgements were not used, and the phasing of the routing packets on the various channels was not synchronized with that of the measurement experiments. We, therefore, expect to see the same period displayed but possibly a different shape for the delay curves. Indeed Figures 1 and 10 show a curve with a period of 320 messages. Notice that the period differs from that of U(n*t) for this data rate. This is due to the fact that while the data rate causes a stable rate of four routing updates per basic period, the return path carrying RFNM's alternates between four and five updates per basic period. This results in a stable rate of nine updates in each two basic periods. That is, 1280 is the minimum time r for which the

pattern of routing arrivals is fixed in the intervals (0,r), (r,2*r), (2*r,3*r),... . Therefore, the round-trip delay curve has a period of LCM(124,1280)/124 = 320 messages. Figure 1 may exhibit a period of 160 messages due to the node-to-node acknowledgements which elevate the traffic on the backward channel just enough to force a constant rate of four updates per basic period. One also notices that the minimum values for Figures 1 and 10 are not the same. This is due to several factors. In the simulation we have estimated the channel propagation time and the nodal processing time; both are likely lower than their actual value. Another factor is that we have assumed zero acceptance time for the message at the destination node and for the RFNM at the source node. These assumptions drive the delay down for the simulation and hence the vertical offset.

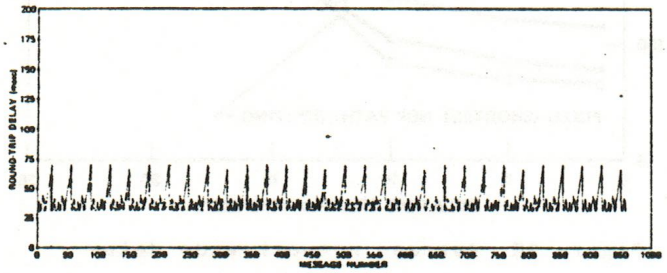

Figure 10. Round-trip delay simulation (1 hop, s=124)

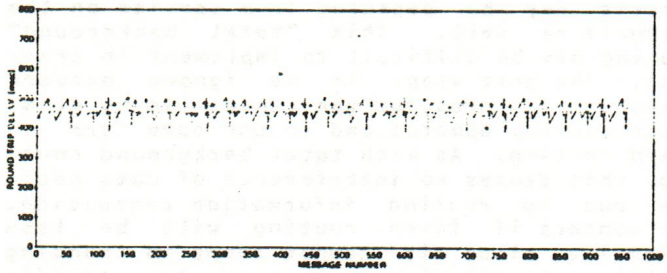

Figure 11. Round-trip delay simulation (10 hops, s=248)

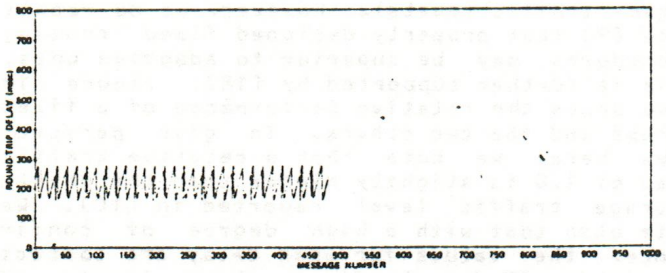

Figure 12. Round-trip delay simulation (5 hops, s=248)

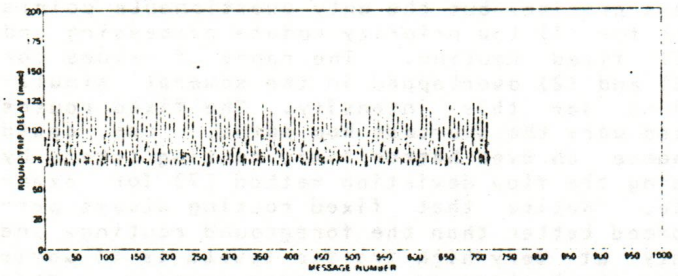

Figure 13. Round-trip delay simulation (2 hops, s=165)

Figures 2 and 3 are too random to identify a period. However, one should notice the slow climbing and rapid falling in the curves in Figures 2, 3, 11 and 12 (though the measured data is quite noisy). The most rapid variation, both for simulation and measurement, is the packet rate of approximately six per second shown in Figures 13 and 4 respectively. Both curves possess a period of 128 messages.

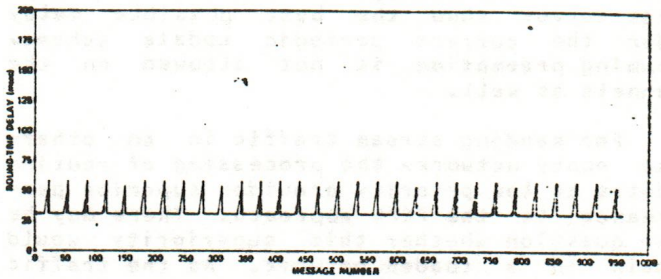

Figure 14. Round-trip delay simulation with low priority routing processing (1 hop, s=124)

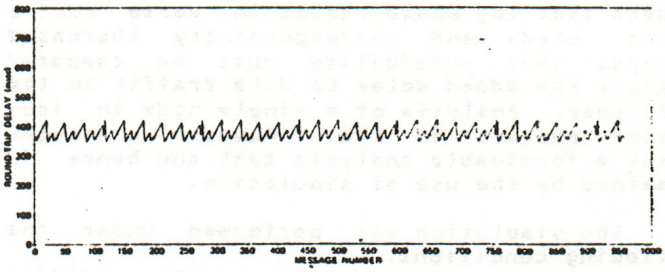

Figure 15. Round-trip delay simulation with low priority routing processing (10 hops, s=248)

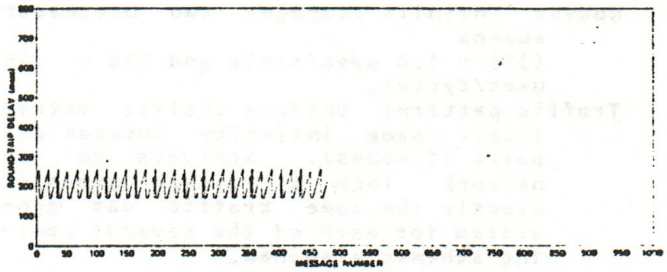

Figure 16. Round-trip delay simulation with low priority routing processing (5 hops, s=248)

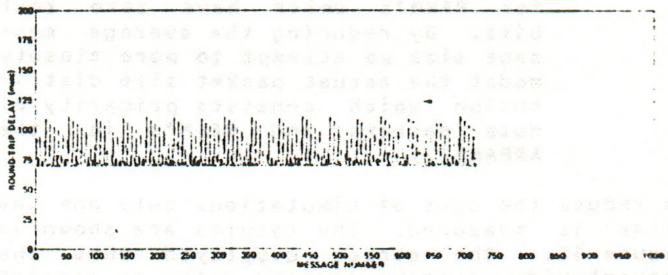

Figure 17. Round-trip delay simulation with low priority routing processing (2 hops, s=165)

Nestled between the large delays in Figures 1 and 10 are some small mounds. These are due to the interference between routing updates and stream traffic in the "TASK" queue [14] (i.e., data packets waiting to be placed on an output queue and routing packets waiting

and being digested into the local routing table). The TASK queue is currently served in FCFS fashion. A conceptually simple modification is to serve routing packets at low priority from the TASK queue. Figures 14 through 17 show the effect of sending routing updates at the same rate as before, but processing data packets by preempting the processing of interfering routing updates. Notice the decrease in both average and variance of delay. These curves show the best possible delay under the current periodic update scheme, assuming preemption is not allowed on the channels as well.

For sending stream traffic in an otherwise empty network, the processing of routing updates at low priority provides superior performance to the FCFS approach. There may be some question whether this superiority would remain in a loaded network. As the traffic increases, the routing updates remain on the TASK queue for a longer time thus causing a possible lag in the propagation of recent routing information. Eventually, one might expect that lag would result in worse routes being used, and correspondingly increased delays. This possibility must be compared against the added delay to data traffic in the FCFS case. Analysis of a single node in isolation appears in [16]. However, a network poses a formidable analysis task and hence is examined by the use of simulation.

The simulation was performed under the following conditions:

- Topology: ARPANET (June 1975) modified to exclude satellite links.
- Lines: ARPANET capacities (mostly 50 kb/s, some 230.4 kb/s).
- Nodes: Infinite storage, two processor speeds (316 - 1.6 usec/cycle and 516 - .96 usec/cycle).
- Traffic pattern: Uniform traffic matrix (i.e., same intensity between all pairs of nodes). Arrivals to the network form a Poisson process. Exactly the same traffic was generated for each of the several routing schemes examined.
- Message length distrbution: Exponential truncated at 8064 bits, with a mean of 122 bits. (The mean is one half of that reported in [10] to allow for RFNM's which have zero text bits. By reducing the average message size we attempt to more closely model the actual packet size distribution which consists primarily of data packets and RFNM's in the ARPANET.)

To reduce the cost of simulation, only one way delay is measured. The results are shown in Figure 18. The curves displayed show the network-wide average message delay as network load increases. We see that the routing information lag caused by the preemption of the processing of routing updates never causes worse delay than that caused by the FCFS scheme. The expense of performing routing processing in the foreground is higher than the gain (i.e., more current routing information and possibly better routes) for this traffic mix. If one must propagate routing information in this way then clearly a better approach is to process routing updates at low priority.

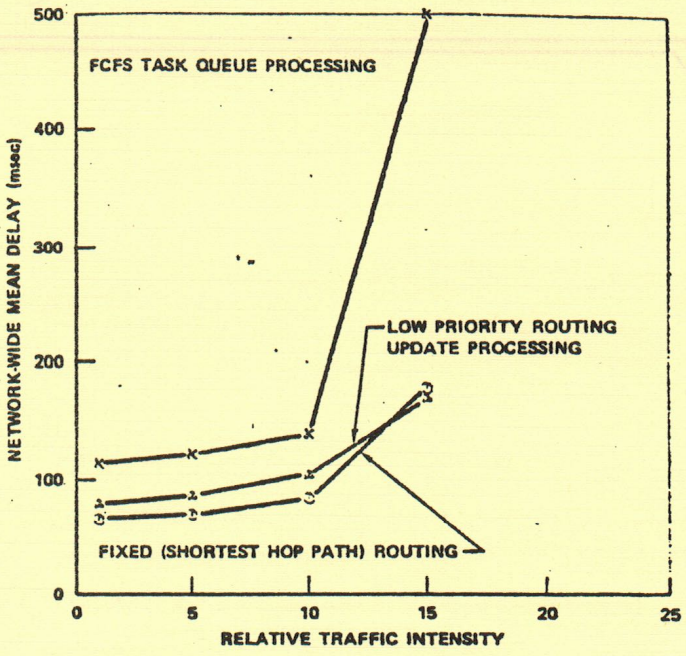

Figure 18. Average network-wide delay

One may go one step further and consider the performance of a system where routing packets may be peempted from service on the channels as well. This "total background" routing may be difficult to implement in practice. The next step, if we ignore network componert failures for the moment, is to eliminate routing updates and to use some form of fixed routing. As with total background routing, this causes no interference of data packets due to routing information processing. One woncers if fixed routing will be less effective since it cannot adapt to changing traffic patterns (i.e., at very low traffic levels fixed routing performs better than adaptive schemes, but perhaps may fail at higher traffic levels). However, we do recall from [9] that properly designed fixed routing procedures may be superior to adaptive ones. This is further supported by [18]. Figure 18 also shows the relative performance of a fixed scheme and the two others. To give perspective here, we note that a relative traffic load of 1.0 is slightly higher than the weekly average traffic level reported in [1C]. We note also that with a high degree of confidence, the values for mean delay are correct to within ±7% for the first three levels of relative traffic intensity (i.e., 1.0, 5.0, and 10.0). The values at intensity 15.0 are less precise, but the only questionable points are for (1) low priority update processing and (2) fixed routing. The range of values for (1) and (2) overlapped in the several simulations for this intensity. The fixed routes used were the shortest hop paths. One could choose an even better fixed routing scheme by using the flow deviation method [7] for example. Notice that fixed routing always performed better than the foreground routing, and only at very high traffic levels is it worse than the backgroud processing routing. This suggests that the cost of routing in the ARPANET is extremely high indeed, since traffic levels are currently very low.

We have so far ignored network failures. It is clear that failures do occur in practice. Long term monitoring of the ARPANET [13] show a mean time between failures (MTBF) of 431 hours for lines and 221 hours for nodes. Failures cause topological changes to occur in the network in the following two ways. When a channel fails, it is as if it were removed from the network. When a node fails, all its attached channels are removed from the network. We define the network-wide MTBF to be the mean time between channel removals. Then with the 57 nodes and 65 full duplex channels, in the June 1975 ARPANET, and assuming that each node is of average connectivity (i.e., approximately 2.24 [3]), these figures yield a network-wide MTBF of 3.76 (=1/(65/431+57/221/2.24)) hours.

A far better method of routing, it appears, would be to use a passive scheme. In such a scheme one establishes routes and continues to use them in a fixed routing fashion until some catastrophe occurs (i.e., a failure or possibly even severe congestion). At the time a catastrophe occurs one could "turn on" routing updates until the tables "stabilize". From the above data one can see that the average trigger time for routing due to failure would be almost two hours (or approximately 10000 times the basic routing period)!

Conclusions

In this paper we have shown some interesting message delay phenomena attributable to the periodic routing scheme used in the ARPANET. We conclude that periodic routing is quite costly in medium sized (and bigger) networks. In order to assure good performance in the face of failure (or heavy congestion) one pays a high price in terms of message delay for periodic routing update procedures in networks of the size of the ARPANET. We suggest the use of a passive routing scheme, in which updates are scheduled (only) as the result of failure (or heavy congestion). The results presented here suggest that this method could provide superior performance at reduced cost.

References

[1] Specifications for the Interconnection of a HOST and an IMP, Report No. 1822, Bolt Beranek and Newman, Cambridge, Mass., May 1969.

[2] Interface Message Processors for the ARPA Computer Network, Quarterly Technical Report No. 4, Report No. 2717, Bolt Beranek and Newman Inc., Cambridge, Mass., January 1974.

[3] Interface Message Processors for the ARPA Computer Network, Quarterly Technical Report No. 2, Report No. 3106, Bolt Beranek and Newman Inc., Cambridge, Mass., July 1975.

[4] Cegrell, T., "A Routing Procedure for the TIDAS Message-Switching Network," IEEE Transactions on Communications, Vol. COM-23, No. 6, June 1975, pp. 575-585.

[5] Cohen, D., Personal communication, 1974-76.

[6] Fultz, G. L., Adaptive Routing Techniques for Message Switching Computer Communication Networks, Engineering Report No. UCLA-ENG-7252, University of California, Los Angeles, 1972.

[7] Gerla, M., The Design of Store-and-Forward (S/F) Networks for Computer Communication, Engineering Report No. UCLA-ENG-7319, University of California, Los Angeles, 1973.

[8] Forgie, J. W., "Speech Transmission in Packet-Switched Store-and-Forward Networks," AFIPS Conference Proceedings, Vol. 44, NCC, 1975, pp. 137-142.

[9] Kleinrock, L., Communication Nets: Stochastic Message Flow and Delay, McGraw Hill, N. Y., 1964, reprinted by Dover, N. Y., 1972.

[10] Kleinrock, L., and W. E. Naylor, "On Measured Behavior of the ARPA Network," AFIPS Conference Proceedings, Vol. 43, NCC, 1974, pp. 767-780.

[11] Kleinrock, L., and H. Opderbeck, "Throughput in the ARPANET - Protocols and Measurement," Proceedings of the Fourth Data Communications Symposium, Quebec City, Canada, October 1975, pp. 6.1-6.11.

[12] McCoy, C., Jr., Improvements in Routing for Packet-Switched Networks, Ph. D. Dissertation, School of Engineering and Applied Science, George Washington University, Washington, DC, 1975.

[13] McKenzie, A. A., Letter to S. D. Crocker, 16 January 1974.

[14] McQuillan, J. M., W. R. Crowther, B. P. Cosell, D. C. Walden, and F. E. Heart, "Improvements in the Design and Performance of the ARPA Network," AFIPS Conference Proceedings, Vol. 41, pp. 741-754.

[15] McQuillan, J. M., Adaptive Routing Algorithms for Distributed Computer Networks, Report No. 2831, Bolt, Beranek, and Newman, Inc., Cambridge, Mass., 1974.

[16] Naylor, W. E., Real-Time Communication in Packet Switched Networks, to appear, University of California, Los Angeles, 1976.

[17] Pickholtz, R. L., and C. McCoy, Jr., "Effects of a Priority Discipline in Routing of Packet-Switched Networks," IEEE Transactions on Communications, Vol. COM-24, No. 5, May 1976, pp. 506-516.

[18] Price, W. L. Survey of NPL Simulation Studies, 1968-72, NPL Report CLM 60, National Physical Laboratory, Teddington, UK, November 1972.