Set FT

B O L T   B E R A N E K   A N D   N E W M A N   I N C

C O N S U L T I N G · D E V E L O P M E N T · R E S E A R C H

Report No. 2161

A STUDY OF THE ARPA NETWORK DESIGN AND PERFORMANCE

August 1971

Robert E. Kahn
William R. Crowther

Submitted to:

Director
Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

Attention: Dr. L.G. Roberts

This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC15-69-C-0179.

Report No. 2161

A STUDY OF THE ARPA NETWORK DESIGN AND PERFORMANCE

August 1971

Robert E. Kahn
William R. Crowther

Submitted to:

Director
Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

Attention: Dr. L.G. Roberts

This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC15-69-C-0179.

PREFACE

This document reports on a study which may contribute to the planning of IMP software modifications. This document does not describe the current status of the operational IMP software, nor does it describe a specific planned modification to that software.

Cambridge, Mass.
August 1971

TABLE OF CONTENTS

FIGURES

1. INTRODUCTION

This document reports on the results of a study by Bolt Beranek and Newman Inc. of the design and performance of the ARPA Network. Two important objectives of the study were to evaluate the ability of the network to handle heavy traffic loads and to increase our understanding of the relation among the number of packet buffers, the design of the system, and the system's performance. As one result of the study, several improved system algorithms for handling traffic were developed. The study was conducted over a six-month period from September 1970 to February 1971; participants were Dr. Robert E. Kahn and Mr. William Crowther of BBN and, in the latter phase, Dr. Robert Sittler of ARCON Corp., under contract to BBN.

The ARPA Network has been operational since the fall of 1969. During this time it has evolved from a four-node initial network to a fifteen-node network, as of July 1971, without significant modification to the original system design. A year and one half of experience by BBN and the participating Host organizations both in operating and in using the net has demonstrated that the IMP subnet will handle normal interactive traffic of all message lengths when each Host is prompt in accepting and delivering network traffic. However, under heavy traffic load, as may occur with many continuous file transfers or with unresponsive Hosts, the network performance can become substantially degraded.

These design and performance issues were reviewed during the initial phase of the study with the aid of data obtained

through simulation experiments and field testing. As the study effort proceeded, a set of improved system algorithms was developed to upgrade the system performance. Toward the conclusion of the study, a rough estimate was made of the amount of storage required for efficient system operation with these algorithms. A final evaluation of the needed core memory has been deferred, however, awaiting a preliminary coding of the algorithms.

2. DISCUSSION OF CURRENT OPERATION

In our study we identified four conditions that may lead to performance degradation or system malfunction:

1) *Reassembly Lockup* — a continuous flow of multi-packet traffic on many links to a given destination may produce an excessive demand for utilization of the limited supply of reassembly buffer space at the destination IMP. This conflict results in an increased frequency of retransmission to the destination, making for inefficient use of its phone lines, and degrading the Host/Host throughput of data. It may cause this throughput to be reduced to zero, in which case the network will have entered a null state, called reassembly lockup, from which it cannot logically recover until an automatic reset occurs about one minute later.

2) *Store-and-Forward Lockup* — traffic that is flowing on many links may cause the store-and-forward buffers to be filled in the IMPs at each end of one or more circuits and cause the traffic flow on that circuit(s) to halt. This condition is called store-and-forward lockup.

3) *Congestion* — The RFNM (Ready for Next Message) mechanism prevents a single link from itself using more than eight network buffers, but does not prevent congestion resulting from the combined effects

of many links in operation. A Host that does not accept incoming messages at a rate at least equal to the arrival rate of messages may cause some or all of the network's store-and-forward buffer storage to become filled with backed-up traffic and may cause other Host/Host traffic to encounter unnecessary or excessive delays.

4) *Routing* — the routing algorithm does not always make efficient use of two or more paths between a pair of Hosts. A maximum Host throughput in excess of 30 to 40 kbps may be achieved only when two or more paths, each containing at most two or three IMPs, exist between the Hosts, or when the two Hosts are connected to the same IMP.

Our study has led us to consider whether and how to modify the operational IMP system in order to obtain the desired improved performance. A phased modification of the program will undoubtedly occur with early emphasis on correcting reassembly lockup problems, but the implementation plans have not yet been settled.

In this section, we present a brief technical description of the four conditions described above. Several performance curves are included; these curves summarize data obtained during both field testing and simulation runs. The field testing was performed over a period of many months during 1970 and was supported by additional testing in the BBN test cell. The simulation was performed with a program that models the IMP

program in detail and runs at approximately real time on a Honeywell DDP-516 computer. The simulation is basically straightforward and is not otherwise discussed in this document.

2.1 Lockup

The term lockup denotes one of several null states that the network can enter and from which it cannot recover without being reset. Two types of lockup may occur with the current net design: reassembly lockup and store-and-forward lockup.

2.1.1. Reassembly Lockup

In the current system design, one packet of a multi-packet message can be logically prevented from reaching the destination IMP even though reassembly buffer space is already reserved for that packet and a path exists for the packet to reach the destination. Such a situation arises whenever the destination's neighboring IMPs become filled with store-and-forward packets which are also headed for that destination, but are continually rejected by the destination IMP for lack of additional reassembly space. This filling of the neighbors can block a partially reassembled multi-packet message from being completely reassembled until the missing packet or packets have passed through the neighboring IMPs; but the neighboring IMPs will not allow any packets to pass through until one or more of its packets are first accepted by the destination. The resulting deadlock is called reassembly lockup.

Reassembly lockup is the most serious network deficiency. It is easily made to occur by a Host and may cause the network

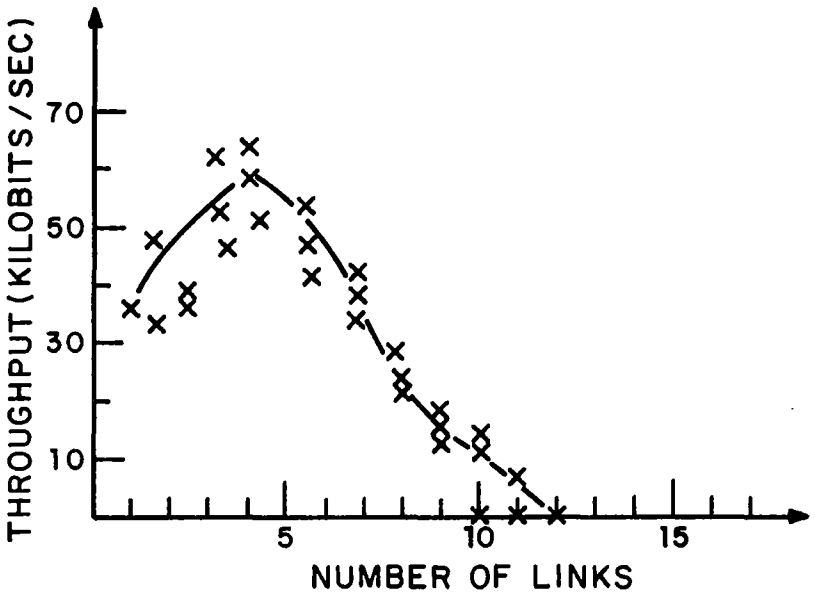
flow to halt. In particular, the attempt to achieve high throughput on many links to a given destination will ordinarily cause reassembly lockup to occur shortly thereafter.

A curve of throughput versus number of links is shown in Figure 1 for traffic consisting of 8-packet messages sent on each link immediately upon receipt of the previous RFNM. The network topology and traffic flow pattern are also shown in the figure. For this case, there are 21 store-and-forward buffers and 32 reassembly buffers available in each IMP. The curve of throughput rises to approximately its theoretical maximum, then decreases as additional links are used, and becomes equal to zero after lockup occurs. In each case where the throughput is indicated as zero, lockup typically occurs within tens of messages after the start of a transmission.

2.1.2 Store-and-Forward Lockup

Consider a set of IMPs I_1, I_2, ...,I_N each filled with store-and-forward packets, and assume that reassembly buffer space is available for any packet that reaches its destination. If the packets in each IMP I_k, k=1,..N are continually rejected by its neighboring IMPs because the neighbors' store-and-forward space is also filled, then a store-and-forward lockup is said to be present.

It is possible for a store-and-forward lockup to occur in the network, but the probability of its occurrence is low. It may be made to occur in carefully arranged testing, but it requires either a simultaneous occurrence of input traffic on



21 S/F BUFFERS
32 REASSEMBLY BUFFERS
8000 BIT MESSAGES

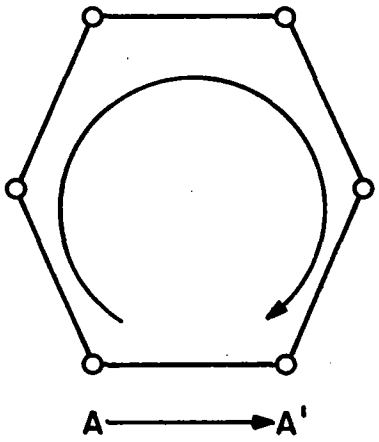
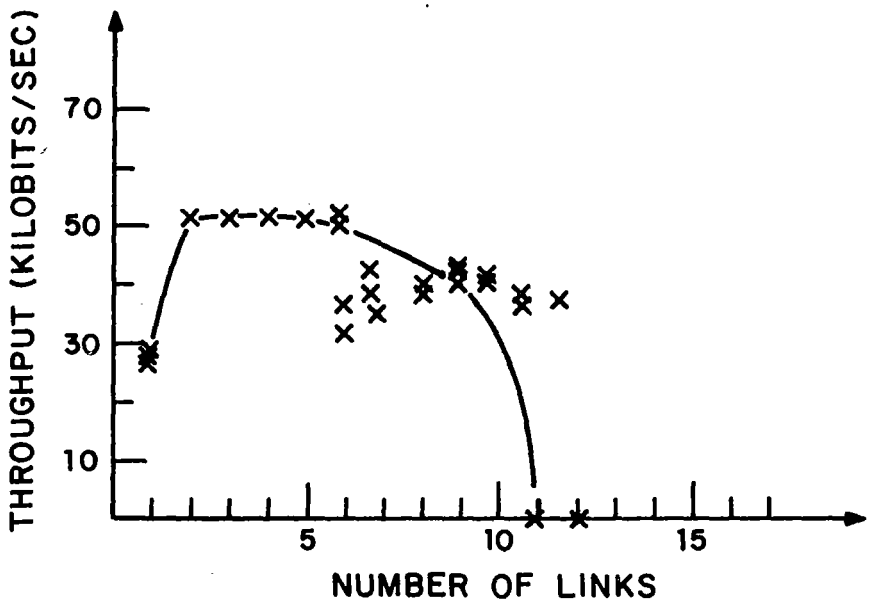


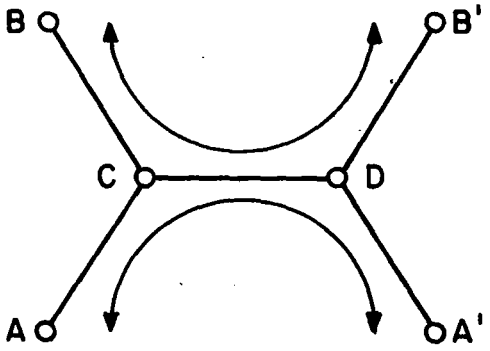
FIG.1 REASSEMBLY LOCKUP

many links or a network topology that funnels large amounts of traffic in both directions onto a single path. Two types of store-and-forward lockup, known as direct and indirect, are indicated below.

In Figure 2, we illustrate a case of direct store-and-forward lockup. Two-way single-packet traffic (1000 bits/packet) flows between A and A' and also between B and B' and is constrained by the network topology to use the circuit between IMPs C and D. All the buffer storage in IMP C can become filled with packets on the output queue to IMP D, and all the buffer storage in IMP D can become filled with packets on the output queue to IMP C. In this situation, IMPs C and D will be engaged in a direct confrontation in which both IMPs receive no acknowledgments for transmitted packets and continue to discard all incoming packets for lack of additional store-and-forward space. A curve of total throughput versus number of links in use per IMP is shown in the Figure. The different nature of the curve below and above 6 links indicates the point at which conflicts begin to occur.

In Figure 3 we illustrate a selected net involving 8 IMPs that can enter an indirect store-and-forward lockup. The arrows indicate an offered traffic load from each IMP to the IMP two hops away counterclockwise. Each IMP can become filled with packets destined for the IMP one hop away counterclockwise. Let us assume the IMPs are consecutively lettered from A to H counterclockwise. All the packets in A can be on the queue to B, all the packets in B can be on the queue to C, all the packets





21 S/F BUFFERS
∞ REASSEMBLY BUFFERS
1000 BIT MESSAGES

FIG.2 DIRECT STORE-AND-FORWARD LOCKUP

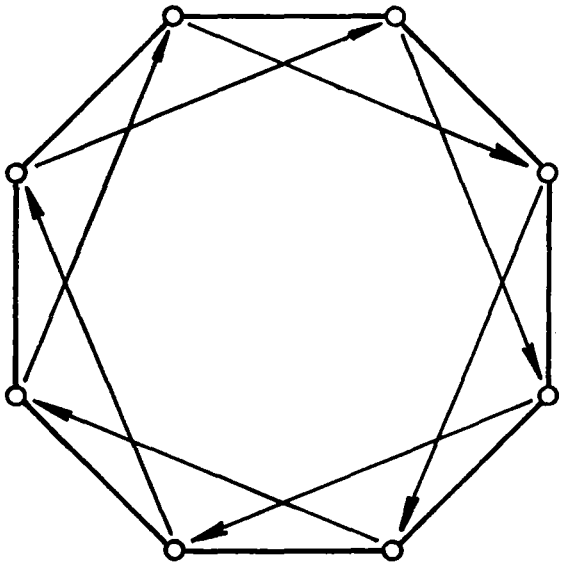


FIG.3 INDIRECT STORE-AND-FORWARD LOCKUP

in C can be on the queue to D and so forth. Finally all the packets in H can be on the queue to A. There is no mechanism in the current system for allowing any of these IMPs to deliver their packets to the destination, or to recover from this situation.

This net can enter a pure indirect store-and-forward lockup or a combination of direct and indirect lockup depending on the rate on the Host line, the routing update, the number of store-and-forward buffers, etc. The traffic load shown in Figure 3 was simulated using an early model of the ARPA Network, with 21 store-and-forward buffers and 1000-bit single-packet traffic sent on each link as soon as possible after the RFNM arrived. The simulation showed a lockup with 32 links in use per IMP, but no lockup with 1, 2, 4, 8, or 16 links in use.

2.2 Congestion

Congestion typically arises when a large amount of traffic is arriving at a destination Host at a sustained rate in excess of the rate at which the Host is accepting the traffic. A particularly important case is a Host that is temporarily accepting no network messages, but whose ready line has not been turned off, and to which a large number of messages are in transit through the net. The presence of network congestion may cause substantial delays to be encountered by other traffic in the net or even cause all traffic to halt.

The IMP RFNM mechanism guarantees that only one message may be in the net on a single link. This message can neither

use all the network buffer storage nor cause congestion by itself. However, the combined messages on many links to a given destination can cause the reassembly storage at the destination IMP to become filled and cause messages to become backed up into the net (and possibly stagnate there) when the Host is not accepting messages at a sufficiently fast rate.

Congestion may be expected to occur from time to time under heavy traffic loads, particularly just after a Host failure. A poorly designed Network Control Program (NCP) may also contribute to causing congestion by introducing unnecessary or unusually lengthy delays before the Host takes each message.

2.3 Routing

The routing algorithm does not guarantee the effective use of multiple paths in transmission between any two Hosts. In particular, only one output line is used at a time to route packets to any given destination. Consequently, the achievable throughput between two Hosts may be significantly reduced from its maximum value, often by a factor of two or more.

The routing algorithm relies upon queue length measurements to estimate the minimum time to reach each destination and selects the corresponding output line to route packets to that destination. However, the queue lengths can change much faster than the routing information can be propagated from one IMP to its neighbors, much less to all IMPs in the net. As a result, under heavy load, the routing selection is often no better than a poor or random choice, resulting in the choice of poor routes, inefficient use of communication facilities, and significantly lower Host/Host throughput than is theoretically possible.

3. ALGORITHMS TO IMPROVE SYSTEM PERFORMANCE

3.1 Congestion Control

The mechanism described below prevents the occurrence of congestion and reassembly lockup by:

1) Discarding all single-packet messages from the net that the destination IMP is unable to accept due to insufficient reassembly space and holding a copy of the packet at the source IMP for later retransmission, if necessary.

2) Preventing each multi-packet message from entering the net until space has been reserved for it at the destination IMP.

Whenever reassembly space becomes unavailable, the flow of traffic to that destination is automatically shut off at each source before any congestion can occur. The actions taken for single-packet messages and for multi-packet messages are described separately below.

3.1.1 Single-Packet Message

The source IMP distinguishes a single-packet message from a multi-packet message if the end of message signal occurs at the end of the first packet. The IMP will hold a copy of this message and dispatch the original as a discardable message to the destination IMP.

If an unreserved buffer exists at the destination IMP when the discardable message arrives, the message is accepted and placed on the output queue to the Host. After the packet is sent to the destination Host, a RFNM is returned to the source IMP, which then discards its copy of the single-packet message and passes the RFNM along to the source Host. If space is not available at the destination IMP when the discardable message arrives, it is immediately discarded and an entry is made into the receive link table to indicate that space is desired when available. When space does become available, a buffer is reserved and a message indicating 1 reserved buffer is returned to the source IMP, which will then transmit its copy of the priority message, and release the buffer.

A small number of buffers (initially four) are set aside in the source IMP for the express purpose of storing single-packet messages while the IMP waits for space to be allocated in the destination IMP. These buffers are never used to hold packets of multi-packet messages. If all these buffers become occupied, the Host-to-IMP line will be stopped until at least one of them is again available. The use of a small number of these buffers is sufficient to make it normally unnecessary to stop the Host-to-IMP line, since a RFNM for one message will typically return before all the buffers have been occupied.

A single-packet message containing fewer than 80 bits of data is defined to be a priority message and is given special handling by the IMPs to speed delivery.

3.1.2 Multi-Packet Messages

The source IMP will identify a multi-packet message by the absence of an end of message indication at the end of the first packet. Upon receipt of the first packet, the source IMP will stop the Host-to-IMP line and dispatch a small discardable message to the destination IMP asking to reserve 8 buffers in reassembly. When 8 buffers are available, the destination IMP will place them in reserve and then return a message to the source IMP to indicate the reservation. When this message is received by the source IMP, it will then restart the Host-to-IMP line and attempt to transmit the entire multi-packet message.

The first packet of a multi-packet message is held by the source IMP while waiting for space to become available at the destination. The Host-to-IMP line is stopped after the first packet is received (provided space was not previously made available at the destination, as for example during high-bandwidth transfers) and restarted when space is known to be available. This technique typically increases the time to complete the transfer of an occasional 8-packet message or the first of a long sequence of messages from the Host to the IMP by tens of milliseconds in a lightly loaded net.

This additional setup delay is not present for each message after the first of a continuous stream of multi-packet messages. Whenever a multi-packet message is received by a destination IMP, it will not return a RFNM to the source until the first packet has been sent to the Host and an additional 8 buffers have been reserved for that source. The source IMP

will keep a record of the new buffer reservation for about 125 msec after completing the transfer of the RFNM to the Host. For each such record of a RFNM, the source IMP will allow one multi-packet message to that destination to be transmitted without first sending a discardable message and without halting the Host line.

A Host that wishes to obtain high throughput must complete the transfer (into the source IMP) of the first packet of its next multi-packet message to the given destination within 125 msec after receipt of the RFNM. Any sequence of single-packet or multi-packet messages which fits into the 125 msec interval may precede this next multi-packet message. By continuing to transmit each successive message immediately upon receipt of the RFNM, a Host will be able to avoid the setup delay and thus allow high throughput of data to be achieved. If the source IMP does not receive a multi-packet message for the given destination within the 125 msec time period, it will discard its record of the buffer reservation and dispatch a short message to the destination IMP to free the 8 reserved buffers, and the source Host may then experience the short setup delay when it sends its next multi-packet message.

3.2 Routing Algorithm

The algorithm described below allows the capacity on one or more paths connecting Host/Host pairs in the ARPA Network to be used efficiently. The algorithm is able to achieve efficient use of multiple paths by providing for each IMP to select more than one output line to each destination. In this scheme, each

IMP individually selects the set of output lines on which it will route traffic based on 1) status information received from neighboring IMPs, and 2) the traffic pattern it encountered over the last several seconds. The traffic flow along these output lines is adjusted by a metering procedure for stability and to obtain increased overall traffic flow in the net.

Host/Host traffic is normally routed by the IMPs over a path containing the fewest IMPs between the source and destination. When one or more circuits on that path become fully occupied, the routing algorithm attempts to route further traffic via a shortest alternate route with unused capacity.

The routing algorithm is designed to avoid establishing alternate routes in response to rapid changes in traffic flow either within the net or to and from the Hosts. Rather, there are controls on both the rate of increase of traffic on each path and the interval before alternate routes will be established. By smoothing the rate at which traffic may increase, this algorithm is able to provide alternate routes only in response to sustained demands for high throughput, either from the Hosts or within the net due to concentration of traffic flow.

We assume that the queue lengths in each IMP are constrained to be small and, as a consequence, cannot provide much useful information. In addition, this information cannot be propagated about the network sufficiently fast relative to changes in queue length and is not used in the routing computation. A number of specific properties of the algorithm are listed below; a more detailed description follows later.

1) The routing selection is performed independently by each IMP, based on information received from its neighbors and the traffic pattern it encountered.

2) The algorithm attempts to guarantee that the individual routing decisions are globally sensible.

3) Periodic, half-second updating of routing tables always occurs. In addition, more rapid updating of these tables is allowed to occur when important status changes occur, such as circuits becoming fully occupied or breaking.

4) In computing the selection of routes, each IMP considers the network to be decomposed into the union of many identical and overlapping subnetworks, one per destination, with separate routing computed for each subnetwork.

5) Minimum Hop routing is used in an unloaded net. That is, in each subnetwork, the IMP selects an output line on a path with the fewest number of IMPs to the given destination.

6) In a loaded net, the routing algorithm avoids trying to divert additional traffic over paths containing fully occupied circuits whenever possible. Rather, the IMPs attempt to find and use a path with the fewest number of IMPs and no fully occupied circuits.

7) Most changes in the selection of routes for each subnetwork are allowed to occur only infrequently, on the order of seconds apart. Only the sustained flow of traffic according to a new traffic pattern or an IMP or circuit failure will normally cause a change to occur.

8) If sustained heavy traffic for a given destination is received by an IMP, it will establish additional paths (if necessary) one at a time and at intervals seconds apart. The packets will then be able to leave the IMP on any of several lines. An IMP will only establish an alternate path to a destination if the path contains no fully occupied circuits.

9) Traffic flow on any line in a subnetwork may proceed in only one of the two directions at a time, not both.

10) Each line in a subnetwork has a direction associated with it corresponding to the allowed direction of flow. Directions may be changed only infrequently and only by first passing through a neutral state for a few seconds.

11) The maximum allowed traffic through each IMP in a subnetwork will be regulated (metered) so as to change slowly. This metering provides stability and allows the overall traffic flow to be adjusted for increased flow.

12) For each subnetwork, any loops in the routing will be quickly be detected and broken.

A description of the routing algorithm is given in more detail below. We describe, in order, the routing action taken on a packet received by an IMP; the procedure for opening and closing alternate paths; the propagation of routing information between the IMPs; the method used to decide if a line is fully occupied; and finally the metering procedure and loop breaking.

The details of this algorithm are currently being studied and will undoubtedly require some adjustment at a later time. For example, the measures of circuit occupancy may require refinement during coding, and the metering procedure and line priority assignments may require adjustment after both simulation and field testing.

3.2.1 To Route a Packet

Upon receipt of a packet, the IMP first determines if the packet is for a Host at that site, in which case the routing is obvious. If the packet is for a Host at another site, the IMP tests each output line that has been opened for routing to the destination. In particular the IMP first applies a metering test to determine if traffic is allowed to flow, and then applies a buffer allocation test to one or more lines to determine if the buffer may be placed on the output queue for that line.

The lines are tested in the order they were opened for output. The IMP will always be able to open at least one output line to a destination whenever at least one path exists to that

destination. A packet will be placed on the first line that passes both tests, if one exists, and the packet will be acknowledged if necessary. If all lines fail the tests, a new path will be opened, if possible, and used if it passes the tests. Otherwise, the packet will be discarded by the IMP, and an acknowledgment will be returned to the neighboring IMP indicating that the packet was correctly received but discarded by the program. If the packet was from the Host, the Host line will be hung until a line passes both tests. At this time, the details of the line testing are not finalized. A number of strategies are being studied that allow the output line priorities to be dynamically rearranged. This will allow lines to be tested in a new order. The resulting traffic flow appears to be substantially affected by the strategy adopted.

3.2.2 Opening and Closing Routes

Each IMP maintains a table of open output lines on which it is permitted to route packets to a given destination. A new line is added to the table only when a packet arrives and the currently open routes do not pass both the buffer allocation and metering tests and the routing algorithm has specified a useable alternate route.

A useable alternate route is a stable path with excess capacity. A stable path is one that has not changed for a few seconds. If there are no routes at all, the minimum Hop path is also a useable alternate route.

Each IMP knows the identity of both the output line for the minimum Hop path and the output line for the next unoccupied

path to each destination. The next unoccupied path is defined at any time to be the minimum Hop path with excess capacity. The same output line may be indicated for both paths and, in fact, the two paths will often be identical. A count on next unoccupied path is initialized to +7, decremented by one every half second, and the path may be used only after the count reaches Ø. If no route exists and the next unoccupied path has not counted down, the minimum Hop path is tested and used when allowable. Otherwise the packet is discarded.

Entries in the table are timed-out. When a line has had no traffic to a given destination for about 3 seconds, its entry is deleted from the table of open routes to that destination and the above procedures must be used before it may be reinserted in the table.

Each entry in the table represents a line designated for output. A line on which traffic is arriving will never be inserted in the table and, consequently, not used for output. However, traffic may happen to arrive for a destination on a line already opened for output to that destination if, for example, a line at a neighboring IMP has just broken. However, the network will soon adjust to the new topology and the IMPs will agree on one direction for the flow of traffic along this path.

3.2.3 Transmitted Routing Information

Each half second, each IMP transmits to its neighbors a routing message consisting of the minimum Hop, minimum unoccupied Hop and maximum Hop to each destination. The minimum

unoccupied Hop is used to select the next unoccupied path, the minimum Hop is used for detecting disconnected destination IMPs and for routing in the event of circuit failures, and the maximum Hop is used to detect loops in the routing. Whenever a circuit has broken or becomes occupied and the new minimum Hop and the number of Hops on the next unoccupied path changes, this information is sent by the IMP to its neighbors at the next 125 msec medium timeout.

3.2.4 Circuit Occupancy

A circuit is marked as either occupied or unoccupied. A six-bit counter for each output line is used to compute the state. The counter is initialized to Ø and decremented by 1 every 25 msec if greater than Ø. It is incremented by N for every placement on the output queue when the circuit is occupied and by M<N for every placement when the circuit is unoccupied. If the counter reaches 64, the circuit will be marked occupied and fast routing information will be sent at the next 125 msec timeout. The circuit will be marked as unoccupied when the counter returns to Ø. The quantities M & N will be determined experimentally or by simulation.

3.2.5 Metering

The maximum allowed traffic flow per destination through each IMP is regulated to change slowly. This provides for stability in the traffic flow. In addition, it allows the traffic flow to be adjusted to provide efficient traffic flow patterns. In the net, this scheme requires the use of two kinds of negative

acknowledgments: one for packets that were incorrectly received and one for packets that were correctly received but discarded. Incorrectly received packets will be retransmitted over the same output line, if possible. The information about discarded packets will be used to adjust the metering.

As of this writing, the details of the metering procedure have not been finalized. Various alternatives are being studied and simulated. However, the basis of the current scheme is as follows. Each IMP uses two counters C_i and D_i per destination in the metering of traffic. Counter C_i provides a time base and is incremented by a constant every 25 msec. This time base is also decremented by D_i for each packet queue placement to destination i. The counter D_i is used in conjunction with C_i to regulate the rate at which traffic is allowed onto each line for destination i. If $D_i > C_i$ traffic is inhibited by the IMP from flowing to that destination. Equivalently, a packet will not be placed on a queue if by so doing C_i would have to become negative. A count is kept of the number of negative acknowledgments returned during each half second interval per line and destination. D_i is incremented by a small constant every half second if the previous count was greater than zero; otherwise it is decremented by a small constant. The tentative limits on D and C are $0 \leq D < 512$ and $0 \leq C \leq 2D$. If all lines are marked occupied, the metering is ignored.

3.2.6 Loops

It is possible for the routing algorithm to occasionally generate one or more paths in a subnetwork that contain loops.

The algorithm will detect the presence of a loop by observing whether the value of the maximum Hop to any destination has reached its maximum value. When an IMP has learned that one of its output lines forms part of a loop, it will break the loop by removing the output line as a possible choice to reach the given destination.

The maximum Hop is computed as follows. Each IMP receives a report from each of its neighbors containing a maximum Hop number to each destination. This report is received at least every half-second and more frequently whenever changes are occurring. The largest value of the maximum Hop received on a line designated for output is incremented and becomes the current maximum Hop value to the given destination from that IMP. Five bits are allocated to the Hop number. If this value ever reaches 31, a loop is assumed to be present and the entry for the corresponding output line is removed from the table of output lines.

The other IMPs learn of this change by the rapid update routing that is sent every 125 msec. Whenever important changes occur, they too then remove the appropriate entries in their table of output lines.

3.3 Buffer Allocation

In this section, we describe the tasks to which buffers are to be allocated. Since no analytical procedures appear to be available for use in the selection of a buffer allocation

scheme, heuristics are used to improve the efficiency of system operation. The following heuristics seem appropriate to a good system design:

1) Not all the IMP's buffers should be allowed to reside on a single output queue. Each input and output line must always be able to get some non-zero share of the buffers. Consequently one buffer is allocated to each output queue to insure that output is always logically possible on each output line.

2) Eight buffers are allocated to handling input on the modem channels. This technique provides full double buffering for 4 or fewer lines. An IMP with 5 lines will reject an incoming packet only if 4 or more packets arrive "coincidentally".

3) Enough buffers should be provided so that all lines may operate at full capacity. A 3000 mi. circuit requires enough buffering to keep the line occupied during the 40 msec round-trip propagation time and to handle occasional surges and line errors. A short line probably needs no more than double buffering to achieve full capacity. A few additional buffers are useful to handle line errors and burst arrivals. The disadvantages appear to outweigh any advantages in allowing extremely large queues to form in an IMP. However, indirect store-and-forward lockup is made more probable with extremely short queues. In an

attempt to select a sensible value, the length of the sent queue + output queue is not allowed to exceed 8; an incoming packet will not be placed on an output queue whose length is greater than or equal to 4. This limitation may be adjusted in accordance with simulation and field test results. One can deploy a uniform allocation to each output line or, for simplicity of implementation, provide a statistically sufficient pool of buffers.

4) Four buffers are allocated to storing copies of single-packet messages from the Hosts, while the source IMP waits for reassembly space to be reserved at the destination IMP.

5) A small buffer is allocated to initiate input of a message from each of the real and fake Hosts. A regular packet buffer is set up to continue accepting input after the small buffer is filled, provided the end of the message has not yet occurred.

6) Ten buffers are allocated to reassembly storage. Additional buffers up to a maximum of 26 will be drawn from a buffer pool shared between reassembly and store-and-forward. All remaining buffers in excess of the minimum reservations are placed in a buffer pool from which they are retrieved on a first come, first served basis by both store-and-forward and reassembly.

Negative acknowledgments are useful in activating dormant buffers more quickly, but they add complexity. Although not

required for system operation, they are useful in achieving a more efficient buffer utilization as well as being useful for metering.

Fifty percent of all packets in the network are acknowledgments. It is possible to save both line capacity and program utilization by using an IMP/IMP serialized transmission strategy. A serial number is attached by the IMP to each transmitted packet out a given line and the receiving IMP expects to receive packets on that line with numbers in sequence. Each arriving packet is acknowledged by the receiving IMP by returning its sequence number to the transmitting IMP. Missing sequence numbers are negatively acknowledged as are packets that are correctly received by the receiving IMP but discarded.

Several acknowledgments can be delivered as a few bits at the beginning of a normal message, thus reducing the amount of acknowledgment traffic by a factor of about five. Furthermore, This scheme assures that no duplicates will be generated, except when a line breaks.

3.4 Overflow

In this section, we describe the use of the two overflow buffers in thwarting the occurrence of indirect store-and-forward lockups. Whenever traffic appears to have stalled in the regular net the IMPs will occasionally mark a packet (henceforth known as an overflow packet) for delivery via special buffers that constitute an overflow net.

An overflow packet is handled by the IMPs in such a way as to guarantee that, at any time, at least one such packet will reach its destination. Packets are delivered according to a randomized priority.

One buffer is used in each IMP to hold an overflow packet that is being transmitted, if any. A second buffer is provided in each IMP to store an overflow packet for subsequent transmission. The provision of two buffers, rather than only one, insures that an IMP will never reject an arriving overflow packet unless both buffers are occupied.

When an IMP has received no acknowledgments for its packets for several seconds, it selects one packet every 125 msec and marks it for delivery via the overflow net. Let us call this IMP an overflow source IMP. The overflow packet will pass from IMP to IMP using only the overflow buffers until it reaches the destination. The destination IMP will return an overflow acknowledgment to the overflow source IMP, if the packet is accepted by the destination IMP. Otherwise, the packet will simply be discarded by the destination IMP and no acknowledgment will be returned. At some later time, the packet will again be transmitted via the overflow net, unless it has already begun to move through the net in the normal fashion.

The overflow ACK also uses the IMP's overflow buffers in returning to the overflow source IMP. The overflow source IMP will discard the designated packet when the overflow ACK returns. If the packet is no longer in that IMP, the overflow ACK is simply discarded and the packet will eventually arrive at the destination as a duplicate packet and be discarded.

A packet is stamped with a unique random number by the IMP when it first delivers it into the overflow net. When an overflow packet is received at an IMP that already has both its overflow buffers filled, the arriving overflow packet will vie for the overflow buffer with its current occupant and the one with the lowest number will emerge victorious. The other packet will be simply discarded. Furthermore, a returning overflow ACK will always emerge victorious over an overflow packet, if they compete for the same overflow buffer.

To completely eliminate reassembly lockup, packets from the Host will also be inserted into the overflow net since the Host line can also be hung for a lack of buffer space.

The overflow network will always deliver the packet which has highest priority, barring phone line errors, and will also deliver any other overflow packets which don't encounter a higher priority packet. In this way, a small fraction of the circuit capacity and buffer space is thus devoted to guaranteeing that a small fraction of the Host traffic will always be able to reach its destination.

4. SUMMARY AND CONCLUSIONS

This study resulted in the identification of a number of problems in the design of the ARPA Network, the evaluation of the system performance, and the development of several algorithms to improve that performance. The study briefly considered the buffer requirements for including these algorithms in the current system design. A detailed evaluation of the storage requirements to code the algorithms is currently in progress.

The study identified congestion and lockup as two major system design problems. It concluded that the RFNM mechanism is not sufficient to prevent congestion or reassembly lockup with the limited supply of core memory currently available. An algorithm was developed to allocate buffers in reassembly so as to eliminate both of these conditions. The study also concluded that a new routing algorithm was needed to obtain high bandwidth between any pair of Hosts. An algorithm capable of achieving high bandwidth was developed and is currently being studied to understand its detailed performance characteristics. The advantages and disadvantages of limiting the length of queues to the minimum amount necessary was considered. It was concluded that no advantage was to be gained from allowing large queues to form in any IMP and an appropriate store-and-forward buffer allocation technique was developed for this purpose.

Four difficult performance areas were judged to be not of immediate importance and, therefore, were not resolved during the course of this study. These issues are listed below for completeness:

1) The effect of introducing deliberate delays on the Host line to insure reliable system operation under heavy load.

2) The extent to which the Host-to-Host protocol and the IMP/Host protocol should be made further inter-dependent.

3) The degree to which sharing of reassembly buffers among multiple Hosts at a given site will affect throughput.

4) The detailed performance and properties of the routing algorithm.