MEMORANDUM
RM-3578-PR
AUGUST 1964

ON DISTRIBUTED COMMUNICATIONS:
III. DETERMINATION OF PATH-LENGTHS IN A DISTRIBUTED NETWORK

J. W. Smith

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM

RM-3578-PR

AUGUST 1964

ON DISTRIBUTED COMMUNICATIONS:

III. DETERMINATION OF PATH-LENGTHS IN A DISTRIBUTED NETWORK

J. W. Smith

This research is sponsored by the United States Air Force under Project RAND—Contract No. AF 49(638)-700 monitored by the Directorate of Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force.

The RAND Corporation

1700 MAIN ST • SANTA MONICA • CALIFORNIA • 90406

PREFACE

This Memorandum is one in a series of eleven RAND Memoranda detailing the Distributed Adaptive Message Block Network, a proposed digital data communications system based on a distributed network concept, as presented in Vol. I in the series.* Various other items in the series deal with specific features of the concept, results of experimental modelings, engineering design considerations, and background and future implications.

The series, entitled *On Distributed Communications*, is a part of The RAND Corporation's continuing program of research under U.S. Air Force Project RAND, and is related to research in the field of command and control and in governmental and military planning and policy making.

The present Memorandum, the third in the series, is a continuation of the model simulation study reported in the previous volume. Since a network of the type proposed had never been built, there was much we did not know about its performance. For example, we wanted to learn more about the distribution of message path-lengths in the network, in order that transmission times might be determined; we wanted to know how the network behaved under heavy loading--such as would occur during a crisis;

---

*A list of all items in the series is found at the end of the Memorandum.

we wanted to know how the network would react when a "hog" station or stations attempted to purposely overload the network; we wanted to know how many message blocks would be lost if a policy of dropping traffic that has circulated longer than some specified time were used.

Because of the complexity of such networks, our only high-confidence tool for predicting performance was the Monte Carlo simulation, wherein messages are created and circulated in a computer model of the network, statistics of traffic-flow are examined, network parameters are changed, and the network is re-examined.

A FORTRAN simulation (for the IBM 7090 computer) was described in Vol. II in the series, together with the results of the simulation performed. There were three shortcomings to this effort: first, the size network that could be accommodated was smaller than desired; second, an undue number of messages was being lost during heavy overload conditions when messages began to take circuitous routes; and third, when the routing doctrine was improved to prevent such message losses, it became economically unfeasible to run the computer simulation long enough to determine the number of messages which could be expected to be lost. A target of less than one lost message per 100,000,000 was sought; this extreme requirement was selected because it was felt that future systems should be able to transmit digital data between computers--and computers are often intolerant of errors.

A new simulator, designed to resolve these problems, was encoded in the SCAT language.* This Memorandum describes that simulator and its appurtenances, and reports on the successful rectification of the previous effort's shortcomings. In addition, some analytical investigations of traffic-flow are described and evaluated.

---

*International Business Machines Corporation, SHARE SOS Reference Manual--SHARE Operating System for the IBM 709, IBM Applied Programming Publication, New York, 1960-61.

SUMMARY

Results of investigations into the behavior of distributed communications networks under various loading conditions are reported. A mathematical model and a deterministic equation for predicting the distribution of message path-lengths are derived and evaluated. A SCAT-encoded* simulation program that corrects deficiencies of earlier simulations is described.

An "input-choking" doctrine, together with a short, purposeful delay of messages passing through each station (when necessary), proved to be a powerful device in preventing loss of messages within networks operating at high loading ratios. The decrease in delay and in message-flow rate caused by the doctrine was negligible.

For the networks studied, a policy of dropping messages that have traversed paths greater than twice the longest possible path between the extremities of the network, resulted in a message dropout rate of less than one in 100,000,000 when the networks were operating at normal, and even higher than normal, loadings. At low loadings there were even fewer messages dropped.

---

*<u>Ibid</u>.

CONTENTS

SYMBOLS

SYMBOL		INTRODUCED ON PAGE
S_i	The i^{th} station in a network, in an N x M array of stations.	1
$H_i(j,k)$	The expected number of stations to be traversed by a message originating at S_i, transmitted via link j of S_i, before being delivered to S_k.	3
$L_{j,i}$	Link j of station S_i; links are numbered from 0 through 7, and are displayed (abstractly) clockwise around the station with $L_{0,i}$ at high noon.	3
S_o	A message's station of origin.	4
S_d	A message's addressee.	4
h	The hand-over number associated with a message; i.e., the number of stations traversed by a message in its wanderings.	4
HMAX	The maximum allowable number of traverses a message may take before being dropped from circulation.	4
B(i,k)	By definition, the length of the best path between S_i and S_k.	5
N_x	The number of best paths of length x.	5
x'	The maximum best-path length.	5
B	The best-path distribution:	5

$$B(x) = N_x / \sum_{j=1}^{x'} N_y .$$

SYMBOL		INTRODUCED ON PAGE
λ	The number of links in a network.	8
α	Message-loading factor for a simulation.	8
π	The probability that a link is impaired (unavailable or busy).	9
ML	Message-unit length.	9

FIGURES

I. INTRODUCTION

The distributed communications network examined in this Memorandum is a collection of communication stations interconnected in a more or less regularized manner and employing a common switching doctrine. The connectivity is such that the number of bi-directional lines (or uni-directional links) between a station and its neighbors is constant over the interior of the network; lengths and bandwidth capabilities of the lines may vary over the network. Switching, or routing, is performed by choosing a "best" initial link toward a desired addressee, rather than by attempting to choose a "best" overall path.

Certain stylized connectivities may be introduced to define the redundancy level of such networks--the concept is illustrated in Fig. 1.* It is useful and instructive to deal with such connectivities, and we shall do so. However, many of the observations reported in this Memorandum are dependent on a constancy of connectivity and routing doctrine rather than on any specific regularity of connectivity.

The choice of a "best" initial link is effected by adjoining to each station, S_i, a matrix, H_i, which assigns

---

*Figure 1 is taken from ODC-I, where the concept of redundancy levels is fully described; ODC is an abbreviation of the series title, On Distributed Communications. The number following refers to the particular volume within the series.

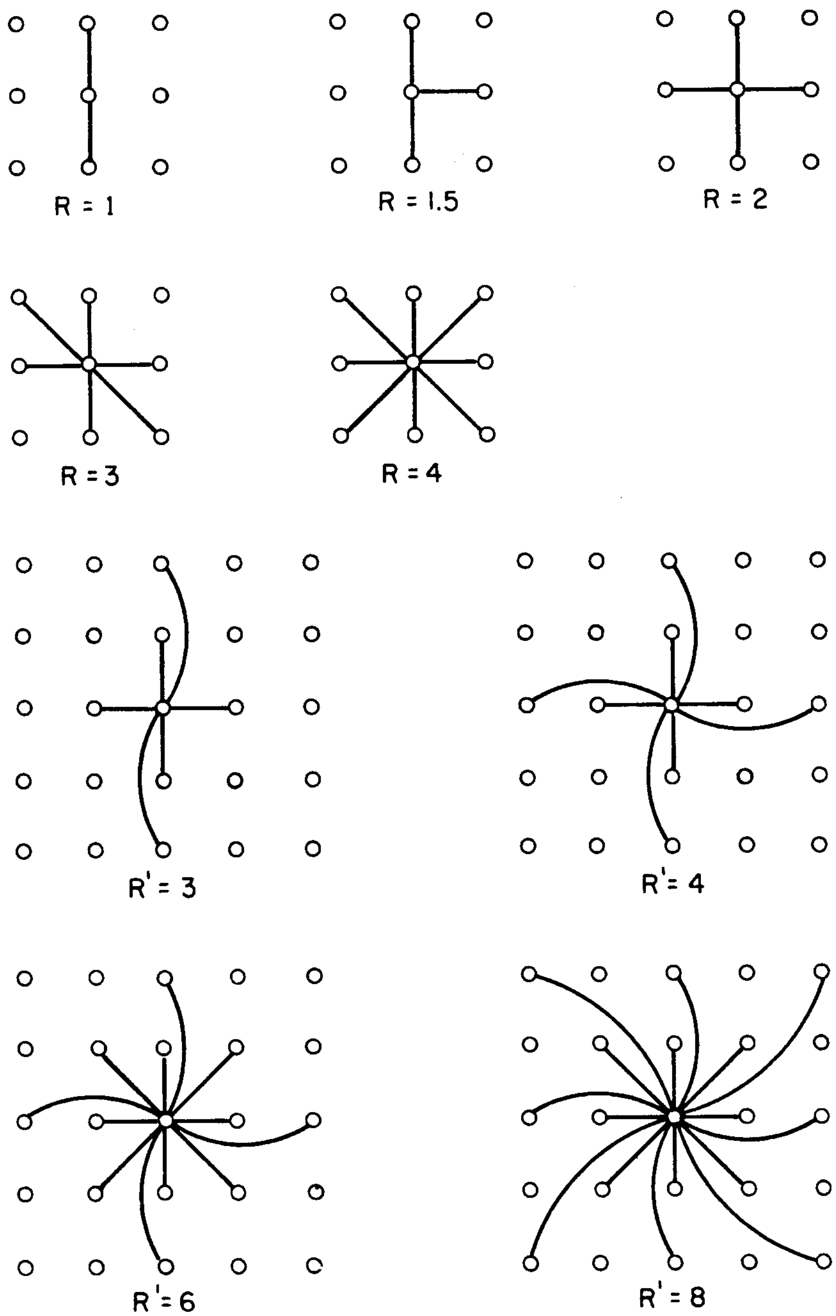


Fig. 1 Definition of Redundancy Level

to each link, $L_{j,i}$, of station S_i a measure, $H_i(j,k)$, of link j's merit as an initial link along which to route messages destined for station S_k--for all stations, S_k, in the network. The figure of merit used in this Memorandum is the expected number of stations to be traversed before delivery of the message. That is, a value, $K = H_i(j,k)$, states that station S_i expects a message destined for station S_k that is sent out along link $L_{j,i}$ to traverse K stations before delivery. By adjoining to every message an integer, h--the hand-over number, which is incremented by unity each time the message is retransmitted by a station--the values of H_i may be continuously updated as a function of the hand-over numbers associated with incoming messages and of the current values of H_i. Messages are routed by applying a decision procedure to those H_i values pertinent to the messages' destinations.

The transient behavior of the H_i, and of message-flow as the H_i change under suitable updating algorithms, is discussed in ODC-II. The present Memorandum is concerned with "steady-state" flow, wherein the H_i have assumed their "best" values and there remain fixed.

II. THE ROUTING DOCTRINE

Let S_o, S_d, and h refer to a message's originating station, addressee, and hand-over number, respectively. Messages arriving at station S_i via link $L_{j,i}$ cause the values of H_i to be updated by the following updating doctrine (U1):

(U1) $$H_i(j,o) = \text{minimum}(h, H_i(j,o)).$$

Let station S_i have links $L_{1,i}$, $L_{2,i}, \ldots$, $L_{n_i,i}$ and let HMAX, the maximimum allowable hand-over number, be a fixed parameter. Station S_i retransmits messages by applying the following routing doctrine (R1):

(R1)
a) if $h \geq$ HMAX, the message is dropped;

b) if all links are in use or otherwise unavailable, the message is stacked until some link is freed;

c) if links are available, h is incremented by unity and the message retransmitted over that link $L_{j',i}$ for which

$$H_i(j',d) = \text{minimum}(H_i(j,d)), \text{ for all available links, } L_{j,i}.$$

In practice, the H_i are originally set equal to HMAX; as messages move through the network, these values eventually assume their absolute minima. These minimum values may be calculated a priori by a best-path* algorithm.

---

*See Sec. VI.

Assume this has been done, resulting in values $\overline{H}_i$. Define, for all stations S_i and S_k, $k \neq i$:

$$B(i,k) = \text{minimum}(\overline{H}_i(j,k)), \text{ for all links } L_{j,i} \text{ of station } S_i; \qquad (2.1)$$

$$N_x = \text{the number of } B(i,k) \text{ equal to } x \text{ for all } i,k; \qquad (2.2)$$

$$x' = \text{maximum}(B(i,k)), \text{ for all } i,k. \qquad (2.3)$$

That is, $B(i,k)$ is the length of the best path from S_i to S_k, N_x is the number of such best paths of length x, and x' is the length of the longest best path. The distribution, B, defined by

$$B(x) = N_x \Big/ \sum_{y=1}^{x'} N_y \qquad (2.4)$$

is called the <u>best-path distribution</u> for the network. It will be shown later that B is of great utility in predicting traffic-flow in steady-state networks. If messages are generated by choosing origins and destinations randomly from a uniform distribution of stations, and if each message is completely routed through the network before the next message is originated, then the resulting distribution of message path-lengths is clearly B. That

B is equally valid for low-load conditions is shown by Fig. 3,* to be discussed later.

The HMAX test in routing doctrine R1 is essential. Traffic will, in practice, be cut into message blocks of fixed length and be sent piecemeal (but serially). If the differential time delay is excessive, then the order of arrival of these message blocks could fall out of sequence. This differential time delay is equivalent to the difference in hand-over numbers of the arriving message blocks. Therefore, the lower the tolerable maximum hand-over number, the higher the overall data rate between two network users will be. We also chose to drop message blocks whose hand-over number exceeds this maximum allowable, to insure flushing out message blocks addressed to nonexisting stations. Since fixing the length of message blocks implies fixing station processing time (for routing), network time may be equated to station traverses by using an average link-length for the network, and temporal decisions can then be made on the basis of some maximum hand-over number and the hand-over number associated with messages. Thus, the problem of choosing an HMAX small enough to maintain "integrity" of communications, yet large enough to guard against excessive message dropout is a central one. The choice

---

*See p. 14.

of such a value for a given network depends on a knowledge of the distribution of message path-lengths of delivered messages under varying traffic densities and using an unbounded HMAX. Three methods for determining such distributions--Monte Carlo simulation, mathematical modeling, and approximate calculation--are described and compared in the next section.

III. COMPUTATIONAL TECHNIQUES

MONTE CARLO SIMULATION

The simulator is described in detail in the program listing of Appendix B; briefly, it operates in the following manner:

1) a network is defined in terms of its size, configuration, and connectivity;

2) link-lengths are assigned--as transmission times--by being drawn from a uniform distribution bounded by the parameters TPMAX and TPMIN;

3) the hand-over number tables, H_i, are preset to either $\overline{H}_i$ or to values bounded by HMAX and HPRIME;

4) a fixed number, $(\alpha)\cdot(\lambda)$, of messages is introduced into the network, origins and destinations of messages being drawn from a uniform distribution of station numbers*--λ is the total number of links (twice the number of lines) in the network and α is a loading factor;

5) the routing and transmission of messages through the network is directly simulated by applying the routing doctrine R1;

6) delivery or dropping of a message results in the insertion of a new "random" message into the network, thus maintaining loading;

---

*This defines "uniform" loading.

7) new messages may be "choked" by applying R1 first to enroute messages, then to new messages (which are, in effect, kept on a special stack).

The parameter α is a measure of the activity of the network. We may assume that under uniform loading, traffic density can be equated with a probability, π, that a link is unavailable or busy, and may then relate π to activity by

$$\pi \cong \left[\frac{ML}{\frac{TPMAX-TPMIN}{2} + 2(ML)} \right] \cdot (\alpha) \qquad (3.1)$$

where ML is the processing time per message-unit (i.e., message-unit length). (TPMAX-TPMIN)/2 is the average line-length (or average transmission time), and a message-unit requires ML time units to be inserted into or withdrawn from a link. In (3.1), α is treated as the number of message-units that may be on a link (either in transit or being transmitted into the link), and the first factor is interpreted as the probability that a message is being transmitted into a link (thus making the link unavailable).

Under uniform loading in a fully loaded network, at each time period all links would be jammed and all stations would be simultaneously accepting and transmitting a message. The average number of messages in a fully loaded network would, therefore, be equal to

$$(\lambda)\ \left(\frac{TPMAX-TPMIN}{2(ML)} + 2 \right) = N\ .$$

The loading ratio of a network is then equal to $N/\alpha\lambda$, which is equal to π; thus, π may be equated to the loading ratio, or traffic density, of a network.

MODEL A (M_a)

A model of network behavior is abstracted by considering the four possibilities that exist for the disposition of messages at any station (disregarding the possibility of dropout). A message being routed at a station has associated with it that station's prediction of the best-path length to the message's addressee. The message may either

1) remain at the station, or

2) be retransmitted to a neighbor whose best-path prediction is

   a) one (1) less than the previous station's prediction,

   b) the same as the previous,

   c) one (1) greater than the previous.

That is, links may be characterized as being "best," "next best," or "worst" for any message. The model defined in Fig. 2 assumes:

1) a uniform distribution of links with respect to "best," "next best," and "worst" overall addressees;

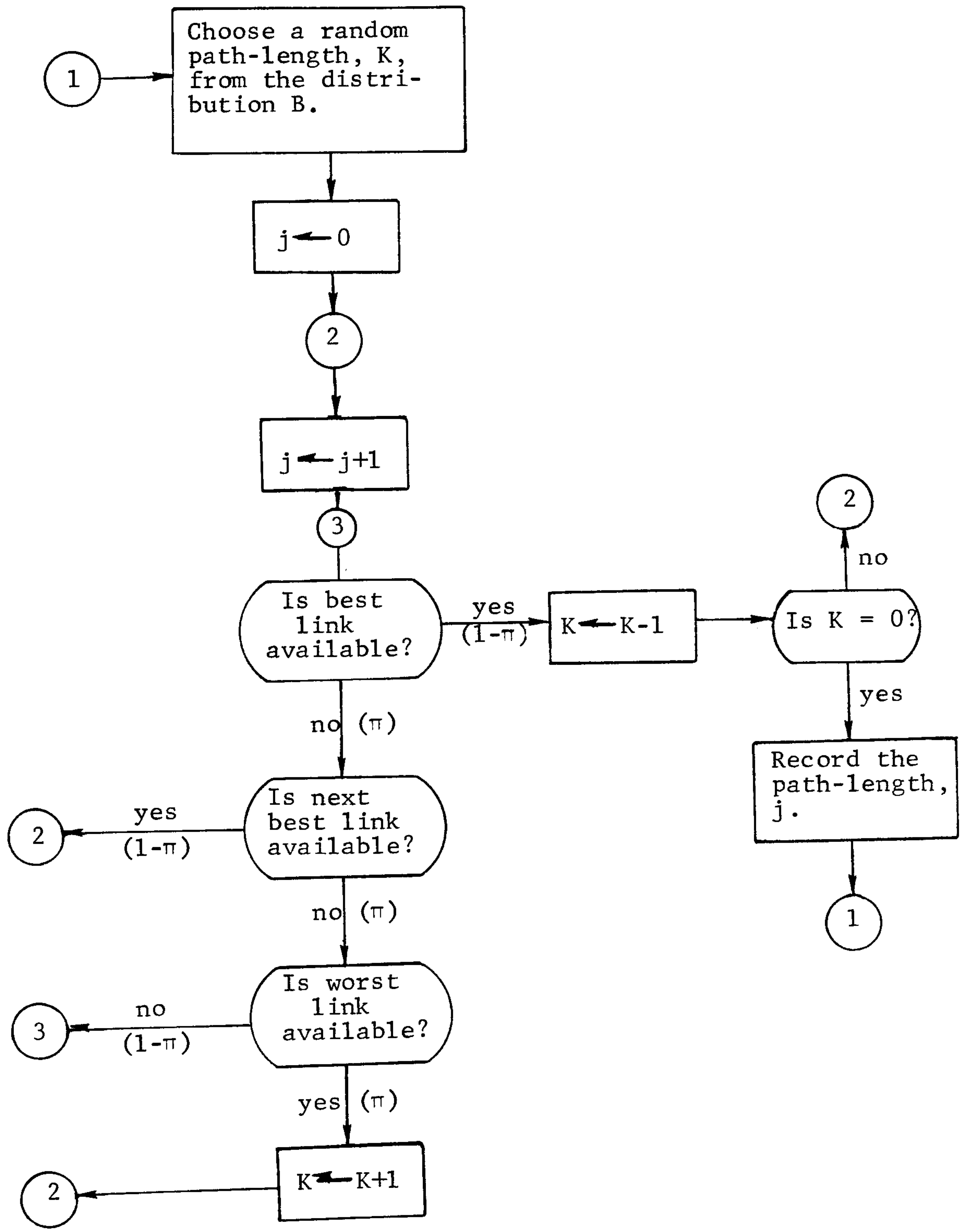


Fig. 2 Flowchart of Model A

2) a uniform probability, π, of link nonavailability;

3) uniform loading;

4) stations connected to neighbors only.

The first assumption is the weakest of the four; however, within the interior of "reasonably" connected networks satisfying the last assumption, it is "reasonably" valid.

MODEL B (M_b)

Model B is obtained from the previous one by aggregating "next best" and "worst" links. In M_b, therefore, a message at a station whose best-path prediction for the message is, say, n, will have a probability p(n,x) of being delivered in x traverses, given by

$$p(n,x) = g(n,x)\ (\pi^{x-n})\ (1 - \pi)^n \ , \tag{3.2}$$

where π is the probability of link nonavailability, and g(n,x) is the number of combinations of "best" paths (chosen with probability $1 - \pi$) and "worst" paths (chosen with probability π) ending in a "best" path. That is:

$$\begin{aligned} g(n,x) &= \begin{bmatrix} x-1 \\ n-1 \end{bmatrix} \\ &= g(n,x-1) + g(n-1,x-1). \end{aligned} \tag{3.3}$$

The distribution, $M(\pi)$, generated by this model is given by:

$$M(x;\pi) = \sum_{i=1}^{x'} B(i)\cdot P(i,x) \quad . \tag{3.4}$$

Letting $\pi = 1/a$, and using (3.2), we obtain:

$$M(x;1/a) = \sum_{i=1}^{x'} B(i)\cdot(a-1)^{i}\cdot(a)^{-x}\cdot g(i,x) \quad . \tag{3.5}$$

Since $\sum_{x=1}^{\infty} g(i,x)\cdot(a)^{-x} = (a-1)^{-i}$, we have:

$$\sum_{x=1}^{\infty} M(x;1/a) = \sum_{i=1}^{x'} B(i) = 1 \quad ; \tag{3.6}$$

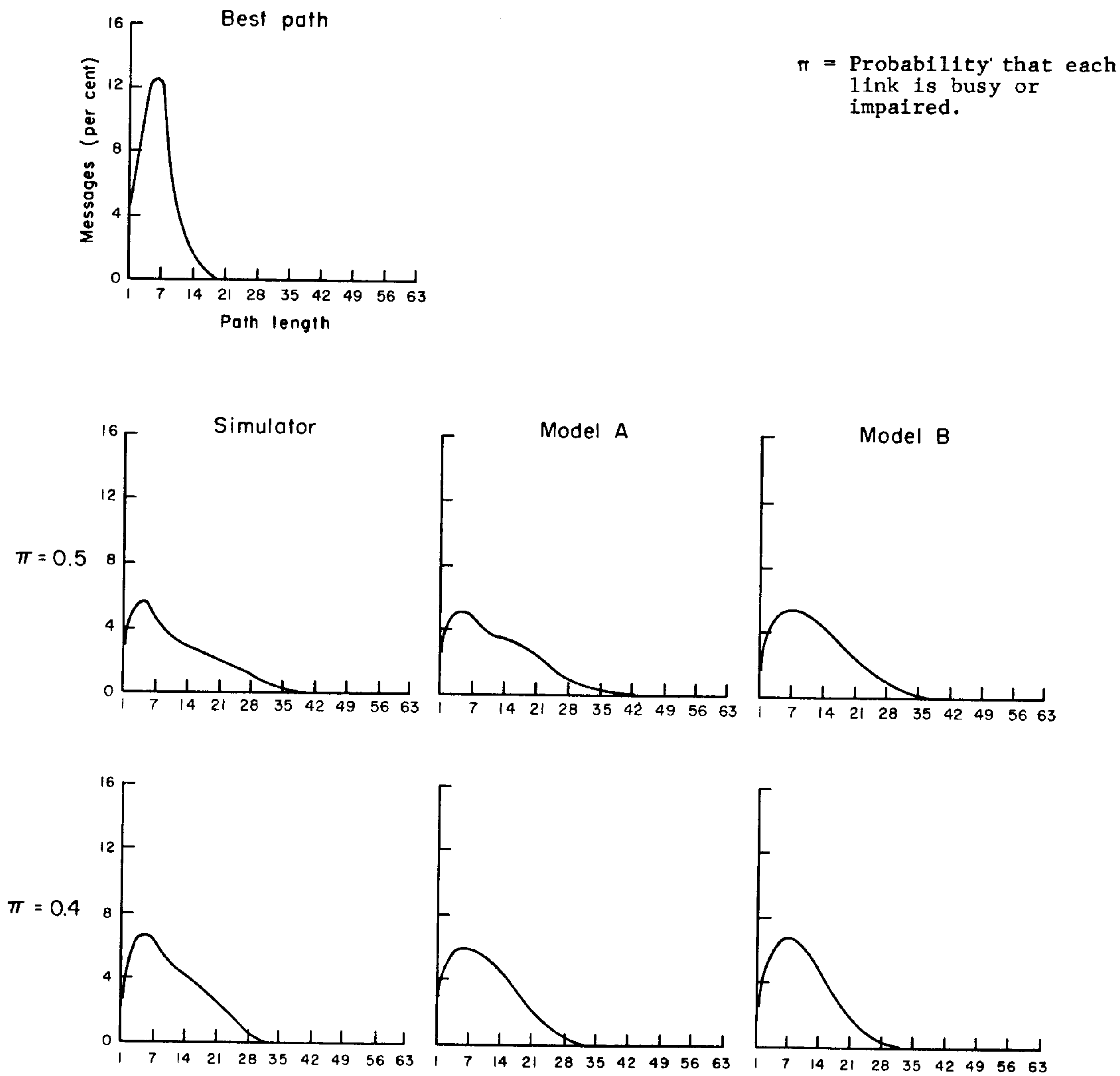
and for the "tail" of the distribution from $x'+k$ on:

$$\sum_{x=x'+k}^{\infty} M(x;1/a) = \sum_{i=1}^{x'} B(i)\cdot(a-1)^{i}\left(\sum_{x=x'+k}^{\infty} g(i,x)\cdot(a)^{-x}\right)$$

$$< \sum_{i=1}^{x'} B(i)\cdot(a-1)^{i-x'-k} \quad . \tag{3.7}$$

COMPARISON OF RESULTS

The three techniques were applied to networks of size 10x10 and 14x7, using redundancy levels of 2, 3, and 4. Figure 3 exhibits the results of the 14x7, redundancy-three case--all other runs produced the same behavior; Fig. 4 tabulates statistics for the various runs.


Best path
Messages (per cent)
Path length
16
12
8
4
0
1 7 14 21 28 35 42 49 56 63
π = Probability that each link is busy or impaired.
Simulator
Model A
Model B
π = 0.5
π = 0.4

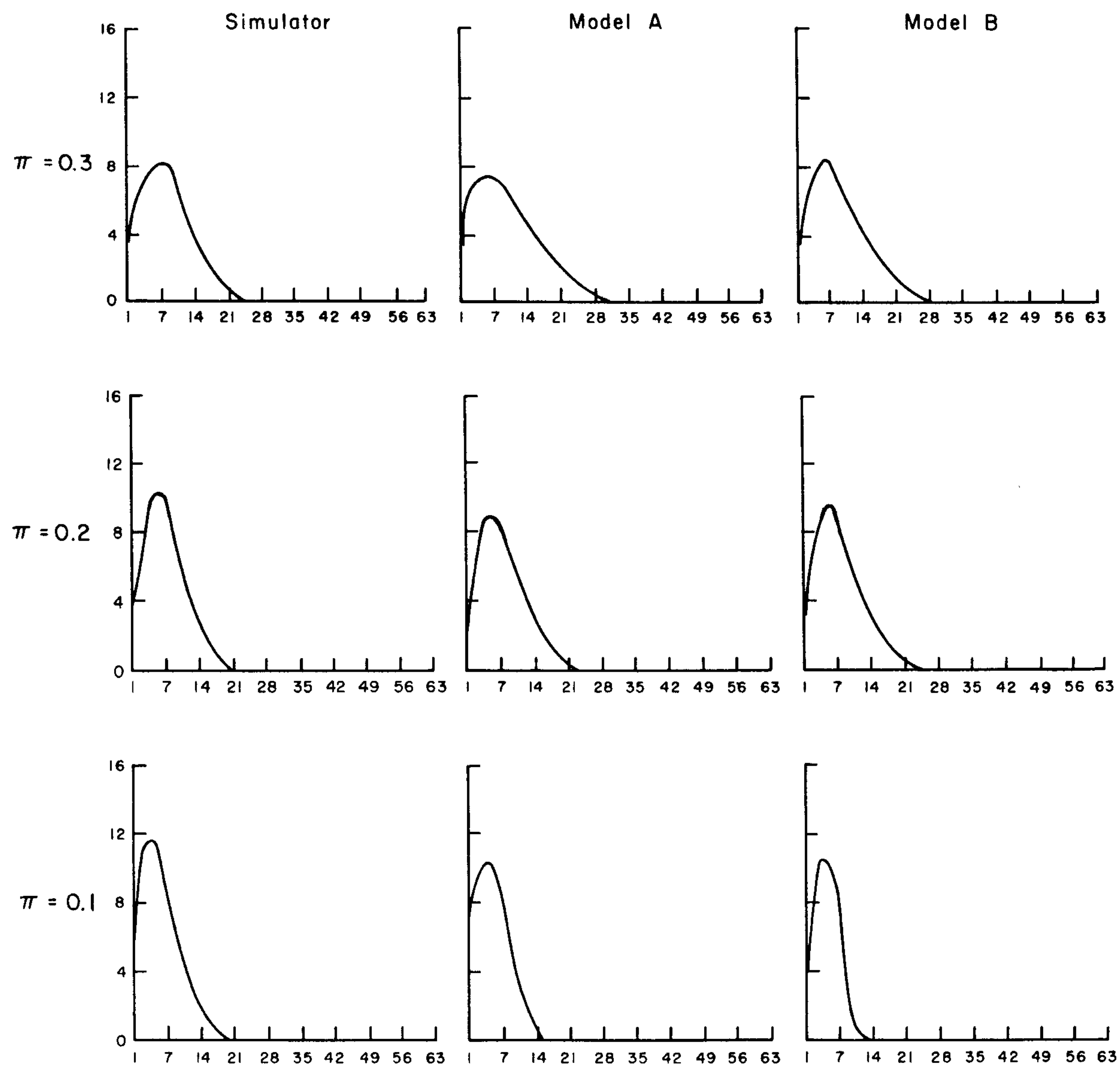


Fig. 3 Distribution of Path-Lengths of Delivered Messages: 14 x 7 Network; R = 3

Net Description	Best-Path Average	Best-Path Longest	π	Average Path-Length S_1	Average Path-Length M_a	Average Path-Length M_b	HMAX Required to Reduce Dropouts to One in 10^8. S_1	HMAX M_a	HMAX M_b
14 x 7 R = 4	5.21	13	.5	11.17	12.16	10.40	61	>63	62
			.4	8.31	9.72	8.67	48	50	49
			.3	6.58	7.95	7.43	25	36	39
			.2	5.82	6.74	6.51	19	28	31
			.1	5.43	5.84	5.78	16	22	25
14 x 7 R = 3	6.11	19	.5	14.36	14.23	12.19	>63	>63	72
			.4	10.56	11.40	10.16	55	60	57
			.3	8.14	9.28	8.71	33	42	46
			.2	7.11	7.86	7.62	26	34	38
			.1	6.43	6.88	6.77	21	24	30
14 x 7 R = 2	7.00	19	.5	17.19	16.24	13.97	>63	>63	74
			.4	15.30	13.05	11.64	>63	63	59
			.3	11.25	10.63	9.98	59	43	47
			.2	8.95	9.04	8.74	42	35	38
			.1	7.66	7.85	7.76	27	26	31
10 x 10 R = 3	5.67	18	.5	12.32	13.23	11.32	>63	>63	70
			.4	9.39	10.50	9.43	51	60	55
			.3	7.53	8.64	8.09	29	37	44
			.2	6.50	7.33	7.07	21	33	36
			.1	5.98	6.37	6.29	19	25	29

S_1 - Simulation M_a - Model A M_b - Model B

Fig. 4 Statistics for Networks

Relation (3.1) was used to equate simulator loading with the probability of link impairment, π, used by the models. Since this is an approximate relation, no exact comparisons between simulation and modeling can be made. Nevertheless, it is clear that both models produce distributions which are commensurate with those produced by the simulator, Model A tending to reproduce the shape of the simulator distributions more faithfully than Model B. Moreover, note that both models tend to produce distribution "tails" which are generally pessimistic. The same comparisons held for the other runs. We therefore conjecture that both models may be used to predict approximate behavior of networks of the type examined, and that (3.7) may be used to obtain approximate estimates of HMAX's for desired dropout rates.

Figure 4 indicates that the transition from a redundancy level of three to a level of four results in diminishing returns. For redundancy levels greater than two, it appears that an HMAX equal to twice the longest best-path is sufficient to reduce dropouts to the "noise" level under normal, and even higher than normal, loadings ($\leq$ 30%).

IV. INPUT-CHOKING AND STACK LENGTHS

Input-choking* in networks has two important consequences. First, since a link can be usurped by a message for no greater period of time than it takes to insert a message into a link (ML), it is clear that stack length can never exceed the number of links associated with a station; that is, stations require only one "word" of stack storage per link. Moreover, choking tends to smooth out potential activity peaks, particularly in the vicinity of "hog" stations and around the center of the network. Removal of input-choking simply means that new messages are treated as enroute messages. Under such conditions, stack lengths cannot be contained. Since stack storage must be finite, a new message-dropping criterion must be added to the routing doctrine, R1:

(R1) d) if no links are available and the stack is full, the message is dropped.

Thus, the choice of a fixed stack-storage length, STACK, large enough to reduce stack dropouts to the noise level, becomes as important as a choice of HMAX. Since stack length is a highly local phenomenon--strongly dependent on traffic distributions--there seemed to be no simple way of determining STACK without direct simulation. Accordingly, the simulator was applied--with no input-choking--to a 14x7 net of redundancy-three, with maximum stack

---

*See p. 9.

lengths of 6, 9, and 12. The distributions obtained were almost identical to the "choked" distributions. They are not reproduced here; however, Fig. 5 compares the pertinent simulations, and leads us to the following conjectures:

1) dropouts will occur under no-choke conditions; to insure a low dropout rate, STACK will have to be unconscionably large--hence, unfeasible;

2) under uniform loading, no-choking produces a small, uniform increase in loading;

3) hence, there seems to be no justification for adopting a no-choking doctrine.

These conjectures are fortified by a consideration of the distribution of total time in stacks (including input-waiting time) for delivered messages. These distributions were identical for the choke and no-choke cases. In fact, the distributions shown in Fig. 6 were typical of all networks simulated; they show that the probability of excessive delays-in-stack is exceedingly small, even under high loading.

Probability of Link Impairment, π	HO Average	HO Variance	Mode Value of HO No.	Choke?	Stack Drops	No. in Stack
.1	6.5 6.5 6.5	13.2 13.5 13.5	4 4 4	yes no no	0 0	12 9
.3	8.0 8.1 8.1	20.6 21.7 21.7	5 5 5	yes no no	0 0	12 9
.5	13.7 14.3 14.2	104 117 114	3-7 3-7 3-7	yes no no	5 72	12 9
.7	17.9 17.4 17.9	166 167 176	3-7 3-7 3-7	yes no no	27 251	12 9

Fig. 5 No-Choking Statistics

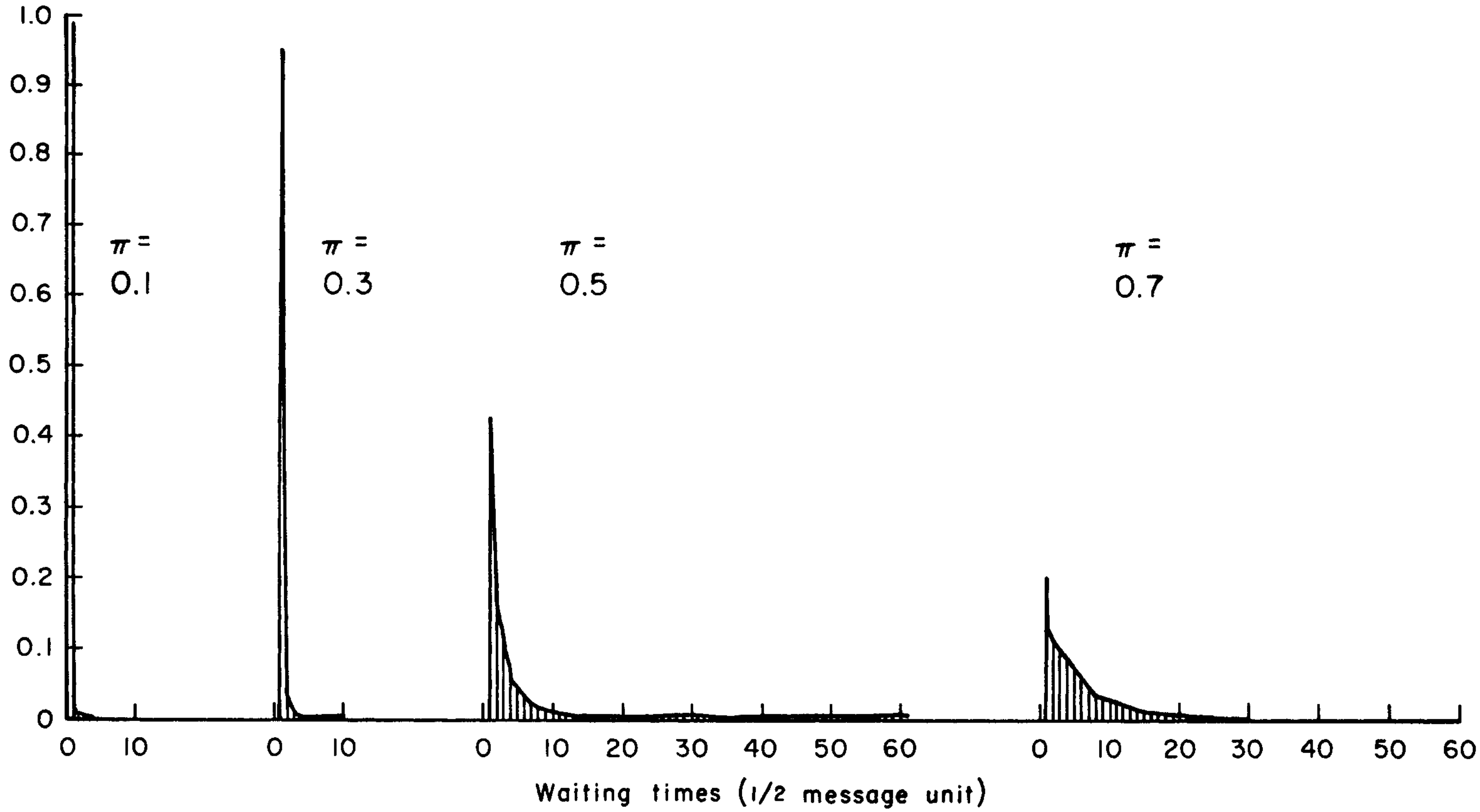


Fig. 6 Distribution of Stack Waiting Times of Delivered Messages
(14 x 7 Network; Redundancy 3; Choking)

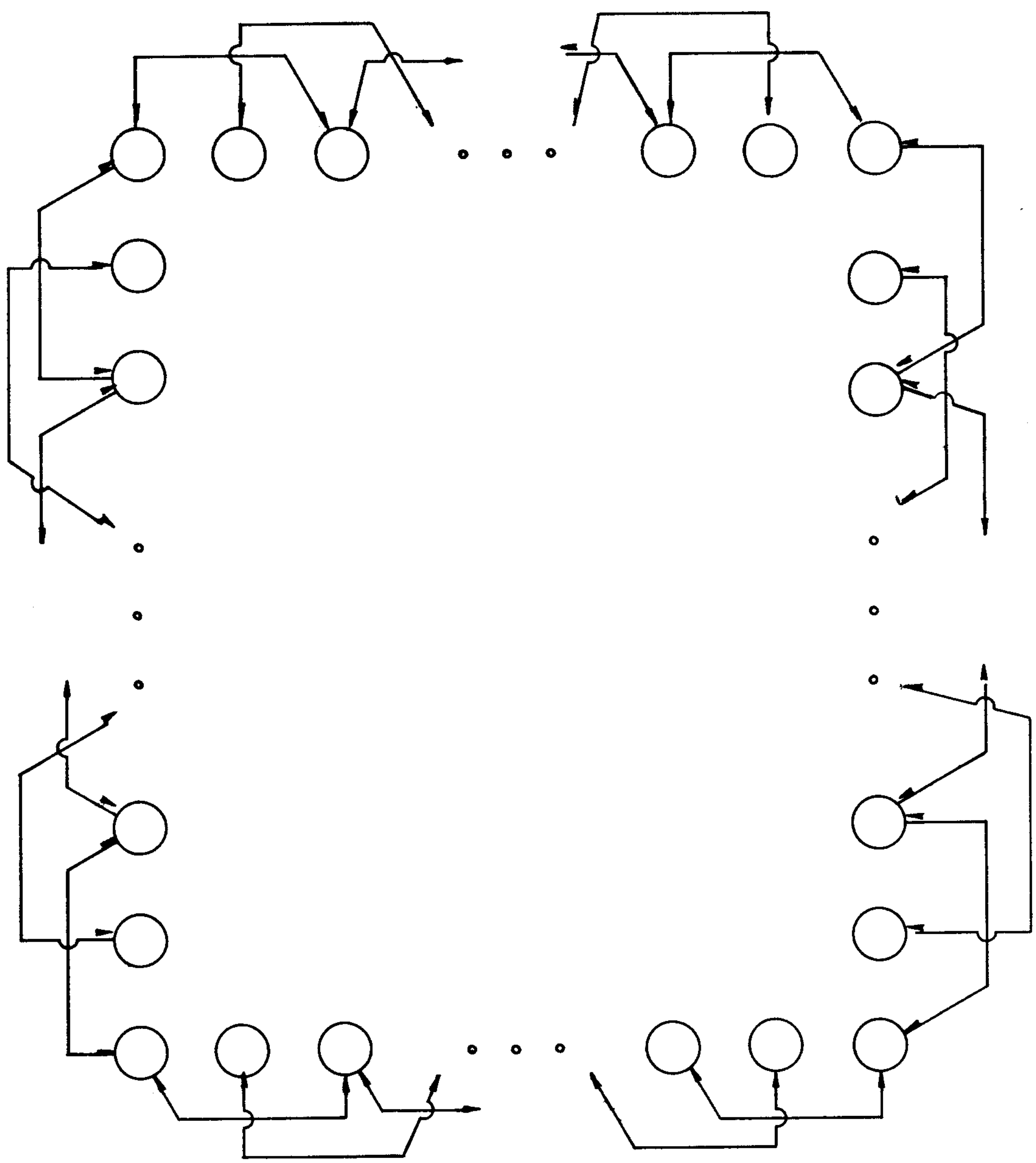


Fig. 7 Edge Binding

V. EDGE BINDING

In any network, much of the routing power of peripheral stations is wasted simply because peripheral links are unused. Thus, messages tend to reflect off the boundary into the interior or to move parallel to the periphery. Providing alternate paths by edge binding, as illustrated in Fig. 7, should tend to shorten average path-lengths measurably. Figure 8 exhibits flow patterns produced by simulations of a 14x7 network of redundancy-three with and without edge binding. Each diagram is a spatial representation of the network, the entry in position (i,j) of the representation indicating the number of messages routed through the corresponding station since the start of the simulation. The effect of edge binding in reducing interior clustering is clear. Figure 10 exhibits the distributions for the edge-bound net, while Fig. 9 gives statistics for the various runs. Although edge binding reduces clustering and results in an increase in flow-rate, it seems clear that the resultant distributions are less desirable than those obtained without edge binding. Apparently edge binding tends to overly penalize paths through the center of the network, and seems to provide higher flow rates at the expense of slightly higher drop-out rates.

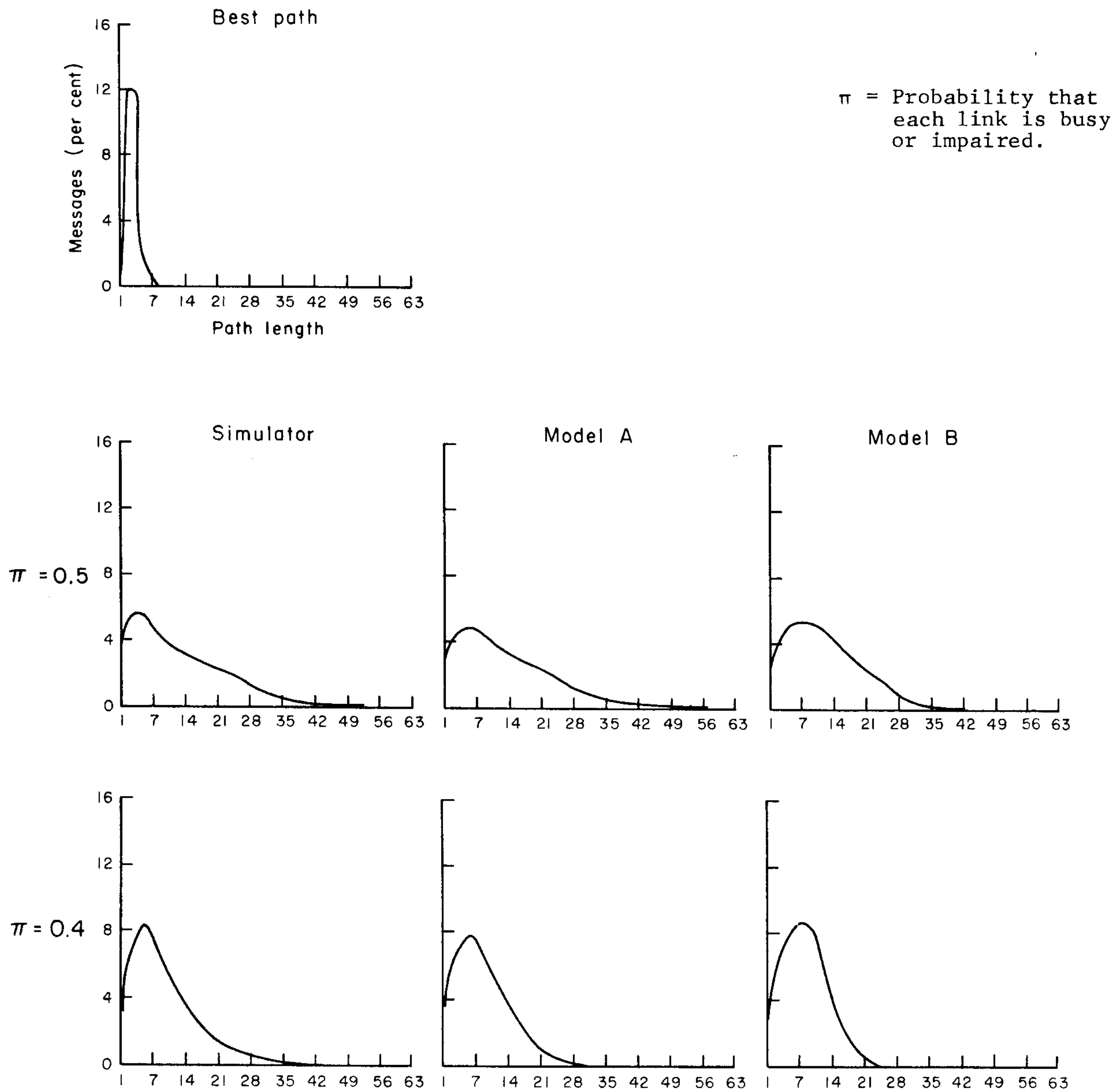
	UNBOUND							BOUND						
	219	408	514	601	566	525	413	872	919	1893	2452	4131	3715	4641
	506	941	1209	1419	1510	1689	1375	1664	1326	1344	2097	3277	4859	4742
	759	1397	1866	2403	3110	3582	2788	2942	2238	1787	2408	3763	5125	5790
	1045	2043	2914	4066	4773	5128	3567	4135	3194	2714	3216	4284	5365	5862
	1476	3114	4399	5263	5579	5687	3860	4753	3971	3451	3831	4615	5477	5836
	2256	4481	5328	5702	5858	5876	3890	5273	4472	4031	4260	4861	5454	5827
	3036	5326	5705	5875	5920	5868	3819	5477	4784	4234	4471	4916	5352	5777
$\pi=0.5$	3513	5656	5824	5855	5864	5695	3575	5548	4807	4322	4429	4749	5296	5757
	3699	5678	5798	5770	5653	5286	3071	5478	4826	4231	4165	4489	5066	5671
	3689	5620	5634	5476	5089	4191	2109	5498	4592	3719	3633	3886	4732	5487
	3568	5282	5145	4613	3710	2657	1251	5312	4208	3156	2855	3098	4000	5111
	3002	4114	3641	2923	2161	1561	799	5168	3732	2464	2045	2051	2972	4077
	1732	2072	1796	1457	1223	1028	551	4022	3279	1859	1428	1203	1768	2434
	493	579	645	627	567	477	249	3747	2240	2701	1442	1444	880	1270
	287	512	594	638	597	490	330	567	616	1179	1216	2109	1763	3138
	658	1122	1300	1434	1404	1296	760	889	776	842	1124	1328	2402	3332
	937	1621	1939	2206	2198	1929	1165	1512	1125	1169	1523	1795	2710	4578
	1176	2096	2554	2933	2895	2538	1500	2448	1592	1674	1974	2113	3195	4796
	1444	2542	3070	3466	3374	2995	1791	3118	2210	2013	2239	2410	3472	5124
	1664	2893	3491	3833	3701	3353	2037	3934	2500	2203	2425	2663	3683	5153
$\pi=0.3$	1842	3224	3782	4054	3932	3553	2120	4170	2850	2325	2454	2670	3669	5109
	2028	3484	3889	4072	3950	3471	1969	4521	2961	2387	2466	2599	3537	4883
	2129	3442	3717	3893	3718	3099	1718	4521	3084	2327	2413	2453	3168	4631
	2002	3153	3448	3518	3292	2659	1462	4558	2857	2155	2137	2143	2742	3952
	1678	2689	2942	2999	2731	2160	1160	4212	2531	1849	1933	1732	2015	3129
	1284	2018	2277	2262	2070	1665	904	3876	2040	1494	1424	1213	1345	2064
	827	1350	1477	1445	1357	1111	631	2725	1873	1180	1042	836	830	1149
	424	542	636	646	593	489	273	2655	1492	1910	1171	1149	634	656
	116	205	244	239	233	183	115	251	279	555	484	891	531	1255
	250	466	542	568	538	480	274	355	334	344	432	431	579	966
	371	665	808	819	790	687	404	624	472	511	625	599	618	1606
	430	800	932	1007	1006	874	509	924	548	738	751	714	690	1637
	516	967	1081	1189	1151	943	591	1122	673	752	860	797	760	1994
	556	988	1169	1298	1216	998	627	1452	675	804	867	832	785	1872
	591	1058	1273	1307	1281	1052	643	1515	701	791	918	861	748	1998
$\pi=0.1$	598	1066	1274	1344	1280	1072	599	1755	666	817	885	830	794	1748
	570	1030	1179	1276	1183	977	577	1571	717	819	886	827	675	1618
	533	959	1089	1150	1086	938	511	1707	691	819	869	781	685	1297
	510	826	949	994	981	757	441	1362	629	723	772	702	602	1028
	415	637	780	810	776	604	346	1423	583	602	606	570	500	646
	287	483	520	520	522	387	261	782	527	407	417	355	323	419
	142	177	216	214	224	180	113	1089	487	805	462	520	269	251

Fig. 8 Traffic Flow Distribution
(Number of Messages Routed Through Node in 7 x 14 Network)

Net Description	Best-Path Average	Best-Path Longest	π	Average Path-Length S_1	Average Path-Length M_a	Average Path-Length M_b	HMAX Required to Reduce Dropouts to One in 10^8. S_1	HMAX M_a	HMAX M_b
14 x 7 R = 4	4.32	10	.5	9.98	10.10	8.61	>63	61	52
			.4	7.77	7.98	7.18	59	44	40
			.3	6.29	6.58	6.16	38	36	32
			.2	5.35	5.58	5.39	23	23	25
			.1	4.78	4.87	4.79	15	16	20
14 x 7 R = 3	4.86	10	.5	12.10	11.31	9.70	>63	>63	55
			.4	9.85	9.02	8.09	>63	59	42
			.3	7.84	7.43	6.93	50	31	34
			.2	6.34	6.27	6.07	31	24	27
			.1	5.44	5.48	5.39	17	18	21
14 x 7 R = 2	5.46	11	.5	14.11	12.66	10.89	>63	>63	57
			.4	11.20	10.09	9.07	62	60	44
			.3	8.56	8.36	7.78	41	36	35
			.2	7.00	7.06	6.81	32	27	28
			.1	6.01	6.11	6.05	20	17	22
10 x 10 R = 3	4.83	10	.5	10.48	11.27	9.65	>63	>63	54
			.4	8.10	8.97	8.04	44	59	42
			.3	6.74	7.37	6.89	31	31	33
			.2	5.84	6.23	6.03	21	23	26
			.1	5.26	5.44	5.36	17	17	20

S_1 - Simulation M_a - Model A M_b - Model B

Fig. 9 Statistics for Networks: Edge Binding


Best path
Messages (per cent)
Path length
π = Probability that each link is busy or impaired.
Simulator
Model A
Model B
π = 0.5
π = 0.4
16
12
8
4
0
1 7 14 21 28 35 42 49 56 63

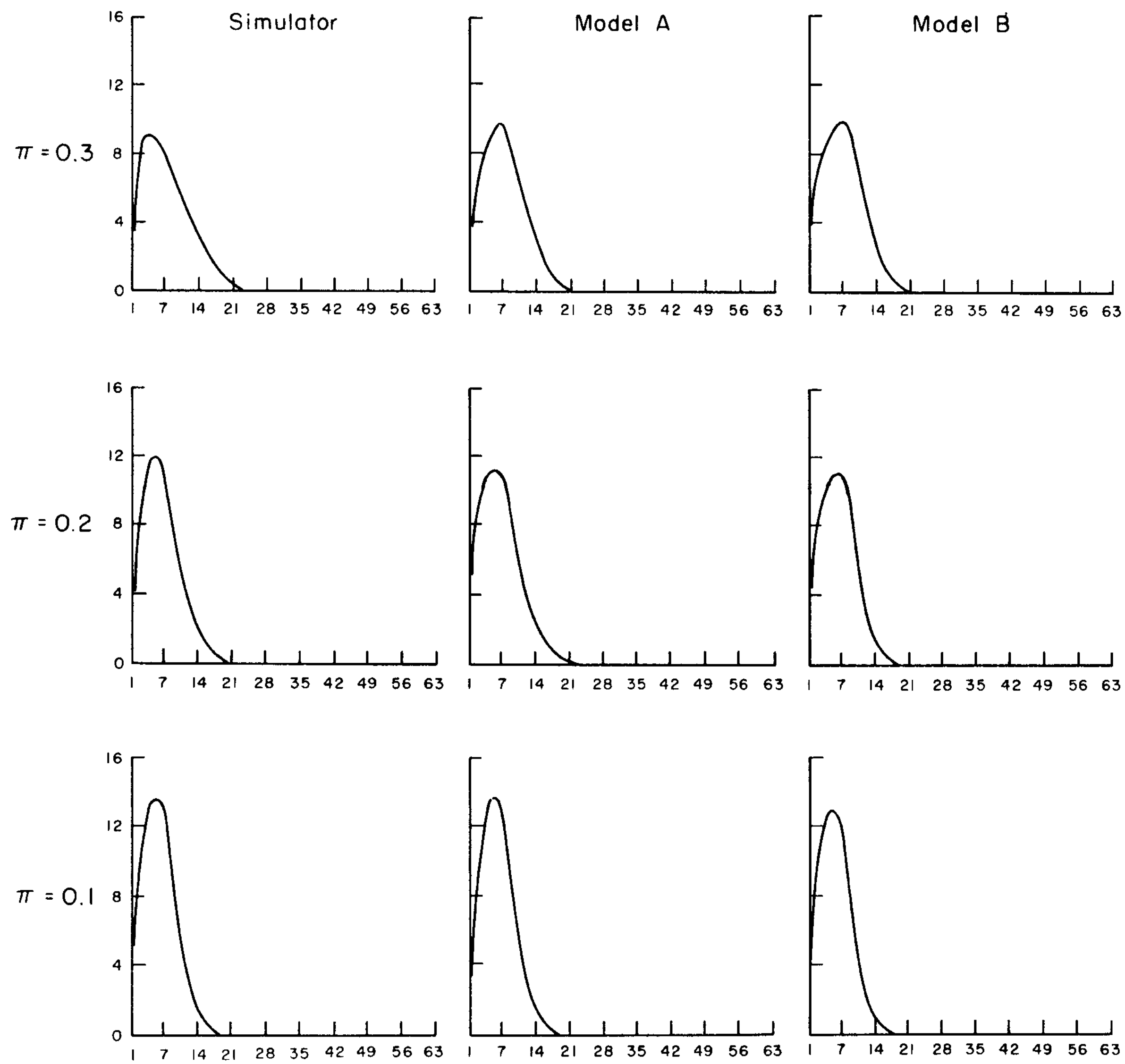


Fig. 10 Distribution of Path-Lengths of Delivered Messages: 14 x 7 Network; R = 3; Edge Binding

VI. VARIATIONS IN TRAFFIC DENSITY

Techniques for smoothing out fluctuations in traffic density are discussed in ODC-II. A technique suggested and evaluated in that Memorandum uses a modification of the HO-table updating algorithm which allows the values of $H_i(j,k)$ to adapt to changing traffic conditions. The adaptation algorithm is described in the footnote to Fig. 12 of the present Memorandum (p. 37), which considers the effects of lengthy bursts of activity from a single station or between a pair of stations. Such effects can be studied by appending to the simulator provisions for treating specified stations as message sources or sinks, with specified weights associated with each station so chosen. A station specified as a source (sink) of weight $k < 1$ will be chosen by the simulator as a message originator (addressee) with probability k. Since correct receipt of each message block is acknowledged by the adjacent receiving station, it is clear that any prolonged increase in the rate of message generation by a single station will result in a commensurate increase in network loading. In such cases we would expect effects similar to those produced by an increase in message loading. In cases where many stations suddenly increase their rate of message generation to a particular station, a similar increase in network loading should occur. This is prevented from occurring by mechanisms described in ODC-VIII.

Several hog cases were simulated on a 7x7 network of redundancy three. The procedure used was the following: first, simulate the network at a 20 per cent loading ratio without sources or sinks; then increase the loading ratio to 40 per cent, specify a commensurate source/sink configuration, and continue the simulation, comparing statistics at each step. The results are summarized:

1) Regions of heavy flow activity centered about sources and sinks;

2) Source/sink configurations caused a 50 per cent decrease in the message-flow rate--this was probably caused by input-choking;

3) Distributions of stack-waiting times--with sources and sinks--were identical to those obtained for 40 per cent loadings without sources or sinks;

4) Single stations acting as both source and sink had little effect on the distribution of path-lengths of delivered messages, the distributions being only slightly less desirable than those obtained for 20 per cent loading with no sources or sinks;

5) Two stations, each acting as both source and sink, had effects on the path-length distribution which depended on the relative positions of the stations within

the network--the more remote the stations, the worse the distributions.

We conjecture that single stations acting as a hog source will reduce message-flow rate but will result in few, if any, dropped messages.

The case of a single source was simulated, such as would occur when a "fraudulent" station attempted to overload the network. The results were as anticipated--the increased loading produced distributions expected from the new loadings. Single sources had essentially the same effects as single source/sinks, with the exception that stack waiting times remained relatively unchanged.

VII. THE BEST-PATH ALGORITHM

Although the algorithm used to set the hand-over number tables, H_i, to their best values was designed solely to minimize computer running time, it might also conceivably find use other than in the Distributed Adaptive Message Block Network in allowing broadcast of best-path information through a network. The algorithm requires a specific, recognizable message-type, an info-message, and a variation of the standard routine doctrine to process such messages. The algorithm is as follows:

1) a station, S_o, that wishes to broadcast best-path information to the rest of the network originates an info-message and transmits the message over all its links, $L_{j,o}$;

2) a station, S_i, receiving an info-message, characterized by (S_o, h), via link $L_{j,i}$ compares h with $H_i(j,o)$:

a) if $h \geq H_i(j,o)$ the message is dropped;

b) otherwise, $H_i(j,o)$ is set to h, h is incremented by unity, and the message is retransmitted over all of S_i's links.

The best-path algorithm used is a parallel application of 1) and 2) above, with all stations acting as originators once and only once. In practice, info-messages may be characterized either by the absence of an addressee or by a "universal" addressee.

Appendix A

PROGRAM DESCRIPTION

A listing of a collection of SCAT-encoded computer routines designed to operate under the aegis of a user-composed supervisory routine is contained in Appendix B. The collection includes routines for defining networks in terms of pertinent parameters, for assigning and re-assigning parameter values, for performing Monte Carlo simulations on networks, for applying Model A and Model B to networks, and for displaying results of simulations and model-runs. The routines are operative on the IBM 7090, require the RAND versions of SOS* for that machine, and are well-described by the listing of Appendix B.

Supervisory routines to perform network calculations must be encoded in SCAT and should use the macro-directives described on p. 4 of the listing. Figure 11 contains an example of a supervisory routine, suitably annotated. The general procedure is to first assign parameter values, then to simulate or model, and finally to display or interrogate results and, perhaps, iterate the procedure. Network parameters are described in Fig. 12; the "normal" values there listed remain in effect until changed by the

---

*Bryan, G. E., Ed., The RAND-SHARE Operating System Manual for the IBM 7090 Computer, The RAND Corporation, RM-3327-PR, September 1962.

```
•       JOB     8109,TEST,JWS618,70,35000,35,C
•       ASSIGN  B2=SYSBR1
•       LOAD    GOIF,NOSQZ,NOLIST,SYSBR1
        CHANGE  NETPGM                   BEGINNING OF SUPERVISORY ROUTINE.
        ROW     14                       14X7 NETWORK,
        COLUMN  7
        WEAVE   3                        REDUNDANCY THREE,
        CHOKE                            WITH INPUT CHOKING, AND
        NOBIND                           NO EDGE-BINDING.
        FLOW                             MESSAGE FLOWS DESIRED.
*
*  FIRST, PRINT BEST-PATH DISTRIBUTION.
        BESTHO  SYSTEM                   EXIT IF NOGO.
* THEN APPLY SIMULATOR, MODEL A, AND MODEL B -- VARYING LOADING FROM 10
*  PERCENT TO 50 PERCENT IN JUMPS OF 10, USING UNBOUNDED HMAX(NORMAL CASE).
        AXT     5,1
•A      IMPAIR  •L,1                     SET LOADING, THEN
        SIMUL   SYSTEM,3000              SIMULATE FOR 3000 CYCLES. THEN DISPLAY
        PRINT   S1DIST                   PATH-LENGTH DISTRIBUTION,
        PRINT   S1WAIT                   STACK-WAITING-TIME DISTRIBUTION, AND
        PRINT   SUFLOW                   MESSAGE-FLOW PATTERNS.
        MODELA  SYSTEM                   NEXT, APPLY MODELS. ( DISTRIBUTIONS
        MODELB  SYSTEM                   ARE DISPLAYED).
        TIX     •A,1,1
        IMPAIR  •L2                      NEXT RUN FOR 3000 CYCLES AT
        SIMUL   SYSTEM,3000              20 PERCENT LOADING.
        IMPAIR  •L4                      NEXT, DOUBLE LOADING AND DEFINE STATION
        SOURCE  1,•5                     1 AS A SOURCE OF WEIGHT 1/2.
        CONT    1000                     CONTINUE, THEN DISPLAY.
        PRINT   S1DIST
        PRINT   S1WAIT
        PRINT   S1FLOW
        SINK    1,•5                     THEN MAKE STATION 1 A SINK OF WEIGHT
*                                        1/2, MAINTAINING ITS SOURCE STATUS.
        CONT    1000
        PRINT   S1DIST
        PRINT   S1WAIT
        PRINT   S1FLOW
        TRA     SYSTEM                   FINI
        DEC     •1
•L2     DEC     •2
        DEC     •3
•L4     DEC     •4
        DEC     •5
•L      EQU     *
```

Fig. 11 Sample Network Program

supervisory program. Much of Figs. 11 and 12, and the macro-directive listing in Appendix B, is self-explanatory.

The parameter WEAVE, which specifies network connectivity, is defined by Fig. 13. The parameter GRAIN defines the ratio of message-unit length (time required to insert a message into a link) to the time required by a station to route a message. Since message-routing time is equivalent to simulation cycle time, GRAIN defines the "coarseness" of the simulation. Note that simulations may be halted and then continued; during these "pauses" the user may display results of the simulation and may change the values of certain parameters. Parameters which may be changed during the course of a simulation are indicated in Fig. 12 with an asterisk. Sources and sinks may be defined, deleted, or have their weights changed during a simulation. Deletion of a source or sink is accomplished by assigning a zero weight. One further restriction exists: the impairment factor, IMPAIR, may not be reset to a value greater than that which held at the initiation of the simulation; this, however, is not a real restriction, since the simulator can be initiated for a zero-time run.

Execution times for application of the two models are negligible. Simulation times are a function of the size and connectivity of the net being simulated and of the traffic density. A reasonable approximation to

Parameter Name	Function	Restrictions	Normal Value
*CHOKE	= 0 implies no input-choking; ≠ 0 implies input-choking.	none	≠ 0
ALPHA	The loading factor, described in Sec. III.	computed as function of IMPAIR	0
HMAX	The maximum hand-over number; HMAX = 63 implies messages are never dropped.	HMAX an integer; $0 < HMAX \leq 63$	63
HPRIME	Used to preset the hand-over table, H; HPRIME = 0 implies H is preset to "best" values, otherwise H is preset with values drawn from uniform distribution between HPRIME and HMAX.	h' an integer $0 \leq h' \leq HMAX$	0
ROW COLUMN	Number of rows / columns in network.	integers	--
GRAIN	The "grain" of the simulation; the ratio of message-length (time to place message on or accept message from a link) to the time required by a station to process the message.	integer > 0	2
TPMAX TPMIN	Link-lengths are assigned values drawn from uniform distribution bounded by (TPMAX+GRAIN, TPMIN+GRAIN).	integral number of station processing times	6 0

Fig. 12 Network Parameters

Parameter Name	Function	Restrictions	Normal Value
STACK	Maximum stack-storage, described in Sec. IV.	integer	8
WEAVE	The connectivity of the network; see Fig. 13.		--
*IMPAIR	The probability of link impairment; used by M_a and M_b, and the simulator.	decimal fraction	0
RANDOM	An initial pseudo-random number.	large, odd integer	976525005
FLOW	$\neq 0$ implies message-flow statistics are desired; $= 0$ implies none are desired.	none	$\neq 0$
BIND	$\neq 0$ implies edge binding (Fig. 7); $= 0$ implies no edge binding.	none	0
*ADAPT	$= 0$ implies H_i table entries, h_i, are updated by h_i = minimum (h_i, h); $\neq 0$ implies H_i table entries are updated by adaptation algorithm.[a]	none	0
*LEARN	Learning factor for adaptation.	$0 < \text{LEARN} \leq 1$	1
*FORGET	Forgetting factor for adaptation.	$0 < \text{FORGET} \leq 1$	0

[a] Let $D = (h - h_i)$

If $D \begin{matrix} > \\ = \\ < \end{matrix} 0$ then $\begin{matrix} H0 = H0 + D \cdot \text{FORGET} \\ H0 = H0 \\ H0 = H0 + D \cdot \text{LEARN} \end{matrix}$

Fig. 12 (Continued)

WEAVE	Redundancy Level
1	1
2	2
3	3
4	4
5	3
6	2
7	2

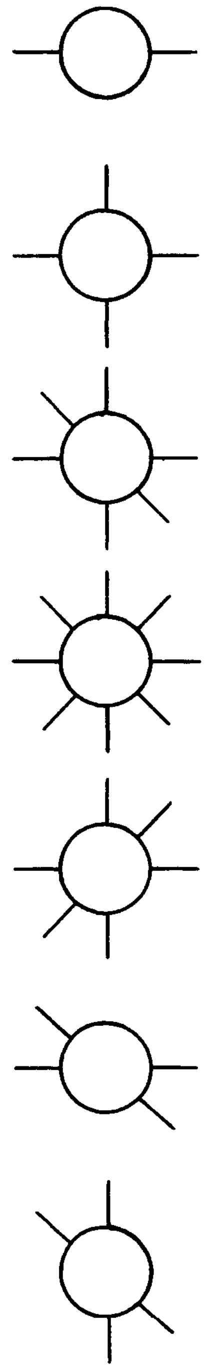


Fig. 13 WEAVE Parameter

execution times for an n x m net of redundancy r is given by:

$$\text{time per simulation cycle} = \frac{2\cdot r\cdot n\cdot m\cdot \alpha}{3} \text{ ms}$$

where α is the loading factor described in Sec. III.

Computer storage required is a function of net size and connectivity, of traffic density, and of other desiderata. An n x m net of redundancy-r requires approximately

$$(4\cdot n\cdot m)(1+r+r\alpha) + \frac{(n\cdot m)}{2} \cdot [n\cdot m\cdot r - (r-1)(n+m)]$$

words of computer storage, 24,000 words being available. A 10x10 network of redundancy-four, or a 11x11 network of redundancy-three can be accommodated.

Appendix B

PROGRAM LISTING

In this appendix is presented the program listing, together with a Table of Contents to the routines listed. The page numbers referred to in this table are the numbers internal to the listing, not the pages in the Memorandum.

TABLE OF CONTENTS

* INDICATES ROUTINES OF DIRECT INTEREST TO USERS

LIST OF MACRO-DIRECTIVES

ALL MACROS PRESERVE IR'S AND SENSE

```
'I' MEANS EITHER OF 1) INTEGER
                    2) LOCATION
                    3) LOCATION TAG
```

MACROS FOR SETTING PARAMS

```
BINC
NCBINC


CHOKE
NCCHOK


FLOW
NCFLOW


ADAPT
NCDAPT     NO ADAPTATION TO BE USED


HMAX      I
WEAVE     I
ROW       I
COLUMN    I
TPMAX     I
TPMIN     I
STACK     I
HPRIME    I
GRAIN     I
IMPAIR    LOC,TAG
```

MACROS TO DEFINE 'HOG' STATIONS. STATION
N IS TO BE ASSIGNED AS A MESSAGE
SOURCE (OR SINK)WITH PROBABILITY K

```
SOURCE    N,K                 K A DECIMAL FRACTION
SINK      N,K
```

```
              MACROS TO CEFINE SPECIAL LINKS TC BE ADDFC
              OR DELETEC.

LINK      N1,K1,N2,K2            LINK FRCM STATICN N1 LINK
                                 NR. K1, TC STATICN N2,LINK
                                 NR K2, IS TO BE ADDED.

CLT       N1,K1,N2,K2
```

MACRC DIRECTIVES

```
        'E' MEANS 'ERROR EXIT'
        'L' MEANS 'LOCATION CF SPECIAL-LINK CONTROL-WCRD'
      WHERE (L) CONTAINS --
          LOCATION OF FIRST SPECIAL LINKS,,NR OF SPECIAL LINKS
        ANC L=0 OR (L)=0 IMPLIES NC LINKS
        I REFERS TC EITHER LCCATION OR INTEGER


SIMUL   E,I,L                START S1 SIMULATION, RUN FOR
                             I OR (I) CYCLES

CCNT                         CONTINUE SIMUL FOR I OR (I) CYCLES

MCCELA  E,L                  APPLY MCDEL A. CISPLAY DIST.
MCDELB  E,L                  APPLY MODEL B. CISPLAY DIST.

BESTHC  E,L                  GENERATE HO-TABLE BY BEST-PATH
                             ALGORITHM. DISPLAY 'BEST' CIST.,HBAR

SETHMX  X,L                  USEC IMMEDIATELY AFTER APPLICATICN OF
          MODEL A OR B TO SET HMAX TC A VALUE THAT, CN THE
          BASIS OF THE APPLICABLE MCCEL'S RESULTS(CISTRIBUTICN)
          , WILL INSURE A FRACTICNAL DRCP-CUT NCT EXCEECING (L)
        EXITS TO X IF NO SUCH VALUE EXISTS.
        MAY ALSO BE USED AFTER OR DURING SIMULATION RUN
          TO ADJUST HMAX.

PRINT   S1CIST          DIST. OF PATH-LENGTHS OF MSGS CLVRC BY S1
PRINT   S1FLOW          TRAFFIC FLOW GENERATEC BY S1
PRINT   S1WAIT          DIST. OF STACK WAITING TIMES
```

```
           BEGINNING OF NETWORK PROGRAM.
NETPGM  TSX     FORMAT,4                TO DEFINE XFORMS
  -             SUPERVISORY PROGRAM FOLLOWS
    +1  TRA     SYSTEM                  PROTECTION


DUMP    CLA*    1,4
    +1  TZE     2,4
    +2  ALS     18
    +3  ADD*    1,4
    +4  STO     *+5
    +5  STL     SYSCB1
    +6  TXL     SYSCB2,,21
        CORE     LIMIT,LIMIT,0
    +7  STL     SYSCB1
   +11  TRA     2,4
        PANEL
ERROR   STL     SYSCB1
    +2  TSX     DUMP,4
    +3  PZE     TABLES
    +4  TSX     DUMP,4
    +5  PZE     LINTBL
    +6  TSX     DUMP,4
    +7  PZE     BUSY
    +8  TSX     DUMP,4
    +9  PZE     NODTBL
   +10  TSX     DUMP,4
   +11  PZE     MSGTBL
   +12  TSX     DUMP,4
   +13  PZE     HODTBL
   +14  TSX     DUMP,4
   +15  PZE     HOTBLE
   +16  TSX     DUMP,4
   +17  PZE     STDTBL
   +18  TSX     DUMP,4
   +19  PZE     SCTBLE
   +20  TSX     DUMP,4
   +21  PZE     SKTBLE
   +22  TSX     DUMP,4
   +23  PZE     POOL
   +24  TSX     DUMP,4
   +25  PZE     PARAM
   +26  TSX     S1DIST,4
   +27  TSX     S1FLOW,4
   +28  TSX     S1WAIT,4
   +29  TRA     SYSTEM
```

```
              PARSET USED TO SET ALL PARAMS

              CALL   SEQU   STL    .EXIT
                            TXL    PARSET,,PARAM NAME
                            TXL    LOC,TAG,0 1 OR 2

              SAVES IR'S AND SENSE


PARSET  SXA     PRSX,4
    +1  CLA*    .EXIT
    +2  ARS     18
    +3  STA     PRS1                      PARAM NAME
    +4  CLA     .EXIT
    +5  ADD     K.A1
    +6  STA     PRS2
    +7  ADD     K.A1
    +8  STA     .EXIT
PRS2    CLA     **0                       FETCH ARGUMENTS
    +1  STA     PRS4                      LOC OF NEW VALUE(OR VALUE ITSELF)
    +2  STT     PRS4
    +3  LDQ*    PRS4
    +4  PDX     ,4                        0,1 OR 2
    +5  TXH     PRS3,4,0                        SETTING FLOATING POINT QUANTITIES
    +6  CAL     SYSORG                    ELSE TEST IF
    +7  ANA     K.M6                            VALUE OR LOC OF VALUE
    +8  CAS     PRS4
    +9  LDQ     PRS4                      VALUE
   +10  LDQ     PRS4                      VALUE
   +11  XCA
   +12  SSP
   +13  CAS     K.M2
   +14  TRA     *+3
   +15  TRA     *+2
   +16  TRA     PRS1-1
   +17  XCA
   +18  PXA
   +19  LLS     8
   +20  SUB     KINT2
   +21  TPL     *+2
   +22  PXA
   +23  STA     *+2
   +24  PXA
   +25  LLS     **0
   +26  XCA
PRS1    STQ     **0                       LOC OF VALUE
    +1  TRA     PRSX
PRS3    XCA
    +1  SSP
    +2  CAS     K.M2
    +3  TRA     *+4
    +4  NOP
    +5  ORA     KINT1
    +6  FAD     KINT1
```

```
     +7  STO*    PRS1
PRSX     AXT     **0,4
     +1  TRA*    .EXIT
PRS4     PZE
```

```
                .ENTRY USED FOR ALL MACRO DIRECTIVES
                CALL   SEQU   STL    .EXIT
                              TXL    .ENTRY,,LENGTH OF CALL SEQU + 2
                              TSX    PROCESSOR,4
                              CALLING SEQU FOR PROCESSOR

                MAX CALL SEQU LENGTH = 8


.EXIT     PZE    **0
.ENTRY    SWT    5                                  EXCESS TIME TEST
    +1    TRA    *+2                                CONTINUE
    +2    TRA    SYSTEM                             FINI
    +3    CLA    .EXIT
    +4    ALS    18
    +5    STD    .EN1
    +6    SXA    .ENX,4
    +7    CLA*   .EXIT
    +8    PDX    ,4
    +9    TXI    *+1,4,.EN2
   +10    SXA    .EN3,4
   +11    PDX    ,4
.EN1      TXI    *+1,4,**0
    +1    SXA    .EN4,4
    +2    SXA    .EXIT,4
    +3    PDX    ,4
    +4    TXI    *+1,4,-1
.EN4      CLA    **0,4
.EN3      STO    **0,4
    +1    TIX    .EN4,4,1
    +2    CLA    .EN5
    +3    AXT    ,4
.EN2      STO*   .EN3
    +1    TSX    ERROR,4
    +2    TSX    ERROR,4
    +3    TSX    ERROR,4
    +4    TSX    ERROR,4
    +5    TSX    ERROR,4
    +6    TSX    ERROR,4
    +7    TSX    ERROR,4
    +8    TSX    ERROR,4
    +9    TSX    ERROR,4
   +10    TSX    ERROR,4
.EN5      TRA    .ENX
.ENX      AXT    **0,4
    +1    TRA*   .EXIT
.ERROR    AXT    1,4
    +1    CLA*   .EN3
    +2    ARS    18
    +3    STA    *+2
    +4    LXA    .ENX,4
    +5    TRA    **0
```

PRESET VALIDATES PARAMS, DISPLAYS PARAMS
INITIALIZES NET, AND ALLOCATES STORAGE.

```
        CALL SEQU    TSX   PRESET,4
                     PZE  LOC OF FIRST SPEC. LINK,,NR OF LINKS
                     RETURN IF STORAGE EXCEEDED OR ERRORS
                     NORMAL RETURN

PRES    STL     DSPY                     ENTRY FROM SIMULATOR
   +1   TRA     PRO
PRESET  STZ     DSPY                     ENTRY FROM MODELS AND BEST-PATH DSPLY
        BEGIN    3,7,1
PRO     TXL     *+7,0,0                  SUBROUTINE LINKAGE
  +11   CLA     1,4
  +12   STO     PR5
            SET RANDOM AND SET ACTIVITY LEVELS
  +13   CLA     TPMAX
  +14   SUB     TPMIN
  +15   SSP
  +16   ORA     KINT1
  +17   FAD     KINT1
  +18   SUB     KINT3                          (TPMAX-TPMIN)/2 = D
  +19   STO     K.T1
  +20   CLA     GRAIN
  +21   ORA     KINT1
  +22   FAD     KINT1                          ML = MESSAGE LENGTH
  +23   STO     K.T2
  +24   FAD     K.T2
  +25   FAD     K.T1
  +26   FDP     K.T2                           (D+2ML)/ML = ACTIVITY FACTOR
  +27   STQ     FACTOR
  +28   FMP     IMPARE                         ACTIVITY LEVEL
  +29   STO     ALPHA
  +30   STO     MAXACT
  +31   CLA     RANDOM
  +32   LBT
  +33   CLA     K.R
  +34   SLW     RANDOM
  +35   SLW     RND
  +36   NZT     DSPY
  +37   TRA     PR7
                DISPLAY PARAMETERS
  +38   TSX     PARAMS,4
  +39   TRA     PR7
```

PARAMS PRINTS PARAMETER VALUES.

```
        BEGIN    1,7,1
PARAMS  TXL     *+7,0,0                        SUBROUTINE LINKAGE
```

```
          XEJECT
    +11   STL     SYSOED
    +14   AXT     4,2
    +15   AXT     RANDOM-ALPHA,1
          XPRINT  D,2
PR1.1     STL     SYSOED
     +3   PZE     PM,1,1
     +4   PZE     RANDOM,1,1
     +5   TXI     *+1,1,-1
     +6   TIX     PR1.1,2,1
          XPRINT  E,2
PR3       STL     SYSOED
     +3   PZE     PM,1,1
     +4   PZE     RANDOM,1,1
     +5   TIX     PR3,1,1
          XPRINT  F,2
     +6   STL     SYSOED
     +9   PZE     PM,,1
    +10   PZE     RANDOM,,1
    +11   AXT     0-4,2
PR6       AXT     ,1
     +1   ZET     CHOKE-1,2
     +2   AXT     1,1
          XPRINT  G,2
     +3   STL     SYSOED
     +6   PZE     PMC-1,2,1
     +7   PZE     NO,1,1
     +8   TXI     *+1,2,1
     +9   TXH     PR6,2,0
          RETURN  PARAMS
    +10   TRA     PARAMS+1


             FLUSH OUTPUT BUFFERS
PR7       STL     PTTGL
     +1   TSX     SYSDSK,4
     +2   PZE     SYSMOT
     +3   TSX     SYSDSK,4
     +4   PZE     SYSMIT
     +5   STZ     SYSIBC
     +6   TSX     SYSBFD,4
     +7   PZE     SYSSBF,,512
     +8   STR
     +9   STZ     BFTST
             TEST AND SET CONNECTIVITY
    +10   LXA     WEAVE,4
    +11   TXH     *+2,4,0
    +12   TRA     PRX
    +13   TXL     *+2,4,7
    +14   TRA     PRX
    +15   LAC     WEAVE,4
    +16   CLA     CTABLE,4
    +17   STO     LINKS
             TEST LINK-LENGTH LIMITS
```

```
    +18  CLA    TPMAX
    +19  TNZ    *+2
    +20  TRA    PRX
    +21  CAS    TPMIN
    +22  TRA    *+3
    +23  TRA    *+2
    +24  TRA    PRX
    +25  ADD    GRAIN
    +26  ADD    K.A1
    +27  CAS    TPHIGH
    +28  TRA    PRX
    +29  NOP
            TEST HANDOVER-NUMBER LIMITS
    +30  CLA    MAXHO
    +31  ALS    3
    +32  TNZ    *+2
    +33  TRA    PRX
    +34  CAS    HOHI1
    +35  CLA    HOHI1
    +36  NOP
    +37  STO    HOMAX
    +38  CLA    INITHO
    +39  ALS    3
    +40  CAS    HOMAX
    +41  TRA    PRX
    +42  NOP
    +43  STO    HOME1
    +44  AXT    LIMIT-MSG2-1,1
    +45  STZ    LIMIT,1                  CLEAR STORE
    +46  TIX    *-1,1,1
    +47  LXD    CNTTBL,1         CLEAR COUNTS
    +48  STZ    CNTTBL,1
    +49  TIX    *-1,1,1
    +50  AXT    TPU,1            CLEAR TRAFFIC-LIST HEADS
    +51  STZ    TP,1
    +52  TIX    *-1,1,1
    +53  SXA    TP,1
    +54  TSX    INL,4                    SET LINK TABLE
    +55  TSX    INS,4                    GENERATE STORAGE MAP
    +56  TRA    PRX                      DRAT
    +57  NZT    NODES                    TEST VALIDITY OF PARAMS
    +58  TRA    PRX
    +59  NZT    LINES
    +60  TRA    PRX
    +61  NZT    MSGS
    +62  TRA    PRX
    +63  TSX    IA,4                     SET LINE TABLE
    +64  TSX    IL,4             SET     LINK-LENGTHS AND LINKS
PR5      PZE    **0,,**0         DEFINES SPECIAL LINKS
     +1  TRA    PRX
     +2  TSX    IH,4
     +3         HOMAX
     +4         HOME1
     +5  TRA    PRX
```

```
            NOW CALCULATE EXACT LOADING FROM TRUE NR OF LIMKS
     +6  LDQ    NODES
     +7  MPY    LINKS
     +8  STQ    K.T4
     +9  LDQ    HONCNT
    +10  PXA
    +11  DVP    NODES
    +12  XCA
    +13  STO    K.T3
    +14  XCA
    +15  MPY    MSGS
    +16  DVP    K.T4
    +17  CAL    MSGTBL
    +18  ARS    19
    +19  ANA    K.M6
    +20  TLQ    *+2
    +21  XCA
    +22  STQ    MSGS                     EXACT LOADING
    +23  STQ    LOAD
    +24  CLA    HONCNT
    +25  SUB    K.T3
    +26  STO    HONCNT
            ALLCT EXTRA BUFFERS IF STORAGE AVAILABLE
    +27  LAC    IH7,1
    +28  TXI    *+1,1,32767
    +29  TXL    PR4,1,255
    +30  STL    BFTST
    +31  SXD    *+4,1
    +32  LXA    IH7,1
    +33  SXA    *+2,1
    +34  TSX    SYSBFD,4
    +35  PZE    **0,,**0
    +36  STR
         RETURN  PRO
PR4      TRA    PRO+1
         DUMP    TABLES
PRX      TSX    DUMP,4
         DUMP    PARAM
     +2  TSX    DUMP,4
         RETURN  PRO,1
     +4  AXT    1,4
PMC      BCI    1,CHOKE
     +1  BCI    1,FLOW
     +2  BCI    1,BIND
     +3  BCI    1,ADAPT
     +4  BCI    1,ALPHA
     +5  BCI    1,IMPAIR
     +6  BCI    1,LEARN
     +7  BCI    1,FORGET
     +8  BCI    1,HMAX
     +9  BCI    1,WEAVE
    +10  BCI    1,ROW
    +11  BCI    1,COLUMN
    +12  BCI    1,TPMAX
```

```
     +13  BCI     1,TPMIN
     +14  BCI     1,STACK
     +15  BCI     1,HPRIME
     +16  BCI     1,GRAIN
PM        BCI     1,RANDOM
```

INL PRESETS LINK TABLE AS FCT OF COLUMN DIM.

```
CALL SEQU TSX      INL,4
                RETURN
```

```
INL       TRA     *+1
          BEGIN    1,7
INLO      TXL     *+5,0,0                  SUBROUTINE LINKAGE
     +8   CLA     CX
     +9   STA     INL1
    +10   PAC     ,1
    +11   AXT     4,4
    +12   AXT     1,2
INL3      SXD     INL2,2
     +1   PXA     ,1
     +2   STA     LINK+4,4
     +3   PAC     ,2
     +4   PXA     ,2
     +5   STA     LINK+8,4
INL2      TXI     *+1,1,**0
INL1      AXT     **0,2
     +1   TIX     INL3,4,1
          RETURN   INLO
     +2   TRA     INLO+1
```

```
              INS COMPUTES STORAGE MAP AS FUNCTION OF
                   CX     =      COLUMN DIM
                   RX     =      ROW     ''
                   LINKS =       LINKS PER NODE
                   ACTIVE=       PERCENT MSG ACTIVITY

              CALL   SEQ    TSX    INS,4
                            RETURN IF STORE EXCEEDED
                            NORMAL RETURN


INS      TRA     *+1
         BEGIN    2,7
INSO     TXL     *+5,0,0                     SUBROUTINE LINKAGE
   +8    LDQ     CX
   +9    RQL     3
  +10    STQ     D1                               8*COLUMN DIM
  +11    MPY     RX
  +12    STQ     LINES                            8*NR OF NODES = LINE TABLE LENGTH
  +13    LRS     3
  +14    STQ     NODES                            NR OF NODES
            CALCULATE MSG LOADING USING APPARANT NR OF LINKS
  +15    MPY     LINKS
  +16    XCA
  +17    ORA     KINT1
  +18    FAD     KINT1
  +19    XCA
  +20    CLA     ALPHA
  +21    CAS     K.M2
  +22    TRA     *+3
  +23    NOP
  +24    STZ     ALPHA
  +25    FMP     ALPHA
  +26    SSP
  +27    XCA
  +28    PXA
  +29    LLS     8
  +30    SUB     KINT2
  +31    TPL     *+2
  +32    PXA
  +33    STA     *+2
  +34    PXA
  +35    LLS     **0
  +36    ZET     ACTTGL
  +37    PXA                         NO MSG STG NEEDED FOR MODELS
  +38    ADD     K.A1
  +39    STO     MSGS                             DESIRED NR OF MSGS
            NEXT ALLOCATE STORAGE FOR ALL TABLES
  +40    ADD     RX                               ALLOW EXTRA STORAGE
  +41    ADD     RX                               FOR POSSIBLE BINDING
  +42    ADD     CX                               OF COARSELY CONNECTED
  +43    ADD     CX                               NETS, OF RED 2 FOR EXAMPLE.
  +44    ALS     1                                LENGTH OF MESSAGE TABLE
  +45    PAX     ,1
```

```
+46   SXD   MSGTBL,1
+47   ADD   K.A1
+48   ADD   BASE                          TOP OF LINE TABLE
+49   STO   LINTBL
+50   LXA   LINES,1
+51   SXD   LINTBL,1
+52   ADD   LINES
+53   STA   LINE1
+54   ADD   K.A7                          BASE OF LINE TABLE
+55   STA   LINEA                         TOP OF NODE TABLE
+56   STA   LINEB
+57   STA   LINEC
+58   ADD   K.A1
+59   STO   NODTBL
+60   CLA   NODES
+61   CAS   K.M5
+62   TRA   INSX                          OVERSIZE NET
+63   TRA   INSX
+64   STO   SKH
+65   STO   SCH
+66   ADD   NODTBL
+67   STA   N1A
+68   STA   N1B
+69   STA   N1C
+70   ADD   NODES
+71   ADD   K.A1
+72   STA   N2A
+73   STA   N2B
+74   STA   N2C
+75   ADD   NODES
+76   ADD   K.A1
+77   STA   N3A
+78   STA   N3B
+79   STA   N3C
+80   NZT   SNAP
+81   TRA   *+6
+82   ADD   NODES
+83   ADD   K.A1
+84   STA   N4A
+85   STA   N4B
+86   STA   N4C
+87   SUB   NODTBL
+88   PAX   ,1
+89   ADD   NODTBL
+90   SXD   NODTBL,1
+91   ADD   K.A1
+92   STO   HODTBL
+93   ADD   MAXHO
+94   STA   HODA                          ALLOCATE HO-DISTRIBUTION TABLE
+95   STA   HODB
+96   STA   HODC
+97   ADD   K.A1
+98   STO   STDTBL
+99   ADD   MAXHO
```

```
+100   STA    STDA
+101   STA    STDB
+102   STA    STDC
+103   LXA    MAXHD,2
+104   SXD    HODTBL,2
+105   SXD    STDTBL,2
+106   ADD    K.A1
+107   STO    SCTBLE
+108   ADD    SSLIM
+109   STA    SC1A
+110   STA    SC1B
+111   STA    SC1C
+112   ADD    K.A1
+113   ADD    SSLIM
+114   STA    SC2A
+115   STA    SC2B
+116   STA    SC2C
+117   ADD    K.A1
+118   STO    SKTBLE
+119   ADD    SSLIM
+120   STA    SK1A
+121   STA    SK1B
+122   STA    SK1C
+123   ADD    K.A1
+124   ADD    SSLIM
+125   STA    SK2A
+126   STA    SK2B
+127   STA    SK2C
+128   STZ    SOURCE
+129   STZ    SINK
+130   STZ    SKCUM
+131   STZ    SCCUM
+132   LXA    SSLIM,1
+133   SXD    *+1,1
+134   TXI    *+1,1,**0
+135   SXD    SCTBLE,1
+136   SXD    SKTBLE,1
+137   ADD    K.A1
+138   STO    BUSY
+139   ADD    LINES
+140   ADD    K.A7
+141   STA    BUSYA
+142   STA    BUSYB
+143   STA    BUSYC
+144   LXA    LINES,1
+145   SXD    BUSY,1
+146   ADD    K.A1
+147   STO    HOTBLE
+148   CAS    UPPER
+149   TRA    INSX
+150   TRA    INSX
+151   CAS    LOWER
+152   TRA    *+3
+153   TRA    INSX
```

```
  +154   TRA     INSX
  +155   LXA     NODES,1
  +156   PXD     ,1
  +157   ADD     NODES
  +158   STO*    N3A                              PRESET REFERENCE TABLE
  +159   SUB     K.A1D1
  +160   TIX     *-2,1,1
  +161   CLA     LINE1
  +162   ADD     D1
  +163   SUB     LINES
  +164   STA     BOTTOM
  +165   ADD     LINES
  +166   SUB     K.A8
  +167   STA     LEFT
         RETURN   INSO
  +168   TRA     INSO+1
         RETURN   INSO,1
INSX     AXT     1,4
```

```
                IA INITIALIZES LINE TABLE,
                IMAGINARY LINES ARE MADE BUSY.

                CALL SECU     TSX   IA,4
                                    RETURN
                CONN = CONNECTIVITY TYPE
                CTABLE(...) DESCRIBED ELSEWHERE

IA        TRA     *+1
          BEGIN    1,7
IA0       TXL     *+5,0,0                    SUBROUTINE LINKAGE
     +8   LXA     CONN,1
     +9   LXA     LINES,2
IA1       AXT     8,4
     +1   LDQ     CTABLE,1
     +2   PXA
IA2       LLS     0                              BUSY-FLAG  OFF FOR REAL LINE
     +1   STO*    LINE1                          ON FOR IMAG LINES
     +2   RQL     1
     +3   TNX     IA3,2,1
     +4   TIX     IA2,4,1
     +5   TRA     IA1
IA3       LXA     D1,2                           NOW SET BOUNDARIES
     +1   SXD     IA9,2
     +2   CAL     K.MZE
IA4       LDQ     B.T.
     +1   AXT     8,4
IA5       TQP     *+2
     +1   ORS*    BOTTOM
     +2   RQL     18
     +3   TQP     *+2
     +4   ORS*    LINE1                          TOP
     +5   RQL     19
     +6   TNX     IA6,2,1
     +7   TIX     IA5,4,1
     +8   TRA     IA4
IA6       LXA     LINES,1
IA7       SXA     *+1,1
     +1   AXT     **0,2
     +2   AXT     8,4
     +3   LDQ     SIDES
IA8       TQP     *+2
     +1   ORS*    LINE1                          RIGHT
     +2   RQL     18
     +3   TQP     *+2
     +4   ORS*    LEFT
     +5   RQL     19
     +6   TNX     IA9,4,1
     +7   TXI     IA8,2,-1
IA9       TIX     IA7,1,**C
     +1   LXA     LINES,1                        DO SAME FOR BUSY TABLE
     +2   TXI     *+1,1,7
     +3   CLA*    LINEA
```

```
+4   STO*    BUSYA
+5   TXI     *+1,1,-1
+6   TXH     *-3,1,7
     RETURN   IAO
+7   TRA     IAO+1
```

```
              IL INSERTS NEW LINKAGES, ASSIGNS TP'S,DRAWN
              FROM UNIFORM DIST. BTWN TP-BCUNDS, TO ALL
              REAL LINES, AND POINTS IMAGE LINES AT EACH CTHER.

              CALL SEQU   TSX IL,4
                          PZE A,,B
                          ERROR RETLRN
                          RETURN
              WHERE B = NR OF SPECIAL LINKS
                    A = ADDRESS OF FIRST LINK SPECIFIER.
              LINKS FROM (I1,J1) TO (I2,J2) ARE SPECIFIED BY
    LINK    I1,J1,I2,J2


          BEGIN   3,7
IL        TXL     *+5,0,0                      SUBRCUTINE LINKAGE
     +8   CLA     TPMAX
     +9   SUB     TPMIN
    +10   ADD     K.A1
    +11   STO     IL.T
    +12   CLA     1,4
    +13   PDX     ,1
    +14   TXL     IL0,1,0              NO SPECIAL LINKS
    +15   AXT     IL4,1
    +16   SXA     IA17,1               SET RETURN SWITCH.
    +17   STL     K.T2
    +18   PAX     ,4
    +19   SXD     *+1,1
    +20   TXI     *+1,4,**0
    +21   SXA     IL1,4
    +22   LXA     LINES,2
    +23   TXI     *+1,2,7              HIGHEST LINE NR
    +24   SXD     IL2,2
    +25   SXD     IL3,2
IL1       CLA     **0,1                GET LINK SPECIFIER
     +1   PAX     ,2                   (I1,J1)
     +2   PDX     ,4                   (I2,J2)
     +3   TXL     ILX,2,7              ARE LINES WITHIN BCUNDS
     +4   TXL     ILX,4,7
IL2       TXH     ILX,2,**0
IL3       TXH     ILX,4,**C
     +1   TPL     IA10                         TO IA1C IF LINK
     +2   CLA     K.MZE                OTHERWISE, CUT LINK
     +3   STO*    LINEB
     +4   STO*    LINEC
     +5   STO*    BUSYB
     +6   STO*    BUSYC
IL4       TIX     IL1,1,1
IL0       ZET     BIND                 TEST FOR EDGE-BINDING.
     +1   TRA     IA18                 TO BE BOLND.
IL00      AXT     IA15,1               NOW, SET
     +1   SXA     IA17,1               RETURN SWITCH, AND
     +2   LXA     LINES,2              BEGIN ORDINARY LINKS.
     +3   TXI     *+1,2,7
```

```
    +4   STZ      K.T2
IA14     CLA*     LINEB
    +1   TMI      IA15                       IGNORE DEAD LINES
    +2   TNZ      IA15                       IGNORE PROCESSED LINES
IA10     LDQ      RND
    +1   MPY      K.R
    +2   STQ      RND
    +3   MPY      IL.T
    +4   ADD      TPMIN
    +5   ADD      GRAIN                      TP + MSG LENGTH USED AS TP
IA13     ALS      18
    +1   STO*     LINEB                      SET TP
    +2   ZET      K.T2
    +3   TRA      IA16                       SPECIAL LINE
    +4   STD      K.T1
    +5   PXA      ,2                               GENERATE
    +6   LGR      3                                IMAGE
    +7   PAX      ,4                               LINE
    +8   PXA                                       NR.
    +9   LGR      3
   +10   CAQ      LINK,1,1
   +11   SXD      *+1,1
   +12   TXI      *+1,4,**0
   +13   XCL
   +14   PXA      ,4
   +15   LGL      3
   +16   PAX      ,4
   +17   CLA      K.T1                             SAME  TP TO
IA16     STO*     LINEC                      IMAGE LINE.
    +1   PXA      ,2                         IMAGES
    +2   STO*     BUSYC                            POINT
    +3   PXA      ,4                               TO
    +4   STO*     BUSYB                            EACH OTHER
IA17     TRA      **0                        RETURN SWITCH.
IA15     TXI      *+1,2,-1
    +1   TXH      IA14,2,7
         RETURN    IL
    +2   TRA      IL+1
         RETURN    IL,1
ILX      AXT      1,4
IA18     AXT      IA19,1                     BINDING EDGES.
    +1   SXA      IA17,1                     SET RETURN SWITCH
    +2   CLA      CX                         COLUMN BECOMES
    +3   STO      .LS-4                      TOP-EDGE-COUNT,
    +4   STO      .LS-2                      BOTTOM-EDGE-COUNT,
    +5   ALS      3
    +6   STO      .LD-3                      RIGHT-SIDE DELTA, AND
    +7   SSM                                       SCALED
    +8   STO      .LD-1                      LEFT-SIDE DELTA (-).
    +9   CLA      ROW
   +10   STO      .LS-3                      ROW BECOMES
   +11   STO      .LS-1                      SIDE-COUNTS.
   +12   XCA
   +13   MPY      .LD-3                      ROW X COLUMN (SCALED)
```

```
    +14  STQ   K.T1
    +15  XCA
    +16  ALS   18
    +17  ADD   K.T1
    +18  SUB   K.A16
    +19  ORA   .LK1-2              LINK(RC,4,RC-2,3)
    +20  STO   .LK2-2
    +21  CLA   .LD-3
    +22  ALS   17
    +23  ADD   .LD-3
    +24  ALS   1
    +25  ADD   .LD-3
    +26  ORA   .LK1-3              LINK(C,2,3C,1).
    +27  STO   .LK2-3
    +28  CLA   K.T1
    +29  ADD   K.A8
    +30  SUB   .LD-3               RC-C+1
    +31  STO   K.T1
    +32  ALS   18
    +33  ADD   K.T1
    +34  SUB   .LD-3
    +35  SUB   .LD-3
    +36  ORA   .LK1-1              LINK(RC-C+1,6,RC-3C+1,5).
    +37  STO   .LK2-1
    +38  AXT   4,1                 INITIALIZE EDGE-COUNT
    +39  TRA   IA20+2
IA20     AXT   **0,1
     +1  TNX   IL00,1,1            FINI IF ALL EDGES BOUND.
     +2  SXA   IA20,1
     +3  CLA   .LD,1
     +4  ALS   18
     +5  ADD   .LD,1               NEW INCREMENT.
     +6  STO   K.T2
     +7  LDQ   .LK2,1              FIRST LINK IN NEXT EDGE.
     +8  CLA   .LS,1               NEW COUNT+2
     +9  PAX   ,1
    +10  XCA
    +11  TXI   IA21,1,-2
IA19     AXT   **0,1
     +1  TNX   IA20,1,1            OUT IF EDGE HAS BEEN BOUND.
     +2  CLA   K.T1
     +3  ADD   K.T2                INCREMENT LINK
IA21     STO   K.T1
     +1  SXA   IA19,1
     +2  PAX   ,2
     +3  PDX   ,4
     +4  TRA   IA10                TO SET LINKS IN NET.
```

IH CHAINS REAL LINES TO HO TABLE, WHICH IS
PRESET FROM UNIFORM DIST BETWEEN (A) AND (B).

```
CALL   SEQU          TSX    IH,4
                            A
                            B
                     RETURN IF STORE EXCEEDED
                     NORMAL RETURN
```

```
          BEGIN    4,7
IH        TXL      *+5,0,0              SUBROUTINE LINKAGE
     +8   CLA      NODES
     +9   ADD      K.A3
    +10   ARS      2                          NR WORDS PER HO-TABLE ENTRY
    +11   STO      HOLINE                     AT FOUR ITEMS PER WORD
    +12   CAL*     2,4
    +13   XCA
    +14   CLA*     1,4
    +15   SSP
    +16   TLQ      *+2
    +17   XCA
    +18   CAS      HOHIGH
    +19   TRA      IHX                        ILLEGAL HO BOUNDS
    +20   TRA      IHX
    +21   STO      IHA
    +22   STQ      IHB
    +23   LXA      LINES,1
    +24   TXI      *+1,1,7
    +25   STZ      HONCNT
    +26   CLA      HOTBLE
IH1       LDQ*     LINEA
     +1   TQP      *+2
     +2   TRA      IH2                        NO ENTRIES FOR IMAG. LINES
     +3   STA*     LINEA
     +4   ADD      HOLINE
     +5   XCA
     +6   CLA      HONCNT
     +7   ADD      NODES
     +8   STO      HONCNT
     +9   XCA
IH2       TXI      *+1,1,-1
     +1   TXH      IH1,1,7
     +2   CAS      UPPER
     +3   TRA      IHX
     +4   TRA      IHX
     +5   CAS      LOWER
     +6   TRA      IH3
     +7   TRA      IHX
     +8   TRA      IHX
IH3       STA      IH7
     +1   SUB      HOTBLE
     +2   PAX      ,1
     +3   SXD      HOTBLE,1
```

```
     +4   CLA     IHB
     +5   TZE     *+3
     +6   SUB     IHA
     +7   TNZ     IH6
     +8   CLA     HCNCNT
     +9   XCA
    +10   MPY     IHA
    +11   STO     HOTOT1
    +12   STQ     HOTOT
    +13   CAL     IHA
    +14   ALS     9
    +15   ORA     IHA
    +16   ALS     9
    +17   ORA     IHA
    +18   ALS     9
    +19   ORA     IHA
    +20   SLW*    IH7
    +21   TIX     *-1,1,1
    +22   ZET     IHB                   IF HPRIME IS ZERC
    +23   NZT     DSPY                  CR ENTERED FROM MCDELS CR BEST-PATH,
    +24   TRA     *+2                         USE BEST-PATH ALGCRITHM.
    +25   TRA     IH6                         ELSE, FILL HCTBLE.
    +26   TSX     BH,4                  PRESET TC BEST VALUES
          RETURN   IH
    +27   TRA     IH+1
IH6       SUB     HOINC
     +1   SLW     IHD
     +2   LXA     NODES,4
IH4       STZ     IHC
     +1   AXT     4,2
IH5       LDQ     RND
     +1   MPY     K.R
     +2   STQ     RND
     +3   MPY     IHD
     +4   ADD     IHB
     +5   ANA     K.M5
     +6   TXL     *+2,4,0
     +7   TXI     *+4,4,-1
     +8   CAL     HOHIGH
     +9   ORS     IHC
    +10   TRA     IH9
    +11   ORS     IHC
    +12   ADD     HCTCT
    +13   STO     HOTCT
    +14   PBT
    +15   TRA     *+4
    +16   CLA     K.A1
    +17   ADD     HOTOT1
    +18   STO     HOTOT1
IH9       CAL     IHC
     +1   TNX     IH7,2,1
     +2   ALS     9
     +3   SLW     IHC
     +4   TRA     IH5
```

```
IH7        SLW     **0,1
      +1   TXH     *+2,4,0
      +2   LXA     NODES,4
      +3   TIX     IH4,1,1
           RETURN   IH
      +4   TRA     IH+1
           RETURN   IH,1
IHX        AXT     1,4
```

BH PRESETS HO TABLE TO 'BEST' VALUES

```
            CALL SEQ   TSX  BH,4
                       RETURN

      BEGIN   1,7
BH    TXL     *+5,0,0                   SUBROUTINE LINKAGE
  +8  CLA     UPDATE              SAVE  HO-TABLE
  +9  STO     BHX                       UPDATING ALGORITHM
 +10  CLA     MIN                 USE MIN AS
 +11  STO     UPDATE                    UPDATER
 +12  LXA     NODES,4             I = NR. OF NODES,N
BH1   SXA     K.O,4               RECYCLE ON I
  +1  LXA     NODES,2
  +2  CLA     HOMAX
  +3  STO*    N2B                       HO(K) = MAXHC , K = U(U)N
  +4  TIX     *-1,2,1
  +5  STZ*    N2C                       HO(I) = C
  +6  STZ     BH.A                ACTION-TOGGLE OFF
  +7  STL     BH.M                MOVE-TOGGLE ON
  +8  TRA     BH3                 INTO INNER LOOP WITH J = I
BH2   STZ     BH.A                BEGIN INNER LOOP, ACTION OFF
  +1  LXA     NODES,4             J = N
BH4   STZ     BH.M                RECYCLE ON J, MOVE IS OFF
BH3   SXA     K.N,4
  +1  CLA*    N2C
  +2  STO     B.HO                B.HO = HO(J)
  +3  AXT     8,2                 K = C,  (COUNT FROM 8-K TO 1)
  +4  PXA     ,4
  +5  ALS     3                   8*J+K IS REL. POS. OF LINE(J,K)
  +6  PAX     ,4                        IN LINE TABLE
  +7  SXD     BH9,4
BH5   CLA*    BUSYC               LINE(J,K)
  +1  TMI     BH6                 IGNORE IF LINE DEAD
  +2  PDX     ,1
  +3  TXL     BH6,1,0             IGNORE IF NO MSG ON LINE
  +4  PXA
  +5  STD*    BUSYC               ERASE MSG
  +6  PXA     ,1
  +7  STO     K.HO                      FOR FUTURE USE BY HOVER
  +8  CAS     B.HC                B.HO = MIN(B.HC, HC CF MSG)
  +9  TRA     *+4
 +10  TRA     *+3
 +11  STO     B.HO
 +12  STL     BH.M                MOVE IS ON IF K.HC DECREASES
 +13  SXA     K.L,4               SAVE LINE NR
 +14  XEC     HOVER               UPDATE HO-TABLE ENTRY
 +15  LXA     K.L,4
BH6   TNX     *+2,2,1             INCREMENT K, TEST
  +1  TXI     BH5,4,1             AND RECYCLE IF K LESS THAN 8
  +2  NZT     BH.M                TEST MOVE TOGGLE
  +3  TRA     BH10                OFF, NODE J SENDS NC MSGS
  +4  CLA     B.HO                      ON
```

```
     +5   LXA     K.N,2
     +6   STO*    N2B                        RESET HC(J) AND INC. B.HC - -
     +7   ADD     HCINC               SENT TO ALL NEIGHBCRS
     +8   ALS     18
     +9   STC     K.HO
BH8       CLA*    BUSYC               IGNORE DEAD LINES
     +1   TMI     BH9
     +2   PAX     ,2                  IMAGE LINE
     +3   CLA     K.HO                MSGS SENT
     +4   STD*    BUSYB               TO NEIGHBCRS
BH9       TXL     *+2,4,**0           FINI IF K = 0
     +1   TXI     BH8,4,-1                  DECREMENT K AND RECYCLE
     +2   STL     BH.A
BH10      LXA     K.N,4                     RECYCLE INNER LCCP CN J
     +1   TIX     BH4,4,1
     +2   ZET     BH.A                TEST ACTION TOGGLE
     +3   TRA     BH2                 CN, REDO INNER LCCP
     +4   LXA     K.O,4                     CFF
     +5   TIX     BH1,4,1             RECYCLF CN I
     +6   LXA     NODES,4             FINI, CLEAR THINGS
     +7   STZ*    N2C
     +8   TIX     *-1,4,1
     +9   CLA     BHX                 RESTORE UPDATING
    +10   STO     UPDATE                    ALGORITHM
          RETURN  BH
    +11   TRA     BH+1
BHX       PZE
```

```
MOVER MOVES MESSAGES THRU NET, UPDATES HQ TABLE
 FOR REROUTED MSGS., STACKS REROUTED MSGS. AT
 APPROPRIATE NODE, PROCESSES DELIVERED MSGS.

 CALL SEQ      XEC   MOVER
               NORMAL RETURN
    SAVES IR1, IR2 AND SENS
```

MA IS A MOVER

```
MA       TSX      *+1,4
         BEGIN     1,7,1
MA0      TXL      *+7,0,0                 SUBROUTINE LINKAGE
    +11  LXA      TP,1
    +12  TXI      *+1,1,1                       NEXT FRAME'S TRAFFIC LIST
    +13  TIX      *+1,1,TPU                     COUNT MODULO TPU
    +14  SXA      TP,1
    +15  CLA      TP,1
    +16  STZ      TP,1
    +17  PDX      ,2
    +18  TRA      MA1+1
MA1      LXD      TP,2
     +1  TXH      MA3,2,0
         RETURN    MA0                  END TRAFFIC
     +2  TRA      MA0+1
MA3      CLA      MSG2,2
     +1  STO      TP
     +2  SXA      K.MSG,2
     +3  PAX      ,1                            EFFERENT LINE NR.
     +4  CLA*     BUSYA
     +5  PAX      ,1                      IMAGE LINE
     +6  STA      K.L
     +7  ARS      3
     +8  PAX      ,4
     +9  STA      K.N                     AFFERENT NODE
    +10  CLA*     NIC                     DROP MSG
    +11  TMI      MA2.1
    +12  LDQ      MSG1,2
    +13  PXA
    +14  LGL      7
    +15  STO      K.O                     ORIGIN
    +16  PXA
    +17  LGL      7
    +18  STO      K.D                     DESTINATION
    +19  PXA
    +20  LGL      4
    +21  STO      K.P                     PRIORITY
    +22  PXA
    +23  LGL      9
    +24  STO      K.ST                    TIME DELAYED INSTACKS
    +25  PXA
    +26  LGL      9
    +27  ADD      HOINC                   HO
    +28  STO      K.HO
    +29  XEC      HOVER                   UPDATE HO TABLE
    +30  CLA      K.N
    +31  SUB      K.D
    +32  TNZ      MA4
    +33  XEC      RECORD                        RECORD ARRIVAL
    +34  TRA      MA1
MA4      CLA      K.HO
     +1  CAS      HOMAX                         DROP MSG IF HO EXCESSIVE
```

```
     +2  TRA   MA2
     +3  NOP
     +4  CAL   MSG1,2              INC HO
     +5  ADD   HOINC1                   IN MSG TABLE
     +6  SLW   MSG1,2
     +7  XEC   DELIVR                   STACK MSG
     +8  TRA   MA1
MA2      CAS   HOHI1
     +1  TRA   *+4
     +2  TRA   *+3
MA2.1    XEC   DROP
     +1  TRA   MA1
     +2  CLA   HOHI1
     +3  STO   K.HO
     +4  TRA   MA1
```

```
   HOVER UPDATES HOTBLE FOR LINE I, MESSAGE J
AFTER INCREMENTING HO IN MSG TABLE

   CALL SEQ. XEC HOVER
             RETURN
         (K.L)  =  I
         (K.MSG)  =  J
        (K.N)  =  CURRENT NODE
         (K.O) = ORIGIN(J)
        (K.HO) = HO(J)
         (K.D) = DEST(J)
         (K.P) = PRIORITY(J)
      SAVES IR1, IR2 AND SENS
```

HA IS A MOVER

```
HA       TSX     *+1,4
         BEGIN    1,7,1
HA0      TXL     *+7,0,0                 SUBROUTINE LINKAGE
   -+11  CLA     K.O
    +12  CAS     K.N
    +13  TRA     *+2
    +14  STZ     K.HO
    +15  SUB     K.A1
    +16  LGR     2
    +17  PAC     ,4                            REL WRD NR OF HO-ITEM WRT ORIGIN
    +18  PXA
    +19  LGL     2
    +20  PAX     ,2                            POSITION IN WORD OF ITEM
    +21  LXA     K.L,1
    +22  CLA*    LINEA
    +23  STA     HA4
HA4      CAL     **0,4                         HO WORD
     +1  XEC     HSH,2                         SHIFT
     +2  ANA     K.M7
     +3  LDQ     K.HO
     +4  SXA     HA1,4
     +5  XEC     UPDATE                  EXECUTE UPDATING ALGORITHM
     +6  TRA     HA2
HA1      AXT     **0,4
     +1  STO     HA.1
     +2  CAL     K.M7
     +3  XEC     HSH1,2
     +4  COM
     +5  ANA*    HA4
     +6  STQ*    HA4
     +7  ORS*    HA4
     +8  CLS     HA.1
     +9  ADD     K.HO
    +10  ADD     HOTOT
    +11  SLW     HOTOT
    +12  TZE     HA5
    +13  TMI     HA3
    +14  PBT
    +15  TRA     HA5
    +16  CLA     HOTOT1
    +17  ADD     K.A1
    +18  STO     HOTOT1
HA5      EQU     HA2
         RETURN   HA0
HA2      TRA     HA0+1
HA3      CLA     HOTOT1
     +1  SUB     K.A1
     +2  SLW     HOTOT1
         RETURN   HA0
     +3  TRA     HA0+1
```

```
              UPDATE EXECUTES HO-TABLE UPDATING ALGORITHM.
              CALL  SEQ   XEC   UPDATE
                          RETURN IF NO CHANGE IN HO-TABLE
                          RETURN IF TABLE TO BE UPDATED
              ENTRY       (ACC) = OLD ENTRY
                          (MQ) = HO OF MESSAGE
              EXIT        (MQ) = NEW ENTRY
              SAVES IR'S AND SENSE



              MIN UPDATES BY MAKING NEW ENTRY
                          = MIN(OLD ENTRY,HO OF MSG)

MIN      TLQ    HA1



               LRN UPDATES HO ENTRIES AS FOLLOWS --
            HO = HO + K1*(MSG HO - HO)  IF (MSG HO - HO) IS NEG.
            HO = HO + K2*(MSG HO - HO)  IF (MSG HO - HO) IS POS.

LRN      TSX    LRN0,4
LRN0     STO    LRN1                      HO ENTRY
     +1  XCA
     +2  SUB    LRN1                      MSG HO - HO
     +3  TZE    1,4                       NO CHANGE
     +4  XCA
     +5  TQP    *+3
     +6  MPY    K1                        LEARNING - USE K1
     +7  TRA    *+2
     +8  MPY    K2                        FORGETTING - USE K2
     +9  ADD    LRN1
    +10  XCA
    +11  CLA    LRN1
    +12  TRA    2,4
```

```
   DELIVR APPENDS MESSAGE J TO TRAFFIC STACK
FOR CURRENT NODE

   CALL SEQU    XEC DELIVR
                RETURN
     INITIAL CONDITIONS AS FOR HOVER
     SAVES IR1, IR2 AND SENS
```

```
            DA IS A DELIVR

       DA STACKS MESSAGES AS FOLLOWS --
            DECREMENT (ADDRESS) OF NODE1(J) POINTS TO FIRST
            ENROUTE (NEW) MSG ON STACK FOR NODE J.
             FOR ALL MSGS IN STACK, HO IS INCREMENTED AND MSG2 IS
                 PREFIX   =   MZE
                 DECR     =   POINTER TO NEXT MSG
                 TAG      =   INCOMING LINK NUMBER
                 ADDRESS    =    TIME IN STACK (ORIGINALLY ZERO)
            A ZERO LINE NR (K.L) INDICATES NEW MSG.
          A ZERO POINTER INDICATES END OF STACK

DA      TSX     DAO,4
        BEGIN    1,7,1
DAO     TXL     *+7,0,0                SUBROUTINE LINKAGE
   +11  LXA     K.MSG,2
DA1     CAL     K.L
    +1  ALS     15
    +2  ANA     K.M3                         STACK-FLAG ON, LINK NR IN TAG
    +3  ORA     K.MZE
    +4  SLW     MSG2,2
    +5  LXA     K.N,1
    +6  CLA*    N1A
    +7  PDX     ,4
    +8  ZET     K.L                    TEST MSG
    +9  TRA     DA4                    ENROUTE MSG
   +10  NZT     CHOKE                  NEW MSG.  ARE WE CHOKING INPUT
   +11  TRA     DA4                    NO - TREAT AS ENROUTE MSG.
   +12  PAX     ,4                     YES - PLACE ON SPECIAL INPUT  STACK
   +13  TXH     DA2,4,0                IS STACK EMPTY - NO
   +14  PXA     ,2                     YES
   +15  STA*    N1A
        RETURN  DAO
   +16  TRA     DAO+1
DA4     TXH     DA2,4,0                      IS STACK EMPTY - NO
    +1  PXD     ,2                     YES
    +2  STD*    N1A
        RETURN  DAO
    +3  TRA     DAO+1
DA2     SXA     DA3,4
    +1  CLA     MSG2,4
    +2  PDX     ,4
    +3  TXH     DA2,4,0
DA3     AXT     **0,4                        PUT MSG ON END OF STACK
    +1  PXD     ,2
    +2  STD     MSG2,4
        RETURN  DAO
    +3  TRA     DAO+1
```

```
RA IS A MESSAGE STACK PROCESSOR.
    FOR EACH MESSAGE ON STACK FOR NODE J, RA CHOOSES
    NON-BUSY EFFERENT LINE HAVING SMALLEST HANDOVER
    NUMBER TO MESSAGE'S DESTINATION. TWO STACKS ARE
    USED - A 'STANDARD' STACK AND A 'NEW-MESSAGE'
    STACK, THE LATTER BEING USED FOR INPUT CHCKING ONLY.
    IF NO LINES ARE AVAILABLE, MSG REMAINS ON STACK. THE
    STANDARD STACK IS PROCESSED BEFORE NEW-MESSAGE STACK,
    AND MESSAGES IN LATTER STACK ARE NEVER DROPPED -
    THUS STACKING AT INPUT IS ENFORCED.


    CALL SEQUENCE    XEC RA
                     RETURN
    SAVES IR'S AND SENSE


RA       TSX    RA0,4
         BEGIN   1,7,1
RA0      TXL    *+7,0,0                  SUBROUTINE LINKAGE
   +11   LXA    LINES,1
   +12   TXI    *+1,1,7
RA03     CLA*   BUSYA                           FOR ALL LINES, ---
    +1   TMI    RA02                            IGNORE DEAD LINES
    +2   PDX    ,2
    +3   TXL    RA02,2,0                 IGNORE FREE LINES
    +4   TXI    *+1,2,-1                 DECREMENT
    +5   PXD    ,2                              BUSY
    +6   STO*   BUSYA                           COUNTER
    +7   TNZ    RA02
    +8   CLA*   LINEA                           FREE LINE IF COUNT BECOMES ZERO
    +9   SLW*   LINEA
RA02     TIX    RA03,1,1
    +1   LXA    NODES,2
RA00     SXA    K.N,2
    +1   STZ    RA.0                            FIRST CYCLE PROCESSES ENROUTE MSGS
    +2   STZ    RA.4                     BUSY-FLAG OFF
    +3   STZ    RA.5                     STACK-USE FLAG OFF
    +4   STZ*   N2B                             CLEAR STACK COUNT
    +5   CLA*   N1B
    +6   STO    MSG2
RA01     AXT    ,1
RA1      SXA    RA.1,1                          LAST MSG NR.
    +1   CLA    MSG2,1
    +2   PDX    ,1
    +3   TXH    RA3,1,0                         TEST END-OF-STACK
RA2      LDQ    MSG2                            END
    +1   RQL    18
    +2   STQ    MSG2
    +3   ZET    RA.0                            TEST CYCLE
    +4   TRA    RAX                             LAST - FINI
    +5   STL    RA.0
    +6   TRA    RA01
RAX      STQ*   N1B
```

```
      +1   TIX     RA00,2,1
           RETURN   RA0
      +2   TRA     RA0+1
RA3        SXA     K.MSG,1                  CURRENT MSG NR.
      +1   ZET     RA.4             ARE ALL LINES BUSY
      +2   TRA     *+4              YES
      +3   TSX     FA,4                     ORDER AVAIL. LINES FOR ROUTING
      +4   LXA     K.AL,4                   NR. OF AVAILABLE LINES
      +5   TXH     RA4,4,0                  ANY AVAIL. LINES
      +6   STL     RA.4             NO,  SET BUSY FLAG
      +7   ZET     RA.0             TEST CYCLE
      +8   TRA     RA9              SECOND CYCLE (INPUT STACK)
RA6        CLA*    N2B                      FIRST CYCLE (STANDARD STACK)
      +1   CAS     STACK                    IS STACK FULL
      +2   TRA     RA7                      YES
      +3   TRA     RA7                      YES
      +4   ADD     K.A1                     NO, INCREMENT COUNT
      +5   STO*    N2B
      +6   CAL     MSG2,1                   INCREMENT TIME-IN-STACK
      +7   ADD     K.A1
      +8   STA     MSG2,1
      +9   STL     RA.5             STACK IN USE
RA9        CAL     MSG1,1
      +1   ADD     STKINC           INC STACK-TIME COUNTER
      +2   SLW     MSG1,1
      +3   TRA     RA1
RA7        CLA     MSG2,1
      +1   LXA     RA.1,1                   DELETE MSG FROM STACK
      +2   STD     MSG2,1
      +3   XEC     DROP
      +4   TRA     RA1
RA4        CLA     AL,4                     ROUTE MSG OVER THIS LINE
      +1   PAX     ,4
      +2   STA     MSG2,1                   LINE NR. TO MSG TABLE
      +3   STP     MSG2,1                   STACK-FLAG OFF
      +4   CAL     K.MZE                    MAKE LINE BUSY
      +5   STP*    LINEC
      +6   CLA     GRAIN
      +7   ADD     K.A1
      +8   ALS     18
      +9   ADD*    BUSYC                    MAKE LINE BUSIER BY GRAIN+1
     +10   STO*    BUSYC
     +11   STD     RA8
     +12   CLA     MSG2,1
     +13   LXA     RA.1,1                   DELETE MSG FROM STACK
     +14   STD     MSG2,1
     +15   CLA*    LINEC                    SET MSG TP
     +16   STD     *+3              MSG ARRIVES NEXT STATION AT TIME
     +17   LXA     TP,1                     T' + TP(LINE) + BUSY(LINE)
RA8        TXI     *+1,1,**0                + GRAIN + 1
      +1   TXI     *+1,1,**0
      +2   TIX     *+1,1,TPU                MODULO TPU
      +3   TSX     CHAIN,4                  STACK MSG
      +4           TP,1                     ON LIST FOR TIME-FRAME-NR
```

```
 +5   LXA    RA.1,1
 +6   NZT    SNAP
 +7   TRA    RA1
 +8   CLA*   N4B              COUNT MESSAGES PASSING
 +9   ADD    K.A1             THRU NODE IF SNAPSHOTS OF FLOW
+10   STO*   N4B              ARE BEING TAKEN.
+11   TRA    RA1
```

```
          FA COMPILES, FOR GIVEN NODE AND MSG, A LIST OF
          AVAILABLE AFF. LINES ORDERED BY INCREASING H.C. NR.
          W.R.T. DESTINATION NODE OF MSG.
          LINES HAVING IDENTICAL H.O. NRS. ARE ORDERED
          RANDOMLY WITHIN THEIR COMMON H.O. NR.

          CALL SEQ.    TSX FA,4
                        RETURN
                (IR1) = MESSAGE NR. = (K.MSG)
                (IR2) = NODE NR. = (K.N)
          EXIT         (K.AL)  =  NR. AVAILABLE LINES
                       (AL(I))  =  I-TH BEST CHOICE OF LINES
               ADD.(AL(I))  =  LINE NR.
               BITS 0-9     =    C
               BITS 10-21   =    ASSOC. HO NR FOR DEST. NODE

        BEGIN   1,7
FA      TXL     *+5,0,0               SUBROUTINE LINKAGE
   +8   STZ     K.AL
   +9   PXA     ,2
  +10   ALS     3
  +11   PAX     ,2                         FIRST LINE NR. FROM SOURCE NODE
  +12   LDQ     MSG1,1
  +13   LGL     7
  +14   PXA
  +15   LGL     7                     DESTINATION NODE
  +16   STO     K.D
  +17   SUB     K.A1
  +18   LGR     2
  +19   STA     FA2                   REL. HO WORD WRT TO DEST NODE
  +20   PXA
  +21   LGL     2
  +22   PAX     ,4                         POSITION IN WORD OF ITEM
  +23   CLA     HSH2,4                SHIFT
  +24   STO     FA3
  +25   AXT     8,1
FA1     CLA*    LINEB                      RECYCLE IF LINE BUSY
   +1   TMI     FA6
   +2   SXA     K.L,2
   +3   STA     *+2
FA2     AXC     **0,4
   +1   CAL     **0,4                      FETCH HO WORD
FA3     NOP                                SHIFT
   +1   ANA     K.M1
   +2   STO     ALHO
   +3   STO     ALWD
   +4   SXA     ALWD,2                     SAVE AL ENTRY  PZE LINE,,H.O.
   +5   TSX     FB,4                       INSERT NEW AL ENTRY
FA6     TXI     *+1,2,1
   +1   TIX     FA1,1,1
        RETURN  FA
   +2   TRA     FA+1
```

FB UPDATES AL LIST, INSERTING ENTRY
FOUND IN (ALWD), ORDERED BY H.O. NR. IN (ALHO).
AL COUNT IN (K.AL)

```
        BEGIN   1,7
FB      TXL     *+5,0,0             SUBROUTINE LINKAGE
    +8  LXA     K.AL,1
    +9  TXH     *+2,1,0
   +10  TXI     FB2,1,1
   +11  LDQ     RND
   +12  MPY     K.R
   +13  STQ     RND
   +14  RQL     12
FB1     CAL     AL,1
    +1  ANA     K.M1
    +2  CAS     ALHO
    +3  TXI     FB2,1,1                   INSERT ABOVE HERE
    +4  TRA     FB3                       EQUALITY - RESOLVE CONFLICT
FB6     TIX     FB1,1,1
FB2     SXD     FB4,1                     SAVE POINTER
    +1  LXA     K.AL,1
    +2  TXI     *+1,1,1                   INCREMENT COUNTER
    +3  SXA     K.AL,1
FB4     TXL     FB5,1,**C
    +1  CLA     AL+1,1
    +2  STO     AL,1                      MOVE UP
    +3  TXI     FB4,1,-1
FB5     CLA     ALWD                      INSERT HERE
    +1  STO     AL,1
        RETURN  FB
    +2  TRA     FB+1
FB3     RQL     1
    +1  TQP     FB6                       FLIP COIN TO RESOLVE CONFLICT
    +2  TXI     FB2,1,1
```

```
          DROP APPENDS DISCARDED TRAFFIC TO DROP-LIST.
          USING MSG TABLE AS STACK. MSG2 BECOMES
             MZE   CURRENT NODE , X , POINTER TO NEXT MSG
          WHERE X = 4 IF MSG DROPPED IN STACK, ELSE 0.
          TOTAL DROPS AND DROPS AT NODE ARE INCREMENTED.


          CALL SEQU    XEC DROP
                       RETURN

            (K.N) = NODE NR
            (K.MSG) = MSG NR
          SAVES IR1, IR2 AND SENSE

DROP    TSX     *+1,4
        BEGIN    1,7
DRO     TXL     *+5,0,0                  SUBROUTINE LINKAGE
     +8 LXA     K.MSG,1
     +9 CAL     MSG2,1
    +10 AXT     ,2
    +11 PBT                                      WAS MSG DROPPED INSTACK
    +12 AXT     1,2                              NO
    +13 ARS     18
    +14 ANA     K.M3                             TAG = 4 IF IN STACK
    +15 SSM                                      PREFIX = MZE
    +16 ORA     K.N                              ADD = NODE
    +17 STO     MSG2,1
    +18 CLA     STKCNT,2
    +19 ADD     K.A1                             COUNT DROPS
    +20 STO     STKCNT,2
    +21 CLA     MSGCNT
    +22 SUB     K.A1
    +23 STO     MSGCNT
    +24 TSX     CHAIN,4
    +25         DR
        RETURN   DRO
    +26 TRA     DRO+1
```

RECORD APPENDS COMPLETED TRAFFIC TO ARRIVAL-LIST

```
CALL  SEQ   XEC   RECORD
            RETURN
      (K.MSG)  =  MSG NR.
      SAVES IR1, IR2 AND SENSE


RECORD  TSX     *+1,4
        BEGIN    1,7
RCO     TXL     *+5,0,0                     SUBROUTINE LINKAGE
    +8  LXA     K.MSG,2
    +9  CLA     K.MZE
   +10  STO     MSG2,2
   +11  CLA     ARRCNT
   +12  ADD     K.A1                               COUNT ARRIVALS
   +13  STO     ARRCNT
   +14  CLA     K.HO
   +15  ADD     ARRHO
   +16  STO     ARRHO
   +17  PBT
   +18  TRA     *+4
   +19  CLA     ARRHO1
   +20  ADD     K.A1
   +21  STO     ARRHO1
   +22  CLA     K.HC
   +23  TZE     *+6
   +24  ARS     3
   +25  PAX     ,1
   +26  CLA*    HODA
   +27  ACC     K.A1
   +28  STO*    HODA
   +29  CLA     K.ST
   +30  PAX     ,1
   +31  TXI     *+1,1,1
   +32  CAS     MAXHO
   +33  NOP
   +34  LXA     MAXHO,1
   +35  CLA*    STDA
   +36  ADD     K.A1
   +37  STO*    STDA
   +38  CLA     MSGCNT
   +39  SUB     K.A1
   +40  STO     MSGCNT
   +41  TSX     CHAIN,4
   +42          AR
        RETURN  RCO
   +43  TRA     RCO+1
```

```
                GN GENERATES A RANDOM NODE NR

                CALL   SEQ    TSX    GN,4
                              RETURN
                EXIT          (ACC) = NODE NR
                       SAVES IR1, IR2 AND SENSE


         BEGIN    1,4
GN       TXL      *+3,0,0                SUBROUTINE LINKAGE
     +4  LDQ      RND
     +5  MPY      K.R
     +6  STQ      RND
     +7  MPY      NODES
     +8  ADD      K.A1
         RETURN   GN
     +9  TRA      GN+1
```

```
                CHAIN ADDS MSG. TO TOP OF LIST CHAINED VIA DECR.
                 HEAD OF LIST IS   PZE  LAST MSG NR ,   , FIRST MSG NR

                CALL SEQ
                            TSX CHAIN,4
                             PZE  LIST-NAME (MAY BE TAGGED)
                            RETURN
                      (K.MSG)  =  MSG NR
                   SAVES IR1, IR2 AND SENS


        BEGIN   2,4
CHAIN   TXL     *+3,0,0                 SUBROUTINE LINKAGE
    +4  LDQ*    1,4
    +5  CLA     K.MSG
    +6  STA*    1,4                           UPDATE LIST'S TAIL
    +7  ALS     18
    +8  XCA
    +9  TNZ     CH1
   +10  XCA                                   EMPTY LIST
   +11  STD*    1,4                           HEAD  =  TAIL
   +12  TRA     CH3
CH1     PAX     ,4
    +1  XCA
    +2  STD     MSG2,4                        CHAIN LAST TO CURRENT
CH3     PDX     ,4
    +1  PXA
    +2  STD     MSG2,4                        CURRENT BECOMES LAST
        RETURN  CHAIN
CH2     TRA     CHAIN+1
```

```
               MG PRESETS MSG STACKS AND GENERATES DESIREC
   NR. OF  MSGS.

               CALL  SEQ    TSX   MG,4
                            RETURN


          BEGIN   1,7
MG        TXL     *+5,0,0                 SUBRCUTINE LINKAGE
      +8  CLA     MSGS
      +9  ALS     1
     +10  PAC     ,1
     +11  STZ     MSGCNT
MG1       SXA     K.MSG,1
      +1  TSX     GM,4
      +2  NOP                                   SYSTEM LCADEC
      +3  TXI     *+1,1,2
      +4  TXH     MG1,1,0
      +5  STZ     AR
      +6  STZ     DR
      +7  STZ     OL
          RETURN  MG
      +8  TRA     MG+1
```

```
          ACTIVE RESETS ACTIVITY LEVELS
          CALL SEQU  TSX ACTIVE,4
                     RETURN

        BEGIN    1,7
ACTIVE  TXL      *+5,0,0                    SUBROUTINE LINKAGE
    +8  LDQ      IMPARE
    +9  FMP      FACTOR
   +10  CAS      MAXACT            IS DESIRED ACTIVITY TOO LARGE
   +11  TRA      *+3               YES
   +12  TRA      *+2               YES
   +13  TRA      *+5               NO
   +14  CLA      MAXACT            USE MAXIMUM
   +15  STO      ALPHA
   +16  CLA      MSGS
   +17  TRA      ACTV1
   +18  STO      ALPHA
   +19  FDP      MAXACT
   +20  PXA
   +21  LLS      8
   +22  SUB      KINT2
   +23  STA      *+2
   +24  PXA
   +25  LRS      **0
   +26  MPY      MSGS
ACTV1   STO      LOAD
    +1  NZT      PRETGL
        RETURN   ACTIVE
    +2  TRA      ACTIVE+1
        XEJECT
    +3  STL      SYSOED
        XPRINT   O,2
    +6  STL      SYSOED
    +9  PZE      PM-RANDOM+ALPHA,,1
   +10  PZE      ALPHA,,1
        RETURN   ACTIVE
   +11  TRA      ACTIVE+1
```

GM GENERATES A RANDOM MSG WHOSE MSG. NR. IS (K.MSG). IF DESIRED MSG LOADING HAS NOT BEEN ATTAINED, MSG ENTERS SYSTEM VIA DELIVR - OTHERWISE MSG IS PLACED ON DROP LIST

```
        CALL SEQU    TSX GM,4
                     RETURNED IF LOADED
                     NORMAL RETURN

          BEGIN   2,7
GM        TXL     *+5,0,0                  SUBROUTINE LINKAGE
GM1       LXA     SOURCE,1                       GENERATE ORIGIN
     +1   TXL     GM3,1,0                        NO SOURCE DIST.
     +2   LXA     SCH,4
     +3   LDQ     RND                            USE SOURCE DIST.
     +4   MPY     K.R
     +5   STQ     RND
     +6   XCA
GM4       TXI     *+1,4,1
     +1   CAS*    SC1A
     +2   TRA     GM3
     +3   NOP
     +4   CAS*    SC2A
     +5   TRA     GM2
     +6   TRA     GM2
     +7   TIX     GM4,1,1
GM3       LDQ     RND                            USE UNIFORM DIST.
     +1   MPY     K.R
     +2   STQ     RND
     +3   MPY     SCH
     +4   ADD     K.A1
     +5   PAX     ,4
GM2       CLA*    N3C
     +1   STZ     K.N
     +2   STA     K.N
     +3   PAX     ,4
     +4   CLA*    N1C
     +5   TMI     GM1                            DISCARD IF DEAD
GM5       LXA     SINK,1                         GENERATE DESTINATION
     +1   TXL     GM7,1,0                        NO SINK DIST.
     +2   LXA     SKH,4
     +3   LDQ     RND                            USE SINK DIST.
     +4   MPY     K.R
     +5   STQ     RND
     +6   XCA
GM8       TXI     *+1,4,1
     +1   CAS*    SK1A
     +2   TRA     GM7
     +3   NOP
     +4   CAS*    SK2A
     +5   TRA     GM6
     +6   TRA     GM6
```

```
     +7   TIX     GM8,1,1
GM7       LDQ     RND
     +1   MPY     K.R
     +2   STQ     RND
     +3   MPY     SKH
     +4   ADD     K.A1
     +5   PAX     ,4
GM6       CLA*    N3C
     +1   ARS     18
     +2   CAS     K.N
     +3   TRA     *+2
     +4   TRA     GM5                   DISCARD IF INCESTUOUS
     +5   STO     K.D
     +6   ALS     29
     +7   XCL
     +8   CLA     K.N
     +9   LGR     7
    +10   LXA     K.MSG,2
    +11   STQ     MSG1,2                ORG, HO=C, DEST, PRIORITY=0
    +12   STZ     MSG2,2
    +13   CLA     MSGCNT                IS SYSTEM LOADED
    +14   CAS     LOAD
    +15   TRA     GM9                   YES
    +16   TRA     GM9                   YES
    +17   ADD     K.A1                  NO
    +18   STO     MSGCNT
    +19   STZ     K.L                   INITIALIZE FOR DELIVR
    +20   XEC     DELIVR                STACK MESSAGE
          RETURN   GM
    +21   TRA     GM+1
GM9       TSX     CHAIN,4
     +1   PZE     DR                    MSG TO DROP LIST
          RETURN   GM,1
     +2   AXT     1,4
```

```
                RM REPLACES DROPPED AND DELIVERED MSGS
                WITH RANDOMELY INITIATED FRESH MSGS.

                CALL SEQU   TSX  RM,4

          BEGIN   1,7
RM        TXL     *+5,0,0                    SUBROUTINE LINKAGE
RM1       LXD     AR,1
     +1   TXL     RM2,1,0
     +2   CLA     MSG2,1
     +3   STD     AR
     +4   SXA     K.MSG,1
     +5   TSX     GM,4
          RETURN  RM                       SYSTEM LOADED
     +6   TRA     RM+1
     +7   TRA     RM1
RM2       LXD     DR,1
     +1   TXL     RM3,1,0
     +2   CLA     MSG2,1
     +3   STD     DR
     +4   SXA     K.MSG,1
     +5   TSX     GM,4
          RETURN  RM                       SYSTEM LOADED
     +6   TRA     RM+1
     +7   TRA     RM2
RM3       STZ     AR
     +1   STZ     DR
     +2   TRA     RM+1
```

```
                SIMUL8 CYCLES THRU SIMULATOR K TIMES AFTER
                INITIALIZING NET.

                CALL SEQU    TSX SIMUL8,4
                             PZE   K OR LOC OF K
                             PZE   L,TAG
                             VFD   H36/K OR LOC OF K
                             RETURN IF NOGC
                             NORMAL RETURN
                WHERE (L,TAG) CONTAINS -
                 LOC OF FIRST SPECIAL LINK,,NR. OF SPECIAL LINKS
                AND L=TAG=0 IMPLIES NO SPECIAL LINKS.


         BEGIN   5,7,1
SIMUL8   TXL     *+7,0,0                      SUBROUTINE LINKAGE
   +11   STZ     ACTTGL
   +12   CLA     1,4
   +13   STO     SIM1
   +14   CLA     3,4
   +15   STO     SIM1+1
   +16   CLA     2,4
   +17   TZE     *+2
   +18   CLA*    2,4
   +19   STO     *+2
   +20   TSX     PRES,4                       INITIALIZE NET, DISPLAY PARAMS
   +21   PZE
   +22   TRA     SIM2                 ERROR
   +23   STZ     INITGL
   +24   TSX     CONT,4
SIM1     PZE
         VFD     H36/
    +1
         RETURN  SIMUL8
    +2   TRA     SIMUL8+1
         RETURN  SIMUL8,1
SIM2     AXT     1,4



                CONTINUE SIMULATION FOR K MORE CYCLES

                CALL  SEQU  TSX CONT,4
                            PZE K OR LOC OF K
                            VFD H36/K OR LOC OF K
   SPACE    2
         BEGIN   3,7,1
CONT     TXL     *+7,0,0                      SUBROUTINE LINKAGE
NGO      EQU     CONT
    -2   AXT     6,1                  TEST IF K OR LOC OF K.
    -1   LDQ     2,4
NGC      PXA
    +1   LGL     6
```

```
    +2  CAS    K.A48                    TEST IF BLANK--
    +3  TRA    NGB                      IMPLIES LOCATICN
    +4  TRA    *+4                      BLANK
    +5  CAS    K.A9                     TEST IF NUMERIC
    +6  TRA    NGB                      IMPLIES LOCATION
    +7  NOP                             NUMERIC--
    +8  TIX    NGC,1,1                  DITTO.
    +9  CLA    1,4                      NUMERIC IMPLIES K
   +10  TRA    *+2
NGB     CLA*   1,4
    +1  SSP
    +2  TZE    NGO+1                          EXIT IF ZERO CYCLE COUNT.
    +3  CAS    K.M2                     TEST IF FLOATING
    +4  TRA    *+3
    +5  TRA    NGD+1
    +6  TRA    NGD+1
    +7  XCA                             CONVERT
    +8  PXA                             TO
    +9  LLS    8                        INTEGER
   +10  SUB    KINT2
   +11  TPL    *+2
   +12  PXA
   +13  STA    NGD
   +14  PXA
NGD     LLS    **0
    +1  NOP
         CALCULATE OUTPUT INTERUPT FREQUENCY FROM K
    +2  STO    NG3
    +3  LDQ    NG3
    +4  ARS    1
    +5  TZE    *+5
    +6  STO    NG3
    +7  ARS    1
    +8  TZE    *+2
    +9  STO    NG3
   +10  XCA
   +11  ZET    INITGL                   IS ENTRY VIA SIMUL8
   +12  TRA    NGA                      NO - CONTINUATION
   +13  STO    NG1                      YES - CLEAR THINGS
   +14  TSX    CLEAR,4
   +15  STZ    COUNTS
   +16  STL    INITGL
   +17  TSX    MG,4                     PRESET MESSAGES - STACK THEM AT NODES
   +18  TRA    *+3
NGA     ADD    NG1
    +1  STO    NG1
    +2  CLA    NG3
    +3  TZE    NGO+1
    +4  ADD    COUNTS
    +5  STO    NG4
    +6  CLA    HMAX                     RESET HMAX IN CASE CHANGED
    +7  ALS    3
    +8  STO    HOMAX
    +9  LDQ    FACTOR
```

```
    +10   FMP    IMPARE
    +11   CAS    ALPHA           HASLOADING BEEN CHANGED
    +12   TRA    *+2             YES
    +13   TRA    *+2             NO
    +14   TSX    ACTIVE,4        YES - CALCULATE NEW ACTIVITY LEVELS
    +15   CLA    LEARN           RESET LEARN AND FORGET
    +16   TSX    FRACT,4
    +17   LDQ    KFULL
    +18   STQ    K1
    +19   CLA    FORGET
    +20   TSX    FRACT,4
    +21   LDQ    KFULL
    +22   STQ    K2
    +23   LDQ    MIN             SET HO-ENTRY
    +24   ZET    ADAPT                 UPDATING ALGORITHM
    +25   LDQ    LRN
    +26   STQ    UPDATE
NG2       SWT    5               EXCESS TIME TEST
     +1   TRA    *+5             OKAY - CONTINUE
     +2   TSX    S1DIST,4
     +3   TSX    S1FLOW,4
     +4   TSX    S1WAIT,4
     +5   TRA    SYSTEM
     +6   XEC    ROUTER          ROUTE MSGS AT NODE
     +7   XEC    MOVER           MOVE MSGS THRU NETWORK
     +8   TSX    RM,4            REPLACE DLVD/DROPPED MSGS WITH FRESH
     +9   CLA    COUNTS          INC. AND TEST CYCLE COUNT
    +10   ADD    K.A1
    +11   STO    COUNTS
    +12   CAS    NG1
    +13   TRA    NG5
    +14   NOP
    +15   NZT    BFTST
    +16   TRA    NG2
    +17   CAS    NG4
    +18   TRA    *+3
    +19   TRA    *+2
    +20   TRA    NG2
    +21   ADD    NG3
    +22   STO    NG4
    +23   TSX    SYSTST,4
    +24   TRA    NG2
NG5       CLA    HOTOT1
     +1   LDQ    HOTOT
     +2   LLS    7
     +3   DVP    HONCNT
     +4   STQ    HOAVG                 MEAN OF HO TABLE
     +5   CLA    DRPCNT
     +6   ADD    STKCNT
     +7   STO    K.T1
     +8   ADD    ARRCNT
     +9   LDQ    K.T1
    +10   STO    K.T1
    +11   MPY    K.A100
```

```
     +12  LLS     10
     +13  DVP     K.T1
     +14  STQ     DROPPC                                PERCENT MSGS DRCPPED
     +15  CLA     ARRHO1
     +16  LDQ     ARRHO
     +17  LLS     7
     +18  DVP     ARRCNT
     +19  STQ     ARRAVG                                MEAN HC CF DELIVERED MSGS.
          RETURN   NGO
     +20  TRA     NGO+1
NG1       OCT     0
```

```
               SKTLU APPENDS NEW SINK TO SINK TABLE

               CALL SEQ     TSX SKTLU,4
                            PZE  NODE NR
                            DEC  PERCENT MSGS TO BE PREEMPTED


          BEGIN    3,7
SKTLU     TXL      *+5,0,0                   SUBROUTINE LINKAGE
     +8   CLA      1,4
     +9   STO      SC1
    +10   TNZ      *+2
    +11   TSX      ERROR,4
    +12   CAS      NODES
    +13   TSX      ERROR,4
    +14   NOP
    +15   CLA      2,4
    +16   TSX      FRACT,4
    +17   TSX      ERROR,4
    +18   STQ      SC2
    +19   LXA      SKH,1
    +20   CLA      SC1
    +21   PAX      ,4
    +22   CAL*     N3C
    +23   ARS      18
    +24   CAS      SC1
    +25   TRA      *-4
    +26   TRA      *+2
    +27   TRA      *-6
    +28   SXD      *+1,1
    +29   TXL      SKT1,4,**0          NEW ENTRY
    +30   SXD      SKT2+1,1            CHANGING OLD ENTRY.
    +31   SXA      SC1,4
    +32   LXA      SC1,2
SKT2      TXI      *+1,2,-1            FIRST,
     +1   TXL      SKT3,2,**0                DELETE
     +2   CLA*     N3C                       OLD
     +3   LDQ*     N3B                       ENTRY
     +4   STO*     N3B
     +5   XCL
     +6   STO*     N3C
     +7   TXI      SKT2,4,-1
SKT3      CLA      NODES               NEXT, DELETE
     +1   SUB      SC1                       FROM
     +2   LXA      SINK,4                    SINK
     +3   SXD      SKT4,4                    TABLE
     +4   PAX      ,6
     +5   TXI      *+1,6,1
     +6   CLA*     SK1C
     +7   SUB*     SK2C
     +8   STO      SC1
     +9   SUB      SKCUM
    +10   SLW      SKCUM
SKT5      TXI      *+1,2,1
```

```
SKT4      TXH     SKT6,2,**C
     +1   CLA*    SK1B
     +2   SUB     SC1
     +3   STO*    SK1C
     +4   CLA*    SK2B
     +5   SUB     SC1
     +6   STO*    SK2C
     +7   TXI     SKT5,4,1
SKT6      ZET     SC2
     +1   TRA     SKT7                      CHANGING ENTRY.
     +2   TXI     *+1,4,-1                  DELETING ENTRY.
     +3   SXA     SINK,4
     +4   TXI     *+1,1,1
     +5   SXA     SKH,1
          RETURN   SKTLU
     +6   TRA     SKTLU+1
SKT1      CLA*    N3C
     +1   LDQ*    N3A
     +2   STO*    N3A
     +3   XCL
     +4   STD*    N3C
     +5   TXI     *+1,1,-1
     +6   SXA     SKH,1
     +7   LXA     SINK,4
     +8   TXI     *+1,4,1
     +9   SXA     SINK,4
SKT7      CLA     SKCUM
     +1   STO*    SK2C
     +2   CLA     SC2
     +3   TOV     *+1
     +4   ACD     SKCUM
     +5   TNO     *+2
     +6   TSX     ERROR,4
     +7   STO     SKCUM
     +8   STO*    SK1C
          RETURN   SKTLU
     +9   TRA     SKTLU+1
```

SCTLU PERFORMS SAME FUNCTION AS SKTLU FR SOURCES

```
          BEGIN    3,7
SCTLU     TXL     *+5,0,0                   SUBROUTINE LINKAGE
     +8   CLA     1,4
     +9   STO     SC1
    +10   TNZ     *+2
    +11   TSX     ERROR,4
    +12   CAS     NODES
    +13   TSX     ERROR,4
    +14   NOP
    +15   CLA     2,4
    +16   TSX     FRACT,4
    +17   TSX     ERROR,4
```

```
     +18  STQ     SC2
     +19  LXA     SCH,1
     +20  CLA     SC1
     +21  PAX     ,4
     +22  CAL*    N3C
     +23  ANA     K.M6
     +24  CAS     SC1
     +25  TRA     *-4
     +26  TRA     *+2
     +27  TRA     *-6
     +28  SXD     *+1,1
     +29  TXL     SCT1,4,**0              NEW ENTRY
     +30  SXD     SCT2+1,1                CHANGING OLD ENTRY.
     +31  SXA     SC1,4
     +32  LXA     SC1,2
SCT2      TXI     *+1,2,-1                FIRST,
      +1  TXL     SCT3,2,**0                    DELETE
      +2  CLA*    N3C                           OLD
      +3  LDQ*    N3B                           ENTRY
      +4  STA*    N3B
      +5  XCL
      +6  STA*    N3C
      +7  TXI     SCT2,4,-1
SCT3      CLA     NODES                   NEXT, DELETE
      +1  SUB     SC1                           FROM
      +2  LXA     SOURCE,4                      SOURCE
      +3  SXD     SCT4,4                        TABLE
      +4  PAX     ,6
      +5  TXI     *+1,6,1
      +6  CLA*    SC1C
      +7  SUB*    SC2C
      +8  STO     SC1
      +9  SUB     SCCUM
     +10  SLW     SCCUM
SCT5      TXI     *+1,2,1
SCT4      TXH     SCT6,2,**0
      +1  CLA*    SC1B
      +2  SUB     SC1
      +3  STO*    SC1C
      +4  CLA*    SC2B
      +5  SUB     SC1
      +6  STO*    SC2C
      +7  TXI     SCT5,4,1
SCT6      ZET     SC2
      +1  TRA     SCT7                    CHANGING ENTRY.
      +2  TXI     *+1,4,-1                DELETING ENTRY.
      +3  SXA     SOURCE,4
      +4  TXI     *+1,1,1
      +5  SXA     SCH,1
          RETURN  SCTLU
      +6  TRA     SCTLU+1
SCT1      CLA*    N3C
      +1  LDQ*    N3A
      +2  STA*    N3A
```

```
     +3   XCL
     +4   STA*    N3C
     +5   TXI     *+1,1,-1
     +6   SXA     SCH,1
     +7   LXA     SOURCE,4
     +8   TXI     *+1,4,1
     +9   SXA     SOURCE,4
SCT7      CLA     SCCUM
     +1   STO*    SC2C
     +2   CLA     SC2
     +3   TOV     *+1
     +4   ADD     SCCUM
     +5   TNO     *+2
     +6   TSX     ERROR,4
     +7   STO     SCCUM
     +8   STO*    SC1C
          RETURN   SCTLU
     +9   TRA     SCTLU+1
```

```
            S1DIST PRINTS DISTRIBUTION OF PATH-LENGTHS
            OF MSGS DELIVERED BY SIMULATOR.


        XEJECT
S1DIST  STL     SYSOED
        XPRINT   J,1
    +3  STL     SYSOED
    +6  PZE     F10.3,,2
        BEGIN    1,4
HOD     TXL     *+3,0,0                        SUBROUTINE LINKAGE
    +4  CLA     HODA
    +5  TSX     PB,4
    +6  TSX     HODX,4
        RETURN   HOD
    +7  TRA     HOD+1


        BEGIN    1,7
HODX    TXL     *+5,0,0                              SUBROUTINE LINKAGE
    +8  LDQ     ARRAVG
    +9  MPY     ARRAVG
   +10  LRS     10
   +11  XCA
   +12  SUB     MU
   +13  SLW     MU
        XPRINT   K,1
   +14  STL     SYSOED
   +17  PZE     F11.1,,12
        XPRINT   A,1
   +18  STL     SYSOED
   +21  PZE     COUNTS,,7
        RETURN   HODX
   +22  TRA     HODX+1


            FLOW DISPLAYS FLOW THRU NETWORK. MESSAGES PASSED THRU
            NODES ARE DISPLAYED ROW-WISE, BEGINNING WITH FIRST ROW.


        BEGIN    1,7
S1FLOW  TXL     *+5,0,0                        SUBROUTINE LINKAGE
FLWO    EQU     S1FLOW
   -16  NZT     SNAP
   -15  TRA     1,4
        XEJECT
   -14  STL     SYSOED
        XPRINT   J,1
   -11  STL     SYSOED
    -8  PZE     F10.6,,2
    -7  CLA     N4A
    -6  STA     FLW1
    -5  LXA     COLUMN,4
    -4  SXD     FLW1,4
```

```
    -3  SXD     FLW2,4
    -2  LXA     NODES,4
    -1  TXL     FLW0+1,4,0
FLW3    CLA     FLW1
    +1  SUB     COLUMN
    +2  STA     FLW1
        XPRINT   H,1
    +3  STL     SYSOED
FLW1    PZE     **0,,**0
FLW2    TIX     FLW3,4,**0
        RETURN   FLW0
    +1  TRA     FLW0+1
```

S1FLOW PRINTS DIST OF DELAYS IN STACKS FOR DLVC MSGS

```
        XEJECT
S1WAIT  STL     SYSOED
        XPRINT   J,1
    +3  STL     SYSOED
    +6  PZE     F10.5,,2
    +7  CLA     STDA


        BEGIN    1,7,1
PB      TXL     *+7,0,0                         SUBROUTINE LINKAGE
   +11  STA     PB1
   +12  STZ     MU
   +13  LXA     MAXHO,1
   +14  CLA*    PB1
   +15  TNZ     PB2
   +16  TIX     *-2,1,1
        RETURN   PB
   +17  TRA     PB+1
PB2     SXD     PB3,1
    +1  AXT     1,1
PB1     LDQ     **0,1
    +1  PXA
    +2  XCA
    +3  DVP     ARRCNT
    +4  TSX     PBX,4
    +5  TXI     *+1,1,1
PB3     TXL     PB1,1,**0
        RETURN   PB
    +1  TRA     PB+1


        BEGIN    1,7
PBX     TXL     *+5,0,0                         SUBROUTINE LINKAGE
    +8  STQ     DT+1
    +9  MPY     K.A100
   +10  RND
   +11  PAX     ,2
```

```
    +12  TXH     *+2,2,0
    +13  TXI     *+1,2,1
    +14  AXT     17,4
    +15  CAL     FILL-6
    +16  TIX     *+3,2,6
    +17  CAL     FILL,2
    +18  AXT     ,2
    +19  SLW     OT.1+18,4
    +20  TIX     *-4,4,1
    +21  SXA     OT,1
         XPRINT   C,1
    +22  STL     SYSOFC
    +25  PZE     OT,,20
    +26  LDQ     OT
    +27  MPY     OT
    +28  MPY     OT+1
    +29  LLS     10
    +30  ADD     MU
    +31  STO     MU
    +32  STL     PTTGL
         RETURN   PBX
    +33  TRA     PBX+1
```

```
            SETHMX SETS HMAX TO VALUE WHICH INSURES LESS THAN
            A FRACTION, K, OF DROP-CUTS.
            CALL SEQU    TSX SETHMX,4
                         PZE  LOC OF K,TAG
                         RETURN IF NOGC
                         NORMAL RETURN
```

```
         BEGIN    3,7,1
SETHMX   TXL     *+7,0,0                  SUBROUTINE LINKAGE
    +11  CLA*    1,4
    +12  XCL
    +13  PXA
    +14  LLS     8
    +15  SUB     KINT2
    +16  STA     *+4
    +17  TZE     *+2
    +18  TPL     2,4
    +19  PXA
    +20  LRS     **0
    +21  STQ     K.T1
    +22  STZ     K.T2
    +23  LXA     MAXHO,4
SET1     LDQ*    HODC
     +1  PXA
     +2  XCA
     +3  DVP     ARRCNT
     +4  XCA
```

```
     +5   ADD      K.T2
     +6   STO      K.T2
     +7   CAS      K.T1
     +8   TXI      SET2,4,1
     +9   TXI      SET2,4,1
    +10   TIX      SET1,4,1
SET2      TXH      *+3,4,63
     +1   SXA      HMAX,4
          RETURN   SETHMX
     +2   TRA      SETHMX+1
          RETURN   SETHMX,1
     +3   AXT      1,4
```

```
               CLEAR ZEROES ALL COUNTS
               CALL SEQU   TSX CLEAR,4
                           RETURN
```

```
          BEGIN    1,4
CLEAR     TXL      *+3,0,0                    SUBROUTINE LINKAGE
     +4   LXA      MAXHO,4
     +5   STZ*     HODC
     +6   STZ*     STDC
     +7   TIX      *-2,4,1
     +8   LXA      NODES,1
     +9   STZ*     N2A
    +10   ZET      SNAP
    +11   STZ*     N4A
    +12   TIX      *-3,1,1
    +13   STZ      DRPCNT
    +14   STZ      STKCNT
    +15   STZ      ARRCNT
    +16   STZ      ARRHO1
    +17   STZ      ARRHO
          RETURN   CLEAR
    +18   TRA      CLEAR+1
```

```
                BESTHO PRESETS HOTABLE TO BEST-PATHS,THEN
                     INVOKES HODIST(BELOW)


                CALL   SEQU   TSX    BESTHO,4
                              PZE    L,TAG
                              RETURN IF NET CAN NOT BE RUN
                              NORMAL RETURN
                WHERE (L,TAG) CONTAINS
                  LOCATION OF FIRST SPECIAL LINKS,,NR OF SPECIAL LINKS
              AND L=TAG=0 IMPLIES NO SPECIAL LINKS.


         BEGIN    3,7,1
BESTHO   TXL      *+7,0,0                      SUBROUTINE LINKAGE
   +11   STL      ACTTGL                 NO ACTIVITY
   +12   CLA      1,4
   +13   TZE      *+2
   +14   CLA*     1,4
   +15   STO      *+2
   +16   TSX      PRESET,4               INIT.NET
   +17   PZE
   +18   TRA      *+3                    NO,GO
   +19   TSX      HODIST,4
         RETURN   BESTHO
   +20   TRA      BESTHO+1
         RETURN   BESTHO,1
   +21   AXT      1,4


                HODIST COMPUTES AND DISPLAYS THE BEST-PATH
                DIST. FOR THE HOTABLE EXTANT.


                CALL   SEQU   TSX    HODIST,4
                              RETURN


         BEGIN    1,7,1
HODIST   TXL      *+7,0,0                      SUBROUTINE LINKAGE
HTD      EQU      HODIST
   -11   TSX      HTF,4                  COMPUTE HOD(I)
   -10   TSX      HTS,4                  COMPUTE HO STAT.
         XEJECT
    -9   STL      SYSOED
         XPRINT   J,1
    -6   STL      SYSOED
    -3   PZE      F10.4,,2
    -2   TSX      HOD,4                  PRINT
         RETURN   HTD
    -1   TRA      HTD+1
```

```
         BEGIN    1,7                COMP. HO-DIST.
HTF      TXL      *+5,0,0                      SUBROUTINE LINKAGE
     +8  TSX      CLEAR,4
     +9  LXA      NODES,1
HTO      SXA      HT1,1
     +1  PXA      ,1
     +2  ALS      3
     +3  PAX      ,1
     +4  SXD      HT2,1
     +5  TXI      *+1,1,7
HT3      CLA*     LINEA
     +1  TMI      HT2
     +2  STA      HT4
     +3  SXA      HT5,1
     +4  AXT      ,1
     +5  LXA      NODES,2
     +6  AXT      4,4
HT4      LDQ      **0,1
     +1  PXA
     +2  LGL      9
     +3  NZT*     N2B
     +4  TRA      *+4
     +5  CAS*     N2B
     +6  TRA      *+3
     +7  TRA      *+2
     +8  STO*     N2B
     +9  TNX      *+3,2,1
    +10  TIX      HT4+1,4,1
    +11  TXI      HT4-1,1,-1
HT5      AXT      **0,1
HT2      TXL      *+2,1,**0
     +1  TXI      HT3,1,-1
     +2  LXA      NODES,1
HT7      CLA*     N2A
     +1  ARS      3
     +2  CAS      MAXHO
     +3  CLA      MAXHO
     +4  NOP
     +5  PAX      ,2
     +6  CLA*     HODB
     +7  ADD      K.A1
     +8  STO*     HODB
HT6      STZ*     N2A
     +1  TIX      HT7,1,1
HT1      AXT      ,1
     +1  TIX      HTO,1,1
         RETURN   HTF
     +2  TRA      HTF+1
```

```
        BEGIN   1,7     COMPUTE AVG AND COUNT OF HD-TABLE
HTS     TXL     *+5,0,0                          SUBROUTINE LINKAGE
    +8  LXA     MAXHO,3
    +9  PXA
   +10  ADD*    HODA
   +11  TIX     *-1,1,1
   +12  STO     ARRCNT
   +13  STZ     ARRAVG
HT8     PXA     ,2
    +1  XCA
    +2  MPY*    HODB
    +3  XCA
    +4  ADD     ARRAVG
    +5  STO     ARRAVG
    +6  TIX     HT8,2,1
    +7  XCA
    +8  PXA
    +9  LLS     10
   +10  DVP     ARRCNT
   +11  STQ     ARRAVG
        RETURN  HTS
   +12  TRA     HTS+1
```

```
          TS COMPUTES AND PRINTS M2'S DIST AS
                FUNCTION OF IMPARE


            CALL  SEQU  TSX   MODEL2,4
                        PZE   L,TAG
                        RETURN IF NET CAN NOT BE RUN
                        NORMAL RETURN
            WHERE (L,TAG) CONTAINS
              LOCATION OF FIRST SPECIAL LINKS,,NR OF SPECIAL LINKS
          AND L=TAG=0 IMPLIES NO SPECIAL LINKS.


        BEGIN   3,7,1
MODEL2  TXL     *+7,0,0                        SUBROUTINE LINKAGE
   +11  STL     ACTTGL                    NO ACTIVITY
   +12  CLA     1,4
   +13  TZE     *+2
   +14  CLA*    1,4
   +15  STO     *+2
   +16  TSX     PRESET,4                  INIT. NET
   +17  PZE
   +18  TRA     TSX                       NO GO
TS      EQU     MODEL2
        XEJECT
   -41  STL     SYSOED
        XPRINT   J,1
   -38  STL     SYSOEC
   -35  PZE     F10.2,,2
        XPRINT   I,1
   -34  STL     SYSOED
   -31  PZE     IMPARE,,1
   -30  CLA     IMPARE
   -29  TSX     FRACT,4
   -28  TRA     TSX
   -27  STQ     K.PI1
   -26  CLA     KFULL
   -25  SUB     K.PI1
   -24  STO     K.PI
   -23  TSX     HTF,4
   -22  TSX     HTS,4
   -21  LXA     MAXHO,1
   -20  NZT*    HODA
   -19  TIX     *-1,1,1
   -18  SXA     TS1,1
   -17  LDQ*    HODA
   -16  PXA
   -15  XCA
   -14  DVP     ARRCNT
   -13  STQ*    HODA
   -12  STZ*    STDA
   -11  TIX     *-6,1,1
   -10  AXT     ,1
    -9  STZ*    STDA
```

```
     -8  STZ     MU
     -7  STZ     ARRCNT
     -6  STZ     ARRAVG
     -5  AXT     1,1
     -4  CLA     K.PI
     -3  STO*    STDA
     -2  PXA     ,1
     -1  STO     TS.T1
TS1      AXT     **0,1
     +1  STZ     K.T2
     +2  LDQ*    HODA
     +3  MPY*    STDA
     +4  ADD     K.T2
     +5  STO     K.T2
     +6  TIX     *-4,1,1
     +7  TZE     TS3
     +8  LXA     TS.T1,1
     +9  STO     TS.T2
    +10  XCA
    +11  TSX     PBX,4
    +12  PXA     ,1
    +13  XCA
    +14  MPY     TS.T2
    +15  LLS     10
    +16  ADD     ARRAVG
    +17  STO     ARRAVG
    +18  TXI     *+1,1,1
    +19  SXA     TS.T1,1
    +20  LXA     TS1,3
TS2      TXI     *+1,2,-1
     +1  LDQ*    STDA
     +2  MPY     K.PI1
     +3  STO     TS.T2
     +4  LDQ*    STDB
     +5  MPY     K.PI
     +6  ADD     TS.T2
     +7  STO*    STDA
     +8  TIX     TS2,1,1
     +9  TRA     TS1
TS3      TSX     HODX,4
         RETURN   TS
     +1  TRA     TS+1
         RETURN   TS,1
TSX      AXT     1,4
```

```
            MD COMPUTES AND PRINTS M1'S DIST AS
               AS FUNCTION OF IMPARE


              CALL   SEQU  TSX    MODEL1,4
                           PZE    L,TAG
                           RETURN IF NET CAN NOT BE RUN
                           NORMAL RETURN
              WHERE (L,TAG) CONTAINS
                LOCATION OF FIRST SPECIAL LINKS,,NR OF SPECIAL LINKS
            AND L=TAG=0 IMPLIES NO SPECIAL LINKS.


         BEGIN    3,7,1
MODEL1   TXL      *+7,0,0                     SUBROUTINE LINKAGE
   +11   STL      ACTTGL               NO ACTIVITY
   +12   CLA      1,4
   +13   TZE      *+2
   +14   CLA*     1,4
   +15   STO      *+2
   +16   TSX      PRESET,4             INIT. NET
   +17   PZE
   +18   TRA      MDX                  NO GO
MD       EQU      MODEL1
   -13   CLA      IMPARE
   -12   TSX      FRACT,4
   -11   TRA      MDX
   -10   STQ      MD.0
    -9   MPY      MD.0
    -8   STO      MD.1
    -7   XCA
    -6   MPY      MD.0
    -5   STO      MD.2
    -4   TSX      HTF,4
    -3   TSX      HTS,4
    -2   AXT      1,1
    -1   STZ*     STDA
MD0      LDQ*     HODA
    +1   PXA
    +2   XCA
    +3   TZE      MD1
    +4   STZ*     HODA
    +5   DVP      ARRCNT
    +6   XCA
    +7   ADD*     STDA
    +8   PBT
    +9   TXI      *+2,1,1
   +10   TRA      MD2
   +11   STO*     STDA
   +12   TRA      MD0
MD1      TXI      MD2,1,-1
MD2      SXA      MD3,1
    +1   TXI      *+1,1,-1
    +2   SXD      MD6,1
```

```
     +3   LXA     MAXHO,1
     +4   SXD     MD8,1
     +5   SXD     MD5+1,1
MD9       AXT     32000,4
     +1   LDQ     RND
MD3       AXT     **0,1
     +1   MPY     K.R
     +2   TRA     *+2
MD4       TXI     *+1,1,-1
     +1   CLA*    STDA
     +2   TLQ     MD4
     +3   AXT     ,2
MD5       TXI     *+1,2,1
     +1   TXH     MD8+1,2,**0
     +2   MPY     K.R
     +3   CLA     MD.0
     +4   TLQ     *+2
     +5   TRA     MD7
     +6   CLA     MD.1
     +7   TLQ     *+2
     +8   TXI     MD5+1,2,1
     +9   CLA     MD.2
    +10   TLQ     MD5+2
MD6       TXH     MD5,1,**C
     +1   TXI     MD5,1,1
MD7       TIX     MD5,1,1
MD8       TXL     *+2,2,**0
     +1   LXA     MAXHO,2
     +2   CLA*    HODB
     +3   ADD     K.A1
     +4   STO*    HODB
     +5   TIX     MD3,4,1
     +6   STQ     RND
     +7   TSX     HTS,4
          XEJECT
     +8   STL     SYSOED
          XPRINT   J,1
    +11   STL     SYSOED
    +14   PZE     F10.1,,2
          XPRINT   I,1
    +15   STL     SYSOED
    +18   PZE     IMPARE,,1
    +19   TSX     HOD,4
          RETURN   MD
    +20   TRA     MD+1
          RETURN   MD,1
MDX       AXT     1,4
```

```
               FRACT CONVERTS FLOATING FRACTION TO BINARY CNE
               CALL SEQU    TSX FRACR,4          (ACC) = FLTNG NR
                            RETURN IF GREATER THAN 1
```

NORMAL RETURN (MQ) = BINARY NR

```
FRACT   XCL                          MAKE POSITIVE.
   +1   PXA
   +2   LLS     8
   +3   SUB     KINT2
   +4   TZE     2,4
   +5   TPL     1,4
   +6   STA     *+2
   +7   PXA
   +8   LRS     **0
   +9   TRA     2,4
```

DEFINE FORMATS

```
        BEGIN    1,4
FORMAT  TXL     *+3,0,0              SUBROUTINE LINKAGE
        XFORM    11
   +4   STL     SYSOED
   +7   PZE     FM1,,4
   +8   PZE     FM2,,3
   +9   PZE     FM3,,4
  +10   PZE     FM4,,4
  +11   PZE     FM5,,3
  +12   PZE     FM6,,3
  +13   PZE     FM7,,3
  +14   PZE     FM8,,3
  +15   PZE     FM9,,5
  +16   PZE     FM10,,5
  +17   PZE     FM11,,2
        RETURN   FORMAT
  +18   TRA     FORMAT+1
```

POOL

```
LIMIT    EQU     32767
TPU      EQU     20
POOL             *,,.POOL-*
K.A1             1
K.A1D1   PZE     1,,1
K.A2             2
K.A3             3
K.A7             7
K.A8             8
K.A9     DEC        9
K.A16    DEC        16
K.A100   DEC        100
K.A48    DEC        48
KINT1    OCT     233000000000
KINT2    OCT     200
KINT3    OCT     1000000000
KFULL    OCT     377777777777
K.MZE    MZE
K.PI     DEC        0
K.PI1    DEC        0
K.R      OCT     11060471625                  MULT FOR RANDOM
K.T1
K.T2
K.T3
K.T4     PZE
K.M1     OCT     77700000
K.M2     OCT     777777777
K.M3             ,7                           TAG
K.M4     OCT     77770000C
K.M5     OCT     777
K.M6     OCT     77777
K.M7     EQU     K.M5
K.D                                           DESTINATION
K.HO                                          HO NUMBER
K.L                                           LINE NR.
K.MSG                                         MESSAGE NR.
K.N                                           NODE NR.
K.O                                           ORIGIN
K.P                                           PRIORITY
K.ST                                    TIME DELAYED IN STACKS
K.AL                                          COUNTER
ACTTGL   PZE     0
         BSS     8
AL       OCT     0
ALHO     OCT     0
ALWD     OCT     0
AR                                            ARRIVAL LIST
BH.A     PZE     0
BH.M     PZE     0
B.HO     PZE     0
BFTST    PZE     0
BOTTOM           **0,2
```

```
B.T.     OCT     1600004C6000
CONN     EQU     WEAVE
CX       EQU     COLUMN
DELIVR   TSX     DAO,4
DSPY     PZE     0
D1       OCT     0
DR                                            DROP LIST
     +1  BCI     1,******
     +2  BCI     1,*****
     +3  BCI     1,****
     +4  BCI     1,***
     +5  BCI     1,**
     +6  BCI     1,*
FILL     BCI     1,
FACTOR   DEC       0
FM1      BCI     4,A,2,10X,4I11,3(F10.4B25)
FM3      BCI     4,C,1X,I10,F12.9B0,A1C8
FM2      BCI     3,B,5X,10(I10)
FM4      BCI     4,D,40X,A6,3H = ,F9.7
FM5      BCI     3,E,40X,A6,3H = ,I5
FM6      BCI     3,F,40X,A6,3H = ,I12
FM7      BCI     3,G,40X,A6,3H = ,A6
FM8      BCI     3,H,2,10X,12I10
FM9      BCI     5,I,40X,9HIMPARE = ,F7.4
FM10     BCI     5,J,40X,A12,13H DISTRIBUTION
F10.1    BCI     2,MODEL-M1
F10.2    BCI     2,MODEL-M2
F10.3    BCI     2,SIMULATOR
F10.4    BCI     2,BEST-PATH
F10.5    BCI     2,STACK-DELAY
F10.6    BCI     2,TRAFFIC-FLOW
FM11     BCI     2,K,2,14X,A72
F11.1    BCI     9,TIME     HO-DROPS  STACK-DROPS DELIVERIES PC-DROPPED DI
     +9  BCI     1,ST-AVG
    +10  BCI     2,  VARIANCE
YES      BCI     1,  YES
NO       BCI     1,  NO
HA.1
HOHIGH   OCT     776
HOHI1    OCT     770
HOINC            8
HOINC1   EQU     HOINC
STKINC   OCT     1000
HOLINE
HOMAX
HOME1
HOVER    TSX     HAO,4
     +1  NOP
     +2  ARS     9
     +3  ARS     18
HSH      ARS     27
     +1  NOP
     +2  LGL     9
     +3  LGL     18
```

```
HSH1    LGL    27
    +1  ALS    15
    +2  ALS    6
    +3  ARS    3
HSH2    ARS    12
IHA
IHB
IHC
IHD     DEC       0
IL.T    DEC       0
INITGL
K1      DEC       0
K2      DEC       0
LRN1    DEC       0
LEFT           **0,2
LOWER   PZE    MSG2
        VFD    15/0,3/2,15/0,3/1
    +1
        VFD    15/0,3/4,15/0,3/3
    +2
        VFD    15/0,3/6,15/0,3/5
    +3
        VFD    15/1,3/0,15/3,3/7
.LK1
    +1  OCT    0,0,0
.LK2    OCT    10,0,-10,C
.LD     OCT    0,0,0,0
.LS     OCT    0
MAXACT  DEC       0
MD.0
MD.1
MD.2
MOVER   TSX    MAO,4
NG3     DEC       0
NG4     DEC       0
OT      DEC       0
    +1  DEC       .2580
CT.1    BCI     ,
   +10  BCI    9,
OL      PZE    0
PRETGL  OCT    0
PTTGL   PZE    0
RA.0    OCT    0
RA.1    OCT    0
RA.2    OCT    0
RA.3    OCT    0
RA.4    OCT    0
RA.5    OCT    0
RND
RX      EQU    ROW
ROUTER  TSX    RAO,4
SC1
SC2
SCCUM   OCT    0
```

```
SCH
SIDES    OCT      34000700CC0
SINK     OCT      0
SKCUM    OCT      0
SKH
SOURCE   OCT      0
SSLIM    DEC         10
TPHIGH   PZE      TPU
TS.T1    DEC         0
TS.T2    DEC         0
UPPER    PZE      LIMIT
UPDATE   TLQ      HA1
COUNTS
DRPCNT                                               NR MSGS DROPPED EN-ROUTE
STKCNT                                               NR. MSGS DROPPED IN STACK
ARRCNT                                               NR OF ARRIVALS
DROPPC                                               PERCENT DROPPED
ARRAVG                                               AVG. HO OF DELIVERED MSGS
MU       PZE      0                                  SECOND MOMENT OF DIST
HOAVG                                                AVG. OF HO TABLE
ARRHC1                                               TOTAL HO OF DELIVERED MSGS
ARRHC
HONCNT                                               NR. OF ENTRIES IN HO-TABLE
HOTOT1                                               SUM OF ENTRIES IN HO-TABLE
HOTOT
CNTTBL   PZE      COUNTS,,*-COUNTS
.POOL             *-POOL
```

PARAMETER LIST

```
PARAM               *,,,PARAM-*-1
CHOKE     PZE       *
SNAP      PZE       *
FLOW      EQU       SNAP
BIND      PZE       0
ADAPT     PZE       0
ALPHA     DEC          0
IMPARE    DEC          0
IMPAIR    EQU       IMPARE
LEARN     DEC          1.0
FORGET    DEC          0
HMAX      DEC          63
MAXHC     EQU       HMAX
WEAVE     DEC          0
ROW       DEC          0
COLUMN    DEC          0
TPMAX     DEC          6
TPMIN     DEC          0
STACK     DEC          8
HPRIME    DEC          0
INITHO    EQU       HPRIME
GRAIN     OCT       2
RANDOM    OCT       110604716255
LINKS     DEC          0
.PARAM
```

BEGINNING OF TABLES AND LISTS

```
TABLES              *,,MSG2-*+1
```

```
        LINK TABLE - CONTAINS IMAGE LINK NRS. AND
                           DISTANCE FROM EFF TO AFF NODE
                     LINKS ARE NUMBERED 0 THRU 7 , VIZ.

                                 7 0 1
                                 6 * 2
                                 5 4 3

        LET L BE A LINE NR., THEN
      L' = EFF. NODE NR. = GRTST. INTEGER (L/8)
      L'' = EFF. LINK NR. = REMAINDER (L/8)
      L1' = AFF. NODE NR. = L' + ADDRESS PART OF LINK(L'')
      L1'' = IMAGE LINK NR. = FIRST 3 BITS OF LINK(L'')


        LINK TABLE
LINK    FOR    **0                         -D
  +1    FVE    **0                         1-D
  +2    SIX    **0                         1
  +3    SVN    **0                         D+1
  +4    PZE    **0                         D
  +5    PON    **0                         -(1-D)
  +6    PTW    **0                         -1
  +7    PTH    **0                         -(D+1)
```

```
             TP LIST CCNTAINS POINTERS TO FIRST AND LAST
          MESSAGES CN TRAFFIC LISTS. TP(I') IS HEAD CF LIST
          FOR TIME I' , WHERE
          I' IS COUNTEC MODULO TPU, THE MAXIMUM PIPE-FILLING TIME.
          TP(I')  =  PZE   LAST MSG NR  ,  ,  FIRST MSG NR
          TP  =  T'  ,  WHERE T IS CURRENT TIME.

TPTBLE  BSS     TPU,0
TP      OCT     0
```

```
                  CONNECTIVITY TABLE
              CTABLE HAS ENTRIES CCRRESPONDING TC ALLCWABLE
           NETWORK CONNECTIVITIES (SEE LINK TABLE)

                      CONN       LINKS USED       CTABLE(CCNN)
                       1          6,2                 10111C11
                       2          6,4,2,C             1C1C1C1C
                       3          7,6,4,3,2,C         CC1CCC1C
                       4          ALL                 0C0CCCC0
              ETC.,  AS REQUIRED

    +1    OCT    314000000000           CONN = 7
    +2    OCT    146000000000           CONN = 6
    +3    OCT    -20000000000           CONN = 5
    +4    OCT    0                      CONN = 4
    +5    OCT    10400000C000           CONN = 3
    +6    OCT    -124000000000          CONN = 2
    +7    OCT    -1660000CCC00          CONN = 1
CTABLE    OCT    -377C0000000C0         ERRCR
                   NOW , NUMBER OF LINES PER NCDE FCR GIVEN CONN.
    +1    DEC      2                    CONN = 1
    +2    DEC      4                    CONN = 2
    +3    DEC      6                      ETC.
    +4    DEC      8
    +5    DEC      6
    +6    DEC      4
    +7    DEC      4
```

```
                    LINE  TABLE
                PREFIX  =  BUSY SIGNAL
              DECR   =  PIPE-FILLING TIME, TP
                 TAG   =     ZERO
                ADDR  =  POINTER TO FIRST HOTABLE ENTRY FOR THIS LINE
                                  (ZERO IF NO ENTRIES)

LINES    DEC        0
LINE1            **0,2
LINEA            **0,1
LINEB            **0,2
LINEC            **0,4
LINTBL           **0,,**0
```

NODE TABLE
THREE PARTS

PART ONE
PREFIX = 4 IF KILLED, ZERC IF ALIVE
DECREMENT = STANDARD-STACK POINTER
TAG = ZERO
ACCRESS = NEW-MESSAGE-STACK POINTER

PART TWO
CURRENT DEPTH OF STANDARD STACK

PART THREE
ADDRESS (DECREMENT) CONTAINS REFERENCE NODE NUMBER
FOR CHOOSING ORIGINS (DESTINATIONS).

PART FOUR EXISTS ONLY IF FLOW SNAPSHOTS ARE BEING TAKEN,
AND THEN CONTAINS NR. OF MESSAGES PASSED THRU NODE.

```
NODES   DEC     0
N1A             **0,1
N1B             **0,2
N1C             **0,4
N2A             **0,1
N2B             **0,2
N2C             **0,4
N3A     PZE     **0,1
N3B     PZE     **0,2
N3C     PZE     **0,4
N4A     PZE     **0,1
N4B     PZE     **0,2
N4C     PZE     **0,4
NODTBL          **0,,**0
```

BUSY TABLE

```
          PREFIX = MZE IMPLIES LINE IS DEAD
          DECR = TIME REMAINING BEFORE LINE IS FREE
          ADDR = POINTER TO IMAGE LINE

BUSY     PZE     **0,,**0
BUSYA    PZE     **0,1
BUSYB    PZE     **0,2
BUSYC    PZE     **0,4
```

```
                 HANDOVER NUMBER TABLE
            INITIALIZED BY ROUTINES INH AND BNH
            PACKED FOUR/WORD, ORDERED,(FOR A GIVEN LINE)
            BY INCREASING NODE NR AND READ FORWARD (SOUTH)


HOTBLE          **0,,**0

             HO DISTRIBUTION TABLE - READ BACKWARD
             LINE I OF HODTBL CONTAINS COUNT OF DELIVERED
             MSGS. WHOSE HO NRS. LIE BETWEEN I AND I+1

HODA            **0,1
HODB            **0,2
HODC            **0,4
HODTBL          **0,,**0

   STDTBL CONTAINS DISTRIBUTION OF DELAYS IN STACKS FOR DLVD MSGS


STDA            **0,1
STDB            **0,2
STDC            **0,4
STDTBL          **0,,**0
```

```
                  SOURCE/SINK TABLE CCNTAINS PERCENTAGE CF
                  CRIGINS/CESTINATICNS TC BE PREEMPTED BY
               SOURCE/SINK NODES.UPPER PERC. BOUND
               IN PART 1, LCWER IN,PART 2.

SCTBLE            **0,0,**0
SC1A              **0,1
SC1B              **0,2
SC1C              **0,4
SC2A              **0,1
SC2B              **0,2
SC2C              **0,4
SKTBLE            **0,0,**C
SK1A              **0,1
SK1B              **0,2
SK1C              **0,4
SK2A              **0,1
SK2B              **0,2
SK2C              **0,4
```

```
            MESSAGE TABLE -- TWC PARTS

                  MSG1  CONTAINS
         7 BITS  -  ORIGIN NODE NR.
         7 BITS  -  DESTINATION NODE NR.
         4 BITS  -  PRIORITY
         9 BITS  -  TIME DELAYED IN STACKS
         9 BITS  -  CURRENT FC

                  MSG2  CONTAINS
         PREFIX = ZERO IF MOVING AND ALIVE, FOUR IF STACKED
         DECREMENT = POINTER TO NEXT MESSAGE ON TRAFFIC LIST
           FOR SAME TIME-FRAME, ZERO IF LAST. SEE TP DESCRIPTION
         TAG = CURRENT LINK NR.
         ADDRESS = CURRENT LINE NR.


LOAD     DEC        0
MSGCNT   DEC        0
MSGS     DEC        0
BASE             MSG2+1
MSGTBL           MSG2,,**C
MSG1
MSG2
         END     NETPGM
```

ACTIVE	0049	SOURCE	C076	CH2	*0047	FM7	C074	IA1	0021
ACTTGL	0073	STCTBL	C084	CH3	C047	FM8	CC74	IA10	0024
ARRAVG	0076	STKCNT	C076	CHAIN	C047	FM9	C074	IA13	*0024
ARRCNT	0076	STKINC	C074	CHCKE	0077	FRACT	C072	IA14	0024
ARRHO1	0076	SYSBFD	C000	CLEAR	0064	GM	C050	IA15	C024
BESTHO	0065	SYSDB1	0000	CONN	0074	GM1 *	CC5C	IA16	C024
BOTTOM	0073	SYSDB2	CCCC	CCNT	0053	GM2	CC50	IA17	0024
CNTTBL	0076	SYSDSK	CC00	CX	0074	GM3	C05C	IA18	C024
COLUMN	0077	SYSIBC	C000	D1	0074	GM4	CC5C	IA19	C025
COUNTS	0076	SYSMIT	C000	DA	*0038	GM5	CC5C	IA2	C021
CTABLE	0080	SYSMCT	C000	DAC	0038	GM6	CC51	IA20	C025
DELIVR	0074	SYSCED	C000	DA1	*0038	GM7	CC51	IA21	C025
DROPPC	0076	SYSORG	C000	DA2	0038	GM8	CC5C	IA3	0021
DRPCNT	0076	SYSSBF	C000	DA3	0038	GM9	C051	IA4	0021
FACTOR	0074	SYSTEM	CCCC	DA4	0038	GN	CC46	IA5	C021
FORGET	0077	SYSTST	C000	DR	C074	GRAIN	CC77	IA6	C021
FORMAT	0072	TABLES	C077	DRC	C044	HA	*CC35	IA7	CC21
HODIST	0065	TPHIGH	C076	DRCP	C044	HA0	CC35	IA8	C021
HODTBL	0084	TPTBLE	*C079	DSPY	C074	HA1	C035	IA9	C021
HOHIGH	0074	UPDATE	C076	DUMP	00C7	HA2	0035	IH	C026
HOINC1	0074	.ENTRY	*C010	ERROR	C007	HA3	0035	IH1	C026
HOLINE	0074	.ERRCR	*C01C	F1C.1	C074	HA4	CC35	IH2	0026
HONCNT	0076	.PARAM	C077	F1C.2	0074	HA5	CC35	IH3	C026
HOTBLE	0084	ACTV1	C049	F1C.3	0074	HA.1	CC74	IH4	C027
HOTOT1	0076	ADAPT	0077	F1C.4	0074	HMAX	CC77	IH5	C027
HPRIME	0077	AL	C073	F1C.5	C074	HCAVG	C076	IH6	0027
IMPAIR	*0077	ALHC	C073	F1C.6	0074	HCD	CC61	IH7	C028
IMPARE	0077	ALPHA	0077	F11.1	C074	HCDA	0084	IH9	0027
INITGL	0075	ALWC	0073	FA	C042	HCDB	C084	IHA	0075
INITHO	0077	AR	C073	FA1	C042	HCDC	CC84	IHB	C075
K.A100	0073	ARRHO	C076	FA2	0042	HCDX	CC61	IHC	C075
K.A101	0073	BASE	0086	FA3	C042	HCHI1	CC74	IHD	0075
LINTBL	0081	BFTST	0073	FA6	C042	HCINC	CC74	IHX	0028
MAXACT	0075	BH	C029	FB	C043	HCMAX	CC74	IL	C023
MODEL1	0070	BH1	C029	FB1	0043	HCME1	C074	IL0	C023
MODEL2	0068	BH10	C030	FB2	0043	HCTOT	C076	IL00	0023
MSGCNT	0086	BH2	C029	FB3	0043	HCVER	CC74	IL1	C023
MSGTBL	0086	BH3	C029	FB4	C043	HSH	CC74	IL2	C023
NETPGM	0007	BH4	C029	FB5	0043	HSH1	C075	IL3	C023
NODTBL	0082	BH5	C029	FB6	C043	HSH2	C075	IL4	C023
PARAMS	0011	BH6	0029	FILL	0074	HTC	CC66	ILX	C024
PARSET	*0008	BH8	C030	FLCW	*0077	HT1	C066	IL.T	C075
PRESET	0011	BH9	C030	FLWO	0061	HT2	C066	INL	C016
PRETGL	0075	BHX	0030	FLW1	C062	HT3	CC66	INLO	0016
RANDOM	0077	BH.A	C073	FLW2	C062	HT4	CC66	INL1	C016
RECORD	0045	BH.M	C073	FLW3	0062	HT5	CC66	INL2	C016
ROUTER	0075	BINC	C077	FM1	0074	HT6	*CC66	INL3	C016
SIDIST	0061	BUSY	0083	FM10	0074	HT7	C066	INS	0017
SIFLOW	0061	BUSYA	C083	FM11	0074	HT8	CC67	INSO	C017
SIWAIT	0062	BUSYB	C083	FM2	0074	HTC	CC65	INSX	C020
SCTBLE	0085	BUSYC	C083	FM3	0074	HTF	C066	K1	C075
SETHMX	0063	B.HC	0073	FM4	0074	HTS	CC67	K2	0075
SIMUL8	0053	B.T.	C074	FM5	C074	IA	C021	KFULL	0073
SKTBLE	0085	CH1	C047	FM6	0074	IA0	C021	KINT1	C073

KINT2	0073	LRN0	0036	NG2	0055	RA6	*0040	SK2B	0085
KINT3	0073	LRN1	0075	NG3	0075	RA7	0040	SK2C	0085
K.A1	0073	MA	*0032	NG4	0075	RA8	0040	SKCUM	0076
K.A16	0073	MA0	0032	NG5	0055	RA9	0040	SKH	0076
K.A2	*0073	MA1	0032	NGA	0054	RAX	0039	SKT1	0058
K.A3	0073	MA2	0033	NGB	0054	RA.0	0075	SKT2	0057
K.A48	0073	MA2.1	0033	NGC	0053	RA.1	0075	SKT3	0057
K.A7	0073	MA3	0032	NGD	0054	RA.2	*0075	SKT4	0058
K.A8	0073	MA4	0032	NO	0074	RA.3	*0075	SKT5	0057
K.A9	0073	MAXHO	0077	NODES	0082	RA.4	0075	SKT6	0058
K.AL	0073	MD	0070	OL	0075	RA.5	0075	SKT7	0058
K.D	0073	MD0	0070	OT	0075	RCO	0045	SKTLU	0057
K.HO	0073	MD1	0070	OT.1	0075	RM	0052	SNAP	0077
K.L	0073	MD2	0070	PARAM	0077	RM1	0052	SSLIM	0076
K.M1	0073	MD3	0071	PB	0062	RM2	0052	STACK	0077
K.M2	0073	MD4	0071	PB1	0062	RM3	0052	STDA	0084
K.M3	0073	MD5	0071	PB2	0062	RND	0075	STDB	0084
K.M4	*0073	MD6	0071	PB3	0062	RCW	0077	STDC	0084
K.M5	0073	MD7	0071	PBX	0062	RX	0075	TP	0079
K.M6	0073	MD8	0071	PM	0015	SC1	0075	TPMAX	0077
K.M7	0073	MD9	*0071	PMC	0014	SC1A	0085	TPMIN	0077
K.MSG	0073	MDX	0071	PCCL	0073	SC1B	0085	TPU	0073
K.MZE	0073	MD.0	0075	PRO	0011	SC1C	0085	TS	0068
K.N	0073	MD.1	0075	PR1.1	0012	SC2	0075	TS1	0069
K.O	0073	MD.2	0075	PR3	0012	SC2A	0085	TS2	0069
K.P	0073	MG	0048	PR4	0014	SC2B	0085	TS3	0069
K.PI	0073	MG1	0048	PR5	0013	SC2C	0085	TSX	0069
K.PI1	0073	MIN	0036	PR6	0012	SCCLM	0075	TS.T1	0076
K.R	0073	MOVER	0075	PR7	0012	SCH	0076	TS.T2	0076
K.ST	0073	MSG1	0086	PRES	0011	SCT1	0059	UPPER	0076
K.T1	0073	MSG2	0086	PRS1	0008	SCT2	0059	WEAVE	0077
K.T2	0073	MSGS	0086	PRS2	0008	SCT3	0059	YES	*0074
K.T3	0073	MU	0076	PRS3	0008	SCT4	0059	.EN1	0010
K.T4	0073	N1A	0082	PRS4	0009	SCT5	0059	.EN2	0010
LEARN	0077	N1B	0082	PRSX	0009	SCT6	0059	.EN3	0010
LEFT	0075	N1C	0082	PRX	0014	SCT7	0060	.EN4	0010
LIMIT	0073	N2A	0082	PTTGL	0075	SCTLU	0058	.EN5	0010
LINE1	0081	N2B	0082	RA	*0039	SET1	0063	.ENX	0010
LINEA	0081	N2C	0082	RA0	0039	SET2	0064	.EXIT	0010
LINEB	0081	N3A	0082	RACC	0039	SIDES	0076	.LD	0075
LINEC	0081	N3B	0082	RA01	0039	SIM1	0053	.LK1	0075
LINES	0081	N3C	0082	RA02	0039	SIM2	0053	.LK2	0075
LINK	0078	N4A	0082	RA03	0039	SINK	0076	.LS	0075
LINKS	0077	N4B	0082	RA1	0039	SK1A	0085	.PCCL	0076
LOAD	0086	N4C	0082	RA2	*0039	SK1B	0085		
LOWER	0075	NG0	0053	RA3	0040	SK1C	0085		
LRN	0036	NG1	0056	RA4	0040	SK2A	0085		

ON DISTRIBUTED COMMUNICATIONS:

List of Publications in the Series

I. Introduction to Distributed Communications Networks, Paul Baran, RM-3420-PR.

Introduces the system concept and outlines the requirements for and design considerations of the distributed digital data communications network. Considers especially the use of redundancy as a means of withstanding heavy enemy attacks. A general understanding of the proposal may be obtained by reading this volume and Vol. XI.

II. Digital Simulation of Hot-Potato Routing in a Broadband Distributed Communications Network, Sharla P. Boehm and Paul Baran, RM-3103-PR.

Describes a computer simulation of the message routing scheme proposed. The basic routing doctrine permitted a network to suffer a large number of breaks, then reconstitute itself by rapidly relearning to make best use of the surviving links.

III. Determination of Path-Lengths in a Distributed Network, J. W. Smith, RM-3578-PR.

Continues model simulation reported in Vol. II. The program was rewritten in a more powerful computer language allowing examination of larger networks. Modification of the routing doctrine by intermittently reducing the input data rate of local traffic reduced to a low level the number of message blocks taking excessively long paths. The level was so low that a deterministic equation was required in lieu of Monte Carlo to examine the now rare event of a long message block path. The results of both the simulation and the equation agreed in the area of overlapping validity.

IV. Priority, Precedence, and Overload, Paul Baran, RM-3638-PR.

The creation of dynamic or flexible priority and precedence structures within a communication system handling a mixture of traffic with different data rate, urgency, and importance levels is discussed. The goal chosen is optimum utilization of the communications resource within a seriously degraded and overloaded network.

V. History, Alternative Approaches, and Comparisons, Paul Baran, RM-3097-PR.

A background paper acknowledging the efforts of people in many fields working toward the development of large communications systems where system reliability and survivability are mandatory. A consideration of terminology is designed to acquaint the reader with the diverse, sometimes conflicting, definitions used. The evolution of the distributed network is traced, and a number of earlier hardware proposals are outlined.

VI. Mini-Cost Microwave, Paul Baran, RM-3762-PR.

The technical feasibility of constructing an extremely low-cost, all-digital, X- or K_u-band microwave relay system, operating at a multi-megabit per second data rate, is examined. The use of newly developed varactor multipliers permits the design of a miniature, all-solid-state microwave repeater powered by a thermoelectric converter burning L-P fuel.

VII. Tentative Engineering Specifications and Preliminary Design for a High-Data-Rate Distributed Network Switching Node, Paul Baran, RM-3763-PR.

High-speed, or "hot-potato," store-and-forward message block relaying forms the heart of the proposed information transmission system. The Switching Nodes are the units in which the complex processing takes place. The node is described in sufficient engineering detail to estimate the components required. Timing calculations, together with a projected implementation scheme, provide a strong foundation for the belief that the construction and use of the node is practical.

VIII. The Multiplexing Station, Paul Baran, RM-3764-PR.

A description of the Multiplexing Stations which connect subscribers to the Switching Nodes. The presentation is in engineering detail, demonstrating how the network will simultaneously process traffic from up to 1024 separate users sending a mixture of start-stop teletypewriter, digital voice, and other synchronous signals at various rates.

IX. Security, Secrecy, and Tamper-Free Considerations, Paul Baran, RM-3765-PR.

Considers the security aspects of a system of the type proposed, in which secrecy is of paramount importance. Describes the safeguards to be built into the network, and evaluates the premise that the existence of "spies" within the supposedly secure system must be anticipated. Security provisions are based on the belief that protection is best obtained by raising the "price" of espied information to a level which becomes excessive. The treatment of the subject is itself unclassified.

X. Cost Estimate, Paul Baran, RM-3766-PR.

A detailed cost estimate for the entire proposed system, based on an arbitrary network configuration of 400 Switching Nodes, servicing 100,000 simultaneous users via 200 Multiplexing Stations. Assuming a usable life of ten years, all costs, including operating costs, are estimated at about $60,000,000 per year.

XI. Summary Overview, Paul Baran, RM-3767-PR.

Summarizes the system proposal, highlighting the more important features. Considers the particular advantages of the distributed network, and comments on disadvantages. An outline is given of the manner in which future research aimed at an actual implementation of the network might be conducted. Together with the introductory volume, it provides a general description of the entire system concept.