2.2 Network Performance

We now turn from consideration of the specific implementation of the ARPA Network to a more general discussion of computer networks and the evaluation of their performance. This analysis of networks leads to a more specific examinaton of the same considerations from the point of view of routing algorithms, a subject taken up in section 2.3 following this section.

Any communications system has two fundamental goals in the processing of data - low delay and high throughput. Each message should be handled with a minimum of waiting time, and the total flow of data per second should be as large as possible. To these two goals we may add two equally important goals, which apply to message processing and to the operation of the network as a whole. The network should be cost-effective. Individual message service should have a reasonable cost in terms of utilization of network resources. Further, the network facilities, primarily the node computers and the circuits, should be utilized to the extent that their cost is also reasonable. Finally, the network should be reliable. Messages accepted by the network should be delivered to the destination with a high probability of success. And the network as a whole should be a hardened computer communications service, fault-tolerant, and able to function in the face of node or circuit failures. We can examine these issues in more detail, and point out the natural tradeoffs.

2.2.1 Delay

In this section we consider what delay performance characteristics are possible in a given network. The topics discussed below include the identification of the components of delay, an analysis of the minimum delay possible for a round trip through a network, the reason for breaking messages up into packets, a brief look at queueing delay, and a discussion of delays for interactive traffic. Some of this material was presented in (MCQU 72), and some of the best other references to this subject matter are (KLEI 64, KLEI 71b, METC 73b). We will return to a discussion of network delay in section 2.3 on the effects of routing on delays.

Components of Delay. We will first discuss the delay experienced by a single packet transmitted over a single hop. The components of delay which we will take to be fundamental variables are as follows:

1. L = propagation delay or speed of light latency.

This is the delay for the first bit of the packet to traverse the circuit.

L = (circuit length [mi.])/(signal propagation rate [mi./sec]).

2. T = transmission delay.

This is the time for the bits of a packet to be clocked out on the circuit.

T = (number of bits in packet [bits])/(transmission rate [bits/sec]).

In the analysis below we will use Tp to denote T for a packet and Tr to denote T for a RFNM, which we will assume is a minimum length packet.

3. C = nodal processing delay.

This is the time it takes the node to process the packet. It has two fixed components, corresponding to the store operation and the forward operation, or receive and transmit, plus a random component due to queued tasks of higher priority. This component measures the interference experienced by packets queued for processor service.

C = Cr[sec] + Ct[sec] + Cq[sec].

receive transmit queueing - random

4. Dq = queueing delay

This is the time that the packet must wait for the transmission of the packets which precede it on the output queue, including the output time of the packet currently being transmitted. Thus, Cq measures input queueing delay, waiting for central processor service, while Dq measures output queueing delay, waiting for circuit service.

5. Dr = delays due to retransmissions.

This is the time that the packet must wait in the event that its first transmission is unsuccessful. This may happen if it was in error or if the other node refused it for some reason, or, in the case of broadcast circuits, there was a collision with another packet.

In general, we will assume that the first two components are much greater than the last three. The tables below give some representative values for L and T. (Note that a 50 Kbs circuit can transmit 1000 bits in 20 ms. and that the first bit can travel about 4000 miles in that time. So a bit is about 4 miles long at 50 Kbs!)

Distance	L	Line Type
10 mi.	0.054 ms.	intra-city line
100 mi.	0.54 ms.	inter-city line
1000 mi.	5.4 ms.	long line
3000 mi.	16.2 ms.	cross-country line
10000 mi.	54 ms.	very long terrestial circuit
45000 mi.	272 ms.	satellite link

Table 2-1 Some Representative Propagation Delays

Circuit Bandwidth	Short Packet $Tp = Tr$ 152 bits	Long Packet Tp 1160 bits
9.6 Kbs.	15.7 ms.	120.5 ms.
50 Kbs.	3.04 ms.	23.2 ms.
230.4 Kbs.	0.66 ms.	5.03 ms.
1344 Kbs.	0.106 ms.	0.81 ms.

Table 2-2 Some Representative Transmission Delays

A typical value of C is probably 1 millisecond for an IMP-like node.

Minimum Round Trip Delay. Now we can examine the minimum round trip delay, by taking the case of Cq = Dq = Dr = 0. Consider a message with P packets traveling over a path of H hops. If the delay at hop i is

D(i) = L(i)+T(i)+C(i)

we can define the natural quantity D(ave), average hop delay, as follows:

D(ave) = (1/H)*Sum[i=1,H](L(i)+T(i)+C(i)).

We can also define the less obvious variable D(max):

D(max) = Max[i=1,H](T(i)+Ct(i)).

With these definitions, we can make the following two statements:

1. The first packet experiences delay equal to H*D(ave).
2. The remaining P-1 packets follow through the network, each packet at most one hop behind the preceding packet, and these packets add (P-1)*D(max) to the total delay.

This analysis can be illustrated by the following numerical example:

P = 3

H = 3

Ct = Cr = 0.5

	HOP		
	1	2	3
L	0	1	3
T	2	3	2

Total delay = 21, as shown in Figure 2-2a:

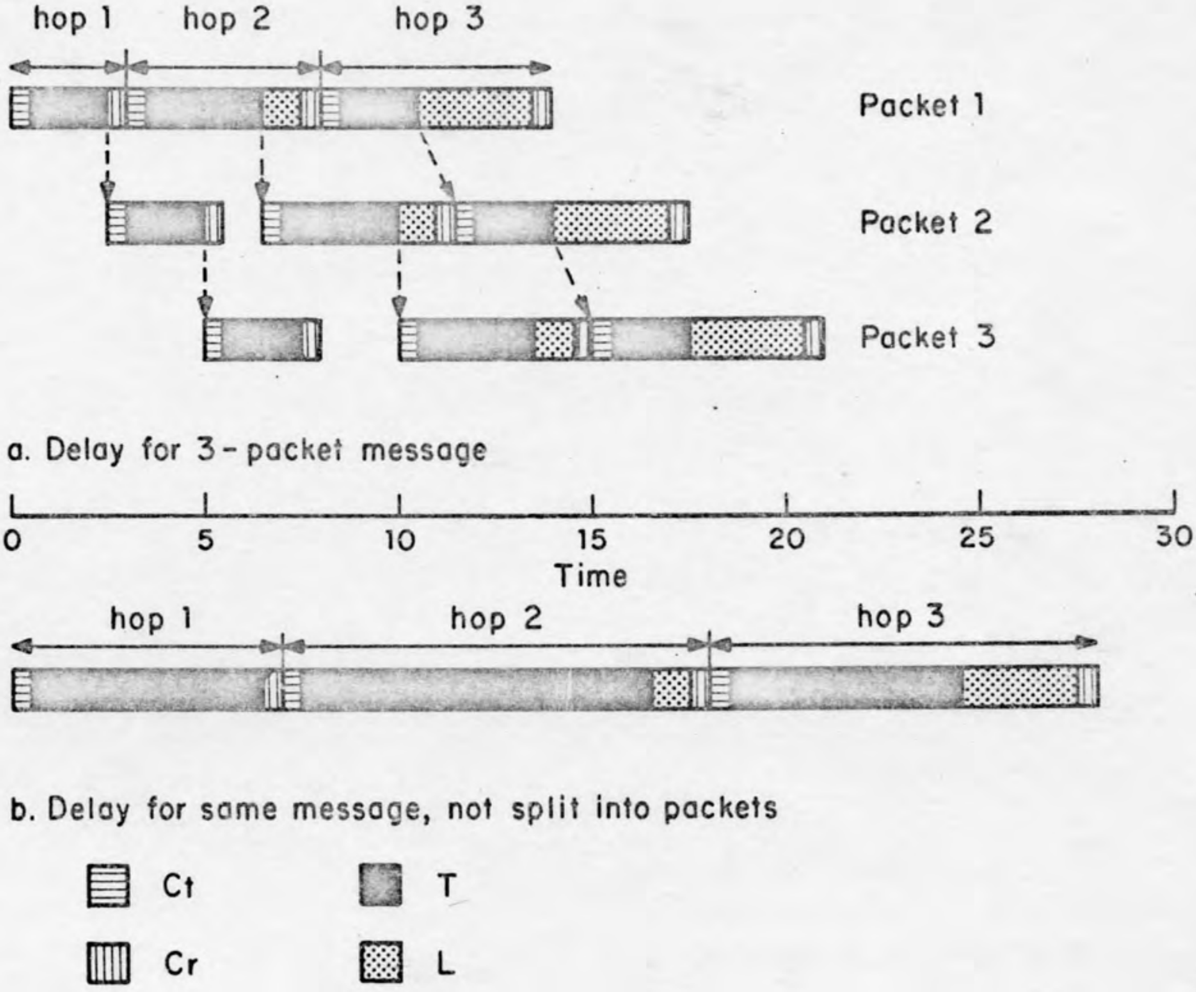

Figure 2-2 An Example of Delays for a 3-Packet Message

Note that the bottleneck hop, in this case hop 2, has the largest T(i) + Ct(i), here equal to 3.5, but not the highest total delay. That is, hop 3 has total delay of 6, compared to the delay of 5 over hop 2. Note also that more than one packet is being transmitted at the same time, giving a pipelined effect, and reducing the total delay for the message.

This means that the delay for a single packet message is

$$D(SP) = Sum[i=1,H](L(i)+Tp(i)+C(i)).$$

and the delay for a multipacket message is

$$D(MP) = (P-1)*Max[i=1,H](T(i)+Ct(i)) + D(SP).$$

The delay for a RFNM is

$$D(RFNM) = Sum[i=1,H](L(i)+Tr(i)+C(i)).$$

Therefore, the minimum round trip delay for a message of P packets over H hops is

$$D(MRT) = Sum[i=1,H](2*L(i)+2*C(i)+Tp(i)+Tr(i)) + (P-1)*Max[i=1,H](Tp(i)+Ct(i)). \quad (2.2.1\text{-}1)$$

If the values of L, C, Ct, and T are the same for each hop, then we have a simplified minimum round trip delay

$$D(MRTS) = \underbrace{H*(L+Tp+C)}_{\text{first packet}} + \underbrace{(P-1)*(Tp+Ct)}_{\text{subsequent packets}} + \underbrace{H*(L+Tr+C)}_{\text{RFNM}}. \quad (2.2.1\text{-}2)$$

Some curves are given on the next page which illustrate the minimum round trip delay through a network for a range of message lengths and path lengths, for two sets of line speeds and lengths over paths of 1 to 6 hops. These graphs are from (MCQU 72).

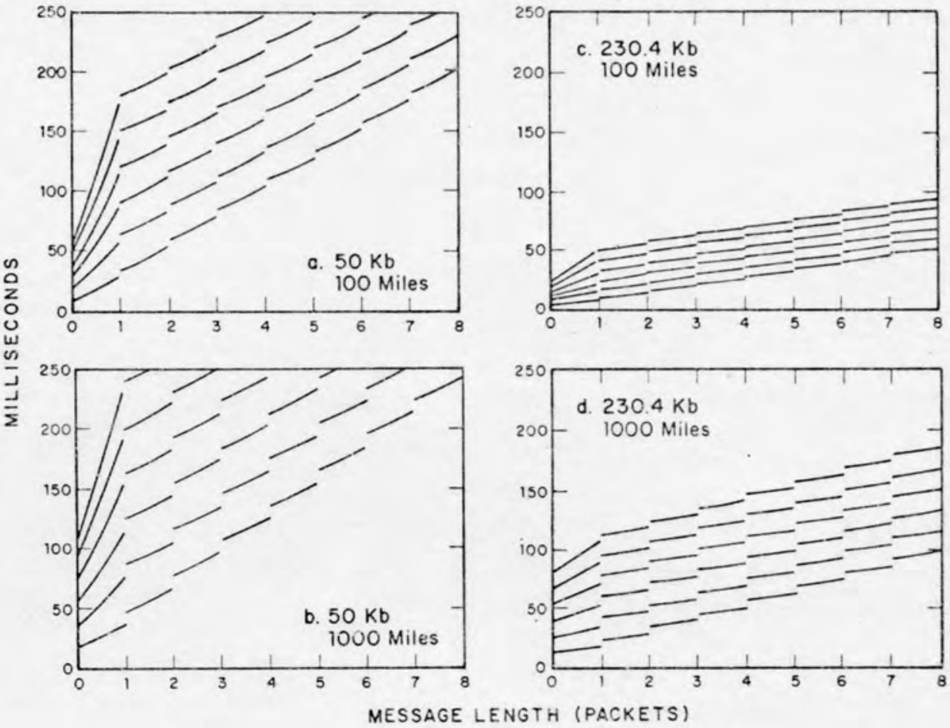

Figure 2-3 Minimum Round Trip Delay vs. Message Length

<u>The Rationale for Packetizing Messages.</u> As a slight digression, we now consider the rationale for breaking up messages into packets in order to reduce delay. It will be shown that this rationale is similar to that for any pipelining technique, and the variables of interest will be identified. We will carry through this discussion for the simplified case of identical values of L, C, Ct, and T at each hop. First we rewrite equation 2 as

$$D(MRTS) = (P-1)*(Tp+Ct) + H*(2*(L+C)+Tp+Tr)$$

(2.2.1-3)

Note that the only one of the variables L, C, Ct, and T which is a function of packet length is T. Let us suppose that instead of a message of P packets, each of which has a transmission delay of T, the message is sent as a whole, with a transmission delay of P*T. The total delay can be calculated as above, considering this new entity as a long single packet message. Using equation 2, we get an equation for minimum round trip delay, simplified , not packetized

D(MRTSNP) = H*(L+P*Tp+C) + H*(L+Tr+C) (2.2.1-4)

We can rewrite this as

D(MRTSNP) = (P-1)*H*Tp + H*(2*(L+C)+Tp+Tr) (2.2.1-5)

Subtracting equation 3 from equation 5 we get a difference in delay

D(MRTDIF) = (P-1)*((H-1)*Tp-Ct) (2.2.1-6)

We make the following observations

1. Clearly, if P=1, D(MRTDIF) = 0

There is no difference between the two techniques for single-packet messages.

2. Assuming P >1, if H=1, D(MRTDIF) = -(P-1)*Ct < 0

That is, if H=1 and P >1, it involves <u>more</u> delay to break a message into packets because of the added processor overhead (generally small).

3. Assuming Tp >Ct, that the processor time is small, if P >1 and H> 1, then D(MRTDIF) > 0.

This is the usual case, and constitutes the basic reason to break messages into packets.

We can return to the numerical example used earlier with the equivalent conditions for a single packet message as long as the 3 packet message considered above:

P = 1

H = 3

Ct = Cr = 0.5

	HOP		
	1	2	3
L	0	1	3
T	6	9	6

Total delay = 28, as opposed to 21 for the packetized case (see Figure 2-2b).

Queueing Delay. We have examined the minimum round trip delay as a function of message length, network path length, and the packetizing strategy. It is appropriate at this point to analyze the effects of additional delays which may be present. To perform the minimum delay analysis, we made 3 assumptions, each of which should be re-examined at this point:

1. Cq = 0. The packet arriving may have to wait on an input queue before it is serviced by the processor. This time is generally quite small, but it is a random variable which may take on large values if there are time-consuming high-priority tasks in the system.

2. Dq = 0. This is the major assumption which must be modified in the analysis of actual delays in a network. A

great deal of theoretical work has been done in studying queueing delay, particularly by Kleinrock (KLEI 64, KLEI 70), who has used both analysis and simulation in this regard. For the simplified discussion at hand, it is sufficient to note that each packet on the queue adds an additional T+Ct to the delay of all packets behind it on the queue.

3. Dr = 0. The accuracy of this last assumption varies widely with the type of network and line being considered. For some wideband circuits, particularly satellite channels, the error rates are very low and few retransmissions may be necessary for reasons of errors. However, satellite links may be used in a broadcast competition mode so that some packets are lost in collision with others. Finally, there is always some chance that the adjacent node will refuse to accept a packet for lack of processing resources.

Low Delay for Interactive Traffic. Perhaps the most important consideration about delay in a network is this: some traffic consists of interactive, high-priority messages and this traffic must be delivered to its destination as rapidly as possible. This is in contrast to bulk transfer traffic which is not so delay-critical. The most obvious case of such interactive traffic is most man-computer dialogue, which consists of rather short messages between the computer and a man at a terminal. Here there is a definite threshold for delay. Below this threshold delay is acceptable, and above it delay is unacceptable. There is no added benefit if the delay is considerably below the threshold, and it is likely that once the delay is much above the threshold, almost any value of delay is equally unacceptable.

Another way of looking at the bimodal nature of network traffic is to consider that most of the delay for an interactive message is in the network itself. That is, it is generated and quickly sent into the network. When it is delivered, it will be processed quickly at the destination Host. Bulk transfers on the other hand may experience lengthy delays outside the network due to buffering considerations and the very size of the data (secondary storage or tapes or cards may be involved in the data transfer, greatly increasing delay).

2.2.2 Throughput

In this section we consider what throughput performance characteristics are possible in a given network. Some of this material was presented in (MCQU 72) and (MCQU 73). The topics discussed below include an analysis of nodal processor bandwidth, an examination of circuit overhead, a quantification of the buffering in packets required for a network line and the buffering in messages required for a network path, the throughput requirements of bulk transfer traffic, and the tradeoff between delay and throughput. The subject of throughput will come up again in section 2.3 on routing where we consider the relationship between routing algorithms and throughput in networks.

The Effective Bandwidth of the Node Processor. We will first examine the effective processing capablity of the node computer in a network with variable message length. Some numerical examples will be given to support the intuitive notion that the processor is most effective for long messages, given some very general assumptions about the packet processing involved.

We begin by defining some new quantities of interest in studying throughput in networks. The quantities which we will take as fundamental variables are as follows:

1. Bd = the number of Host data bits in a packet.
2. Bs = the number of software overhead bits per packet. These bits include header information such as address and identifying information such as message number.
3. Bh = the number of hardware overhead bits per packet. These are typically framing bits for the circuit, and error detection bits such as checksums or redundant information.
4. P = the number of packets per message, as above.
5. Btot = the total number of Host data bits per message.

$$Btot = P*Bd - \text{unused bits in the last packet.}$$

Now we will examine how long it takes the node processor to store and forward a message. We ignore the processing at the source node and at the destination node on the assumption that it is not very much greater than the store-and-forward processing time, and that there are probably many intermediate IMPs between the source and the destination. Thus, the store-and-forward processing bandwidth of the node is the critical parameter. The time to process a store-and-forward message is the sum of:

6. C = the packet processing time, as above.

We assume that C is independent of packet length or type. To the extent that this is not true, the length-dependent component of C can be accounted for in item 8 below.

7. BWPo = the fraction of the bandwidth of the processor taken by overhead.

Due to certain necessary periodic processes within the node, notably the routing computation, effective processor bandwidth is reduced.

8. BWIO = the I/O rate of the node in bits/sec.

We assume here that BWIO is a linear function of the number of bits in the message. In most I/O architectures, it is probably a function of the number of computer words in the message, which is identical apart from unused bits in the last word. We are also assuming that the I/O transfer steals cycles from the processor, reducing its effective bandwidth.

9. I = the I/O transfer time in seconds. We will denote the I/O time for a packet, a message, and a RFNM as Ip, Im, and Ir respectively:

$$Ip = 2*(Bd+Bs)/BWIO$$

$$Im = 2*(Btot+P*Bs)/BWIO$$

$$Ir = 2*Bs/BWIO$$

10. MT = the total time taken to process a message.

$$MT = C*(P+1)*(1+BWPo) + Im + Ir.$$

11. BWPd = the maximum data bandwidth that the node can support.

$$BWPd = Btot/MT.$$

This the the number of Host data bits per second that the node can process.

12. BWPl = the maximum line bandwidth that the node can support.

$$BWPl = (Btot +(P+1)*(Bs+Bh))/MT$$

This represents a processing capability limit on the number and speed of the circuits that can be connected to a node. The difference between the two quantities, BWPl-BWPd, is a measure of the line overhead at a given message length.

At this point, a numerical example may be illustrative. As presented in (MCQU 72), in the ARPA Network, the processor bandwidth of the IMP, both BWPd and BWPl, can be plotted as functions of the message length, and some results are given in the graphs below.

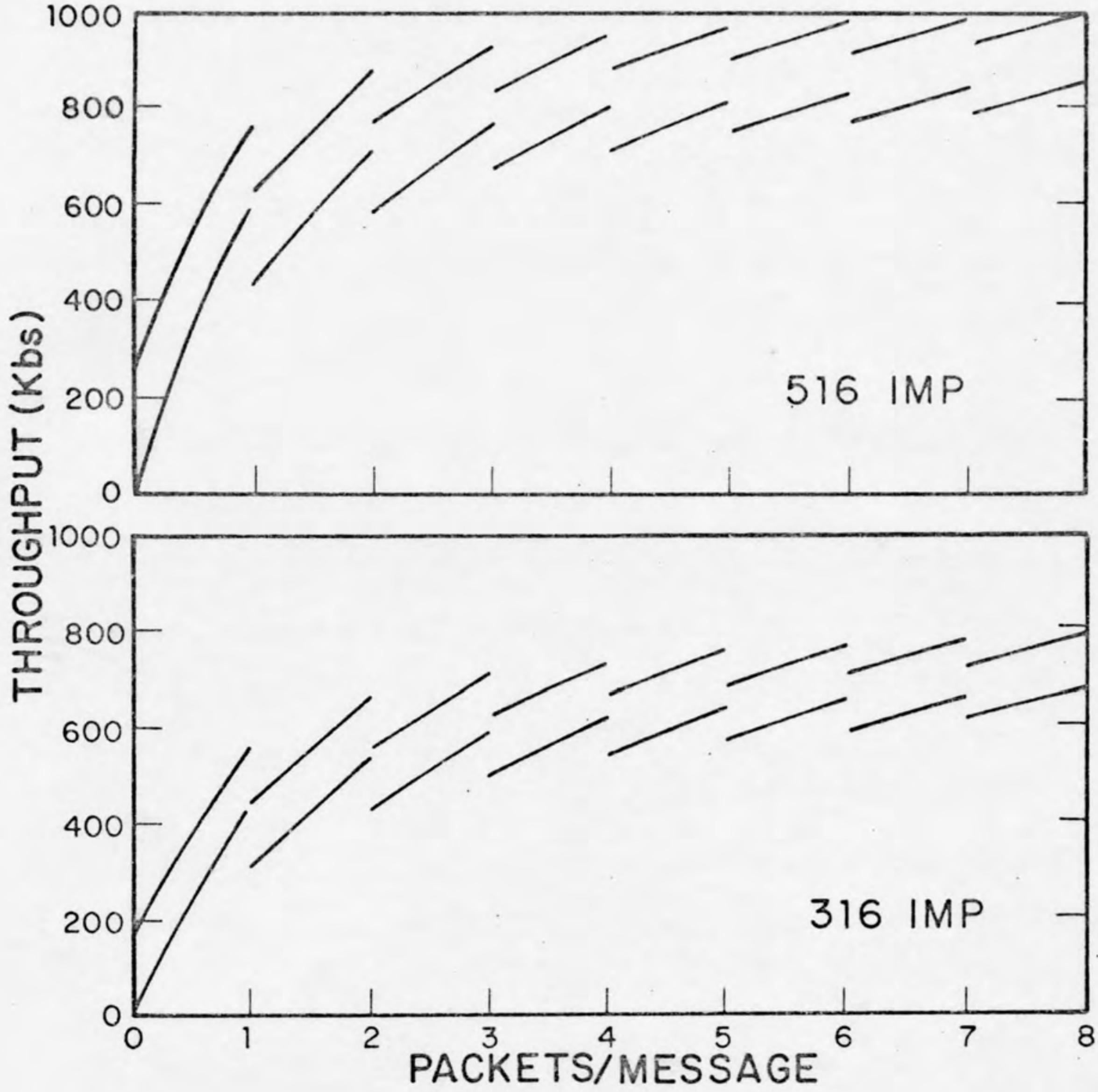

Figure 2-4 IMP Processor Bandwidth vs. Message Length

<u>The Effective Bandwidth of the Network Circuits.</u> In this section we will consider some of the factors acting as overhead to reduce the effective circuit bandwidth for network lines. That is, we wish to catalog all the kinds of transmission that take place on the network lines that are not actual Host data bits, and from this accounting determine which factors are the dominant ones. There are three basic kinds of line overhead:

1. Line overhead in bits/packet.
We have detailed two components of this overhead earlier, Bs and Bh.

2. Line overhead in bits/message.
In the terminology we have been using, the RFNM is overhead on a message basis, and under our assumption that it is a minimum length packet, it contributes a number of bits equal to Bs+Bh.

3. Line overhead in bits/second of network system traffic.
Some examples of this kind of traffic can be given, along with the approximate order of magnitude for the traffic rate in the ARPA network:

routing	1000 bits/sec
line alive/dead	100 bits/sec
NCC status reports	10 bits/sec
core reloads and dumps	1-10 bits/sec

We will make the assumption that routing messages are the primary source of traffic other than messages between Hosts. For this reason, we will define the variables

Br = the number of bits in a routing message

Fr = the frequncy of routing messages [1/sec]

BWCr = Br*Fr = the bandwidth of the circuit used for routing

and ignore the other components of periodic overhead.

We can now total the overhead from all considerations, by converting to a common dimension, bits/sec. In order to do this, we must introduce another variable,

Fp = the number of packets per second. We then can express the total bandwidth of the circuit given to overhead as

$$BWCo = \underbrace{Br*Fr}_{\text{routing}} + \underbrace{(Bs+Bh)*Fp}_{\text{packets}} + \underbrace{(Bs+Bh)*Fp/P}_{\text{RFNMs}}$$

We can rewrite this as

$$BWCo = BWCr + (Bs+Bh)*Fp*(P+1)/P$$

In comparison with this overhead rate is the actual data rate on the circuit,

$$BWCd = Bd*Fp$$

We can evaluate the fractional overhead percentage as

$$BWCo/BWCd = (BWCr)/(Bd*Fp) + (Bs+Bh)*(P+1)/(Bd*P)$$

Another quantity of interest is the maximum data rate that can be attained for given values of the systems parameters. Given a circuit with bandwidth BWC, the maximum data rate occurs when Fp is at a maximum,

$$Fp(Max) = (BWC-BWCo)/Bd$$

Substituting the expression for BWCo above, we get

$$Fp(Max) = (BWC-Br*Fr)/(Bd + (Bs+Bh)*(P+1)/P)$$

The numerator indicates that the routing message bandwidth comes off the top, leaving a reduced effective bandwidth, which is then used for both data bits and packet and message overhead bits. The following graphs show the behavior of some of these variables; the variable plotted is Fp(Max)*Bd, that is BWC-BWCo, which is the maximum data rate for a given message size.

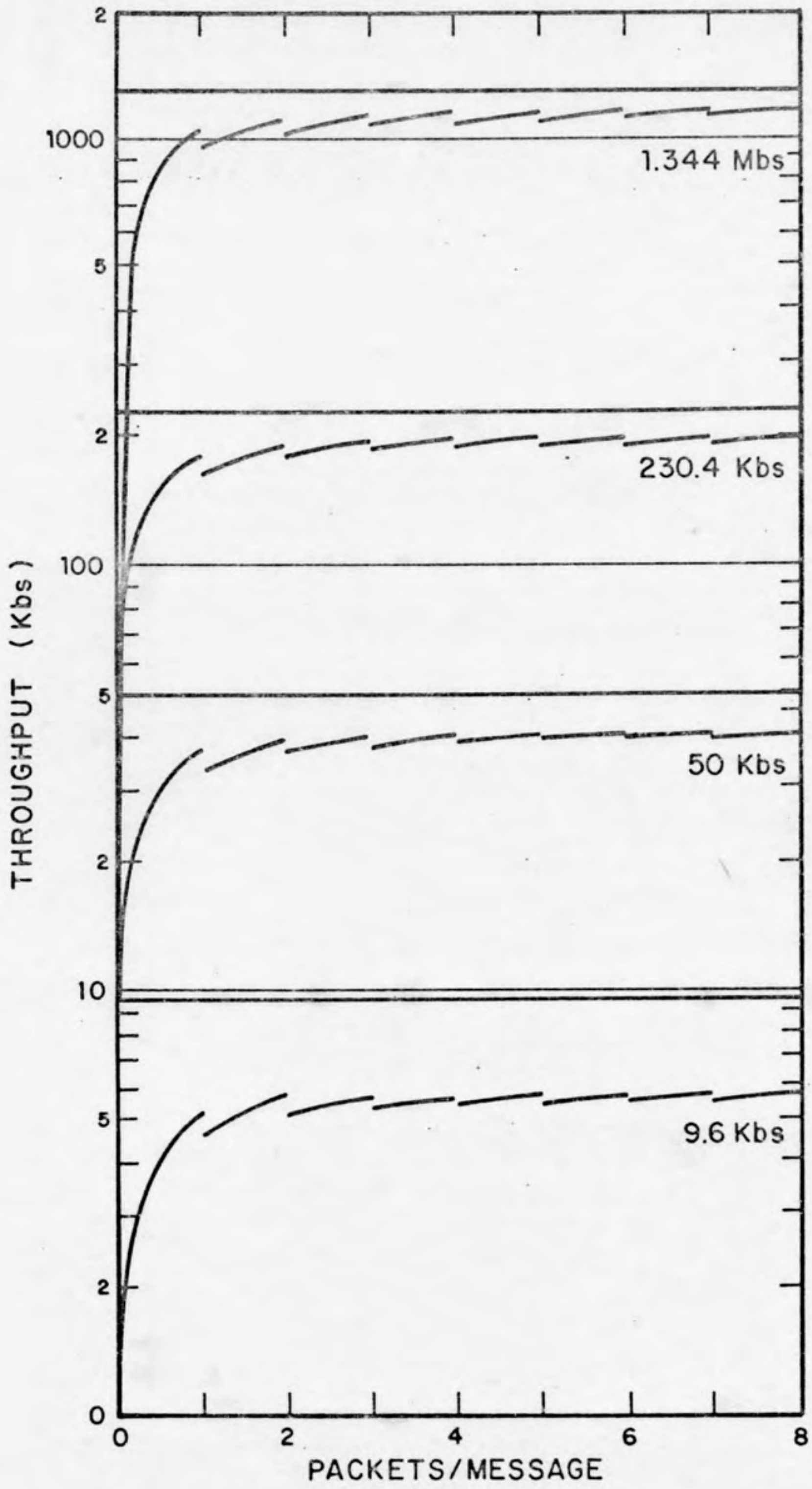

Figure 2-5 Effective Circuit Bandwidth vs. Message Length

The Packet Buffering Required for Network Circuits. We now turn to an examination of the number of packet buffers required to keep a communications circuit fully loaded. This number is a function not only of line bandwidth and distance but also of packet length, nodal delays, and acknowledgement strategy. We will assume here that the node operates in a manner similar to the IMP in the ARPA Network; it buffers each packet that it transmits until it receives an acknowledgement, meanwhile transmitting other packets to utilize the circuit efficiently. If it does not receive the acknowledgement in the expected time, it retransmits the packet. The expected time for an acknowledgement to return is the sum of:

1. Tp = the transmission time for the packet , a function of line bandwidth.
2. L = speed-of-light delay for the first bit of the packet to arrive at the other node, a function of line length.
3. C = Cr+Ct = the processing delay in the other IMP, to receive the packet and return the acknowledgement.
4. Dq = queueing delay for the returning acknowledgement, the time it waits for any other transmissions ahead of it.
5. Ta = the transmission time for the acknowledgement.
6. L = speed-of-light delay for the first bit of the acknowledgement to arrive at the first node.
7. Cr = the processing delay for the acknowledgement.

Our first simplifying assumption is that the processing times are small compared to the other delays and can therefore be ignored:

$$Cr = Ct = 0.$$

We can then state that the number of packet buffers needed to keep a circuit fully loaded is

$$BFp = (Tp + L + Dq + Ta + L)/Tp$$

This can be rewritten in somewhat more meaningful form as

$$BFp = 1 + 2*L/Tp + (Dq+Ta)/Tp$$

This expression indicates that one buffer is always necessary, to account for the packet transmission time itself. More buffers may be required if the circuit is long compared to the packet transmission time, or if the acknowledgement transmission takes a long time compared to the packet. Stated differently, the number of buffers needed to keep a line full is proportional to the length of the line and its speed, and inversely proportional to the packet size, with the addition of a constant term.

In order to proceed further with the analysis, we need to introduce two new terms:

Ts = the transmission time of the shortest allowable packet

Tl = the transmission time of the longest allowable packet

We also need to postulate a traffic mix of long and short packets, with x/y the ratio of short packets to long packets in the channel. Now we can define Dq and Ta in terms of Ts and Tl. We make a worst-case assumption for Dq:

Dq = Tl, the acknowledge has highest priority (equivalently, it piggybacks on all packets), but it must wait for the transmission of a maximium-length packet which has just begun.

The assumption for Ta is rather arbitrary:

Ta=(Ts+Tl)/2, the acknowledgement piggybacks on an "average" length packet.

We now state the result for the number of packet buffers required given the above set of assumptions:

BFp = 1 + (2*L + Tl + (Ts+Tl)/2)/Tp

Using the ARPA Network values of Ts and Tl given above in Table 2-2, and choosing a variety of line lengths and traffic mixes (shown as the ratio of short packets, S, to long packets, L), we can present some numerical results as a family of curves shown below. Note that the knee of the curves occurs at progressively shorter distances with increasing line speeds. In fact, if we define the knee to occur when the linear term is equal to half of the constant term, then the knee occurs when

L = (Ts + 2*Tp + 3*Tl)/4,

or for a line length of 225*10**6/BWC miles. The constant term dominates the 9.6 Kbs case, and it is almost insignificant for the 1.4 Mbs case. Note also that the separation between members of each family of curves remains constant of the log scale, indicating greatly increased variations with distance.

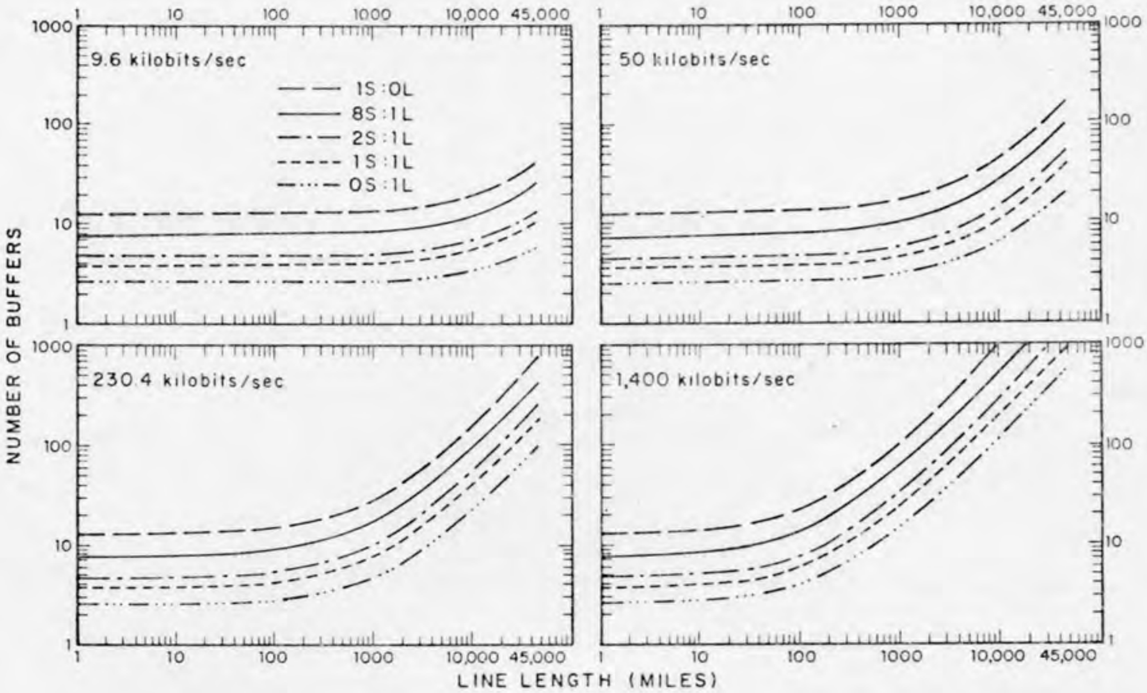

Figure 2-6 Packet Buffering for Full Line Utilization

<u>The Message Buffering Required for Network Paths.</u> This section takes up a topic which closely parallels that of the last section. Here we will examine the number of messages needed to obtain full bandwidth over a network path of many lines. That is, we will compute how many messages must be in flight between two nodes in order to keep all the intermediate lines fully loaded. Actually, the best one can do is to keep all the lines in the path of the lowest circuit bandwidth fully loaded. It turns out that this analysis is quite simple given all the definitions of the

preceding sections. The number of message buffers needed is computed by taking the round trip delay for a message and dividing it by the time taken to transmit a single message. That is,

$$BFm = D(MRT)/(P*(Tp+Ct))$$

Using equation 2.2.1-3, we can obtain a more detailed expression for the simple case of equal delay at each hop:

$$BFm = (P-1)/P + H*(2*(L+C)+Tp+Tr)/(P*(Tp+Ct))$$

It is clear from this expression, and on intuitive grounds, that BFm is a minimum for maximum length messages, that is for large P. The following curves show the dependence of BFm on line characteristics and on the length of the network path, for P=8 and ARPA Network values of the parameters. For each of four line speeds, the buffering requirements are plotted for network paths made up of a number of land lines (the length of the lines is given with each curve). Also shown are the requirements for the same network paths with the addition of one satellite link running at the same bandwidth as the land lines.

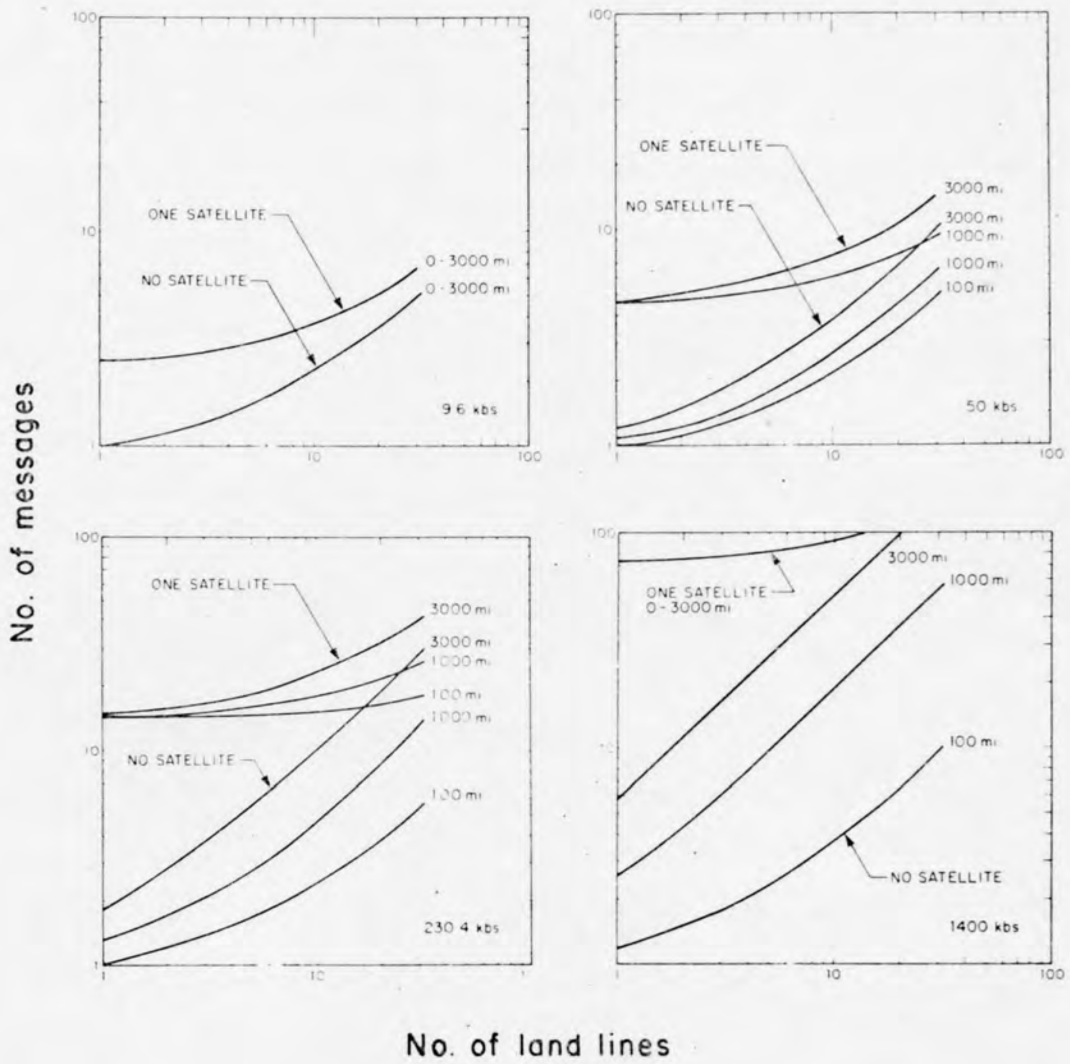

Figure 2-7 Message Buffering for Full Path Utilization

The consequences of the above discussion are several. For the communications subnetwork, it means that the nodes must be able to do bookkeeping on several messages in flight between two nodes. Further, the network must buffer these messages in the memory of the source or destination node for the duration of their flight, in addition to the packet buffering that takes place instantaneously at the intermediate nodes along the path. This has some important ramifications for the design of the software for the node computer. A second set of issues is the effect of message buffering on Host computers. It is clear that the communication protocols that the Hosts use must also be engineered to the parameters of the network, if they are to obtain full throughput levels.

<u>High Throughput for Long Data Transfers</u>. The several topics examined in this section all point to a single conclusion: the larger the packets in a message, and the larger the messages in a data transfer, the higher the level of throughput that is potentially attainable. For reasons of processor overhead, circuit overhead, and buffering considerations within the nodes, it is always better to have long packets and messages if high data rates are desired.

<u>The Tradeoff between Low Delay and High Throughput.</u> It is clear that the two goals of low delay and high throughput are often in conflict because the node has limited resources with which to service its Hosts and lines. It is difficult to guarantee both low delay and high throughput to several competing sources. The approach taken in the ARPA Network is as follows: The IMP program has been designed to perform well under bimodal traffic conditions. It provides quick

delivery for short interactive messages and high throughput rates for long files of data. This optimization of the program for a specific model of traffic behavior occurs at many levels, and is essential for balanced performance characteristics.

2.2.3 Cost

In this section we present some of the primary issues regarding the cost of computer networks. There are two kinds of costs considered here: the cost of the actual network components, and the cost of the use of the network. The first cost is a measure of the expense of connecting some component to the network, and the second is a measure of the expense of utilizing the resources of that component. We are concerned here with outlining the effects of these costs on the balance among the various system parameters of the network. The more specific effects routing algorithms have on the cost of networks are discussed in section 2.3, on routing.

Low Cost for Network Connectivity. The first point to take up in consideration of the cost of networks is the cost of connectivity. The basic variables are circuit costs and node costs. Some typical values for these costs in an actual network may be illustrative; the values for the ARPA Network are as follows:

Bandwidth Kbs/sec	Termination Cost $/month	Line Cost $/month/mile
9.6	650	.40
19.2	850	2.50
50	850	5.00
230.4	1350	30.00

Table 2-3 Some Representative Line Costs

Machine Type	Cost	Configuration (Kbs total)
516 IMP	$100K	up to 4 Hosts (400)
		up to 5 lines (800)
		up to 7 devices total
316 IMP	$50K	up to 4 Hosts (300)
		up to 5 lines (600)
		up to 7 devices total
316 TIP	$100K	up to 2 Hosts (300)
		up to 3 lines (600)
		up to 64 terminals (100)

Table 2-4 Some Representative Node Costs

Several implications follow from these figures:

1. The layout problem for networks is an important one, since reductions in the total number of circuit miles in a network represents substantial dollar savings.
2. Low cost comes at the expense of both low delay and high throughput if lower line speeds are chosen.
3. Low cost also comes at the expense of low delay and high throughput if fewer lines are used, since more traffic is forced to use each line.
4. As a consequence of these points, some network designers, notably the NAC (NAC 70a, NAC 70b), attempt to find a relatively low cost network layout and then test it by simulation to determine if average message delay is below some threshold and the throughput obtained (when all nodes send to all other nodes) is above some threshold.

Low Cost for Network Use. A second aspect of the cost of networks is the cost of the use of the network, which may be counted in one of several ways:

1. \$/bit, \$/character, \$/packet, or \$/message costs for the shipping of data through the network.
2. These charges may be rated per mile or per hop or may be distance-independent.
3. There may be different grades of service at different costs.

These costs are the translation into dollars of the utilization rates for various network resources. These resources can be cataloged as follows:

1. line bandwidth
2. node processor bandwidth
3. node storage

2.2.4 Reliability

In this section we present some of the primary issues regarding the reliability of computer networks. In parallel with the discussion of network cost above, there are two basic topics examined here: the reliability of the network connections themselves, and the reliability of the network services. In the first instance, we are interested in how reliable the network components are. In the second case, the subject is the reliability of the use of the network facilities for data transmission. Again, we are focussing in this section on the broad design issues which affect network performance as a whole. In section 2.3, on routing, we examine closely the effects that routing algorithms can have on network reliability.

High Reliability of Network Connectivity. One can attempt to minimize the probability that a line or node will be inoperative, and thus to reduce the probability that a node cannot communicate with the rest of the network. In practice, this is a matter of maintaining a high mean time between failures and a low mean time to repair. One can also measure reliability in terms of the number of nodes and/or lines necessary to disconnect the network, taking into consideration the probability of the various events. Alternatively, one can consider the size of the components of the disconnected network, or the fraction of node pairs not connected by any network path.

A different level of solution to the problem of network reliability is redundant network design, primarily in the layout of the circuits connecting the nodes. In this way, a network can be constructed which is much more reliable as a whole than any one component. In the ARPA Network, a design constraint has been that an IMP must be connected to the network by at least two circuits, so that the probability that an IMP cannot communicate with the network is very small. Of course, this principle can be applied to other components in the network as well. The node computers can be backed up with alternate computers, or may be equipped with redundant interfaces and processors. The Host computers also may wish to be connected to the network at more than one point by means of separate communications facilities.

<u>High Reliability of Network Use.</u> The reliability of the message processing can be measured in terms of the percentage of messages delivered, or the detected error

rate, or the undetected error rate. Measures to improve the reliability of message processing range from error detecting and correcting hardware to redundancy in software to backup message storage. The costs of these approaches are multiple: they add to the complexity of the system, and may degrade its performance, in addition to representing a dollars cost. In general, the communications subnetwork becomes more costly as these measures of reliability are improved. At some point, it becomes appropriate to pass the cost of these improvements on to the user. However, a minimal level of reliability is necessary for the operation of the communications subnetwork. Guarantees concerning a grade of service better than this minimal level might reasonably cost more.

<u>The Tradeoff between Low Cost and High Reliability.</u> It is clear that there is a natural tradeoff between low-cost networks and high-reliability networks. This tradeoff exists in building either sparse networks or highly-connected networks, and in providing special mechanisms to ensure the reliable transmission of data or choosing not to implement such safeguards. In short, the price for reliability must be paid somewhere, either in the actual cost of constructing and maintaining the network, or in the cost to the user of unreliable network service. It is difficult to generalize, but it may often be the case that a low-cost network without sufficient measures for

reliability may prove more costly to use in the long run than a network with a higher cost for higher network reliability. Stated differently, it may be cheaper to build a network to be fault-tolerant, redundant, and error-detecting than to build these measures into each user process that communicates with the network.

2.3 Routing Algorithms

In this section we consider the process of designing a routing algorithm for use in a computer network. We list the key requirements that any good routing algorithm must meet. We detail the specific functions that the algorithm must perform, and the criteria for judging its effectiveness. We also introduce the basic costs associated with any routing strategy. In the following section, we examine the routing algorithm orginally used in the ARPA Network, as an example of a balanced design, and from the point of view of the changes necessary to increase its effectiveness under changing requirements.

An outline of the process of designing a routing algorithm is shown in Figure 2-8. The two major activities are specification of the algorithm, in terms of its input, its output, and its processing characteristics, and, secondly, evaluation of the algorithm in terms of its performance and cost. The subsections which follow take each of these functions in turn and examine them closely. We are not concerned here with the choice of a particular routing algorithm; the options available are discussed in Chapter 3. Rather, we are pointing out the issues which must be addressed during the design of a routing algorithm. In identifying these areas of concern, it is hoped that the choice between alternative routing proposals can be made in