U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

AD-A016 577

THE PRACTICAL IMPACT OF RECENT COMPUTER ADVANCES ON THE ANALYSIS AND DESIGN OF LARGE SCALE NETWORKS

NETWORK ANALYSIS CORPORATION

PREPARED FOR
ADVANCED RESEARCH PROJECTS AGENCY

JUNE 1974

This information product distributed by

NTIS U.S. DEPARTMENT OF COMMERCE
National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Unclassified

Security Classification

18K

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Network Analysis Corporation
Beechwood, Old Tappan Road
Glen Cove, New York 11542

2a. REPORT SECURITY CLASSIFICATION: Unclassified

2b. GROUP

3. REPORT TITLE

Third Semiannual Technical Report, June 1974, for the Project "The Practical Impact of the Recent Computer Advances on the Analysis and Design of Large Scale Networks"

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Third Semiannual Report, May 1973

5. AUTHOR(S) (First name, middle initial, last name)

Network Analysis Corporation

6. REPORT DATE: June 1974

7a. TOTAL NO. OF PAGES: 384

7b. NO. OF REFS: 122

8a. CONTRACT OR GRANT NO.: DAHC15-73-C-0135

b. PROJECT NO.

c. ARPA Order No. 2286

d.

9a. ORIGINATOR'S REPORT NUMBER(S): Semiannual Report 3B

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report): --

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES: None

12. SPONSORING MILITARY ACTIVITY: Advanced Research Projects Agency, Department of Defense

13. ABSTRACT

New results on the following major questions are reported: A network design algorithm using cut saturation with multicapacity links for distributed packet switched networks; An algorithm for optimizating terminal processor locations in terminal oriented systems; A branch and bound approach to network design; A system for large scale computations and the impact of interactive graphics on network design; Simulation of Packet Radio Networks; network capacity tradeoffs and channel configurations for packet radio system, and algorithms for repeaters location for line of sight area coverage.

PRICES SUBJECT TO CHANGE

14. KEY WORDS

Computer networks, throughput, cost, reliability, packet switching, simulation, packet radio, broadcast radio

DD FORM 1 NOV 65 1473 (PAGE 1) PLATE NO. 21856

S/N 0102-014-6600

Unclassified

Security Classification

Third Semiannual Technical Report

June 1974

For the Project

The Practical Impact of Recent Computer Advances on the Analysis and Design of Large Scale Networks

Principal Investigator
and Project Manager:

Howard Frank (516) 671-9580

ARPA Order No. 2286

Contractor: Network Analysis Corporation

Contract No. DAHC15-73-C-0135

Effective Date: 13 October 1972

Expiration Date: 12 October 1974

Sponsored by

Advanced Research Projects Agency
Department of Defense

SUMMARY

Technical Problem

Network Analysis Corporation's contract with the Advanced Research Projects agency has the following objectives:

- To determine the most economical and reliable configurations to meet growth requirements in the ARPANET.
- To study the properties of packet switched computer communications networks.
- To develop techniques for the analysis and design of large scale networks.
- To determine the cost/throughput/reliability characteristics of large packet-switched networks for application to Defense Department computer communication requirements.
- To apply recent computer advances, such as interactive display devices and distributed computing, to the analysis and design of large scale networks.

General Methodology

The approach to the solution of these problems has been the simultaneous,

- study of fundamental network analysis and design issues,
- development of efficient algorithms for large scale network analysis and design,
- development of an interactive distributed display and computational system to deal with large-scale problems,
- application of the new analysis and design tools to study cost and performance tradeoffs for large systems.

Technical Results

In this report, the following major accomplishments are discussed:

- The third phase of a study of terminal oriented network cost and performance was completed. This phase was directed at developing techniques for optimizing TIP and ANTS like nodes in large networks.

- The new large network design technique, based on "cut-saturation", reported in Semiannual Report 2 was extended to networks with multiple capacity options.
- A method for determining absolute lower bounds on cost for packet switched networks, given traffic and delay requirements, was developed.
- The second phase of an interactive network data handling system has been completed for an IMLAC display editing system for large network graphics.
- The second phase of a detailed, event oriented simulation model to develop flow control and routing algorithms was completed for the packet radio system. This system is now operating on the ARPANET with PDP 10 editing and 360/91 RJS for extensive computations.
- Major progress was made in the development of labeling and initialization schemes for the packet radio system.
- A model for the set covering/repeator location problem was developed and tested.
- An extensive study of flow, delay and throughput in packet radio networks was completed.

Department of Defense Implications

The Department of Defense has vital need for highly reliable and economical communications. The results described in this reporting period reinforce conclusions of earlier periods about the validity of packet switching for massive DOD data communications problems. A major portion of the cost of implementing this technology will occur in providing local access to the networks. Hence, the development of local and regional communication techniques must be given high priority.

Implications for Further Research

Further research must continue to develop tools for the study of large network problems. These tools must be used to investigate tradeoffs between terminal and computer density, traffic variations, the effects of improved local access schemes such as packet radio, the use of domestic satellites in broadcast mode for backbone networks, and the effect of link and computer hardware variations in reliability on overall network performance. The potential of these networks to the DOD establishes a high priority for these studies.

TABLE OF CONTENTS

PART 1: NETWORK TECHNIQUES AND PROPERTIES

Chapter 1: A Cut Saturation Algorithm for Topological Design of Packet Switched Communications Networks—Part 2

	PAGE
I. Introduction	1.1
A. Non-Uniform Requirements	1.2
B. Large Capacity Gaps	1.2
C. Large Scale Economics	1.2
D. Inadequate Reliability	1.3
E. Network Growth	1.3
II. Background	1.5
III. State-of-the-Art	1.7
IV. The CS Method for Multiple Capacity Options	1.9
A. General	1.9
B. Determination of the Saturated Cut	1.9
C. Link Insertion or Upgrading	1.11
D. Link Reduction and Removal	1.12
E. Starting Topologies	1.13
F. The Algorithm	1.14
V. An Application	1.15
VI. Directions for Future Research	1.23
References	1.26

Chapter 2: Terminal Oriented Network Cost and Performance--Part 3 The Access Facility Location Problem

I. Introduction	2.1
II. Basic Problem Statement	2.6
III. Related Problems and Solution Techniques	2.10
A. Warehouse Location Problems	2.10
B. Clustering Problems	2.11
C. Partitioning Problems	2.12

TABLE OF CONTENTS

PAGE

IV. Previous Approaches 2.13

V. Description of General Approach 2.16

A. Simplification by Clustering 2.17
B. Partitioning 2.18
C. Local Optimization 2.19
D. Line Layout 2.20

VI. The Center-of-Mass Algorithm 2.25

COM Algorithm 2.30

Step 0: (Initialization) 2.32
Step 1: (Merge) 2.33
Step 2: (Update) 2.34
Step 3: (Evaluate Each Site) 2.35
Step 4: (Select Best) 2.36
Step 5: (Finish) 2.37
Alternate Step 2: (Alternate Update) 2.38

VII. Performance Results 2.40

A. Uniformly Randomly Distributed Nodes 2.42
B. Randomly Distributed Based on Population 2.43
C. Computation Time 2.44

VIII. Generalizations and Extensions 2.65

A. Line Constraints 2.65
B. TACOM Constraints 2.66
C. Possible TACOM Sites 2.68
D. Multiple Capacity 2.71
E. Staging 2.71
F. RESCOP Variations 2.72
G. Very Large Networks and Further Time Reductions 2.74

IX. Conclusions 2.78

References 2.79

TABLE OF CONTENTS

PAGE

Chapter 3: A Branch and Bound Approach to Topological Network Design—Part 1

I. Introduction 3.1

II. The Lower Bound 3.4

III. The Branch and Bound Concept 3.11

IV. Computational Considerations 3.14

V. Implications for Future Research 3.16

References 3.18

Chapter 4: A System for Large Scale Network Computations—Part 2

Network Reliability Analyzer and Network Routing Program 4.8

References 4.15

Chapter 5: Impact of Interactive Graphics on Network Design

I. Introduction 5.1

II. Functional Specifications of the Interactive Network Analysis Program 5.2

A. Node Properties 5.3
B. Link Properties 5.3

III. Turnaround Delay Performance of Interactive Programs 5.10

A. General 5.10
B. Routing Program Turnaround Delay 5.10
C. Performance of Interactive Reliability Analysis 5.11

IV. Interactive Versus Batch Network Design 5.17

V. Impact of Interactive Graphics on Network Design 5.20

VI. Example 5.22

VII. Future Research 5.27

References 5.28

TABLE OF CONTENTS

PAGE

Chapter 6: Computational Complexity of Network Connectivity Algorithms

I. Introduction 6.1

II. The Lattice of Partitions 6.5

III. Structure of Random Graphs 6.9

IV. Descriptive Combinatorial Quantities 6.14

V. Calculations 6.18

VI. A New Measure of Connectivity 6.24

VII. A Graph and its Complement 6.29

VIII. Application to Reliability and Other Areas 6.31

IX. Graphs with Minimal and Maximal Connectivity 6.35

X. Connectivity of Bipartite Graphs 6.37

XI. Descriptive Combinatorial Quantities in a Graph 6.40

References 6.42

TABLE OF CONTENTS

PART 2: PACKET RADIO

PAGE

Chapter 7: Simulation of Packet Switched Communication Networks

I. Introduction 7.1

II. Objectives of Simulator 7.3

A. Routing Algorithms 7.3
B. Protocols 7.3
C. Identify System Bottlenecks 7.4
D. Flow Control 7.4
E. Software Transfer 7.4
F. Trade-Offs 7.5
G. Stand-by Network 7.5

III. General Structure of Simulator 7.6

A. Performance Measures 7.7
B. Network Measurements 7.7
C. Performance of Communication Devices 7.8
D. Trace Information 7.8

IV. Packet Radio Network Simulator 7.10

V. Data Structures of the Simulator 7.12

A. Event Structure 7.12
B. Active Message Structure 7.13
C. Active Packet Structure 7.14
D. Repeater-Station Structure 7.14
E. Data Collection Tables 7.14
F. An Outline of the Use of Data Structures 7.15

VI. The Communication System Simulated 7.17

A. General System Description 7.17
B. Routing Algorithms 7.18
C. Acknowledgement Schemes 7.20
D. Performance Measures 7.23

TABLE OF CONTENTS

PAGE

VI. The Communication System Simulated 7.17

A. General System Description 7.17
B. Routing Algorithms 7.18
C. Acknowledgement Schemes 7.20
D. Performance Measures 7.23

VII. Logical Operation of Devices 7.26

A. States of Devices 7.26
B. Terminal 7.27
C. Repeater 7.28
D. Station 7.30
E. Flow Diagrams of Devices 7.30

VIII. Subroutines of the Simulator 7.44

A. Data Structure and Management Subroutines 7.44
B. Communication and Device Subroutines 7.45
C. Summary of Acronyms 7.47

IX. Observation of Traffic Flow in the Packet Radio Network 7.48

X. The Tradeoff Between Transmission Range of Devices and Network Interference 7.53

XI. Preliminary Results of Maximum Throughput, Loss, and Delay of CCSDR and CCTDR Systems 7.64

XII. Future Development of the Packet Radio Simulator 7.69

References 7.72

Chapter 8: Packet Radio System Considerations—Network Capacity Tradeoffs

I. Introduction 8.1

II. Problem Description 8.6

III. Transmission from Terminals to Station 8.11

A. Complete Interference System, I = m 8.14
B. The Critical Hop 8.16
C. Numbers of Repeaters and Directional Antennas at Repeaters 8.16

TABLE OF CONTENTS

PAGE

IV. Transmission from Station to Terminals 8.20

V. Conclusions 8.31

A. Transmission to Station 8.31
B. Transmission from Station 8.32

References 8.34

Chapter 9: Packet Radio System Considerations—Channel Configuration

I. Introduction 9.1

II. Preliminary Analysis 9.3

A. The Slotted ALOHA Channel 9.3
B. The Single Source Case 9.3
C. Mixed Sources on a Common Channel 9.4

III. Maximum Utilization of the Split and Common Channels 9.5

IV. Delay Considerations 9.9

V. Conclusions 9.14

References 9.15

Chapter 10: Area Coverage by Line-of-Sight Radio

I. Problem Formulation 10.1

II. Computational Techniques 10.5

III. The "Flat Terrain" Model. A Proposed Solution 10.27

References 10.32

LIST OF FIGURES

CHAPTER 1

		PAGE
1.	Collapsable Chain	1.10
2.	Initial 50 KB/S Configuration	1.17
3.	Throughput vs. Cost Performance	1.18
4.	Topology Obtained with the C-S Method	1.19
5.	Topology Obtained with the C-S Method	1.20
6.	Topology Obtained with the C-S Method	1.21
7.	Lower Bound to Actual Link Cost	1.22

CHAPTER 2

1.	Access Alternatives for ARPANET	2.4
2.	General Problem Formulation	2.5
3.	Basic Network Topology Problem	2.9
4.	Flow Chart of Approach	2.21
5.	Simplification by Clustering	2.23
6.	Partition by Add Algorithm	2.23
7.	Local Optimization	2.24
8.	Line Layout	2.24
9.	New Nearest Neighbor	2.39
10.	Average of Tree and Direct Cost Algorithm Applied to 400 Nodes Distributed in a Random Manner Based on Population	2.47
11.	50 Nodes Randomly Distributed on their MST	2.48
12.	100 Nodes Randomly Distributed and their MST	2.49
13.	200 Nodes Randomly Distributed on their MST	2.50
14.	400 Nodes Randomly Distributed and their MST	2.51
15.	Typical COM Design Experiment #1; 100 Nodes	2.52
16.	Typical COM Design Experiment #2; 100 Nodes	2.53
17.	Typical COM Design Experiment #3; 400 Nodes	2.54
18.	COM Algorithm Cost Performance	2.55
19.	COM Algorithm Applied to 400 Nodes Distributed in a Random Manner Based on Population	2.56
20.	COM Algorithm Applied to 400 Nodes Distributed in a Random Manner Based on Population and with no Constraint on TACOM Capacity	2.57
21.	ATD Algorithm Applied to 400 Nodes Distributed in a Random Manner Based on Population and with no Constraint on TACOM Capacity	2.58
22.	Execution Time of COM and ATD Algorithms	2.59
23.	Execution Times of the COM and ATD Algorithms (Log-Log Scales)	2.60
24.	TACOM Staging	2.77

LIST OF FIGURES

CHAPTER 3

		PAGE
1.	Cost-Capacity Function for an Assigned Link	3.9
2.	Cost vs. Capacity for Undefined Link	3.10
3.	Branch and Bound Tree	3.13

CHAPTER 4

1A.	June 1974 ARPANET	4.3
1B:	Enlarged Network Detail	4.4
1C:	Enlarged Network Detail	4.5
2A:	Reliability Curve	4.6
2B:	Throughput-Delay Curve	4.7

CHAPTER 5

1.	42 Node ARPANET Configuration, all 50 KB/S Lines vs. Increase of 1 Cross Country Chain to 23- KB/S	5.7
2.	42 Node ARPANET Configuration 100, 500, 1,000 Bit Packets	5.8
3.	100 Sample Reliability Runs, Only Links Fail, Both Nodes and Links Fail	5.9
4.	10-Node Network CPU Time—3 Sec.	5.14
5.	26-Node Network CPU Time—11 Sec.	5.15
6.	42-Node Network CPU Time—24 Sec.	5.16
7.	Annual Costs vs. Throughput	5.26

A CUT SATURATION ALGORITHM FOR TOPOLOGICAL DESIGN OF PACKET SWITCHED COMMUNICATION NETWORKS

PART 2

I. INTRODUCTION

The cut-saturation (C S) technique for the topological design of packet-switching networks was first presented in Semi-annual Report #2 [1] and was shown to be computationally much more effective than other available techniques. The algorithm described in Ref. [1] applies to the design of networks in which all communications circuits have the same preassigned line speed (e.g. all 50 kb/s line capacities etc.)

This report extends the C S technique to the topological network design with multiple line capacity options.

The topological design problem here addressed can be very generally formulated as follows:

1. Given the node locations, the traffic requirements between such locations and the line capacity options available;

2. Minimize the total line and modem cost;

3. Such that the average delay and connectivity requirements are satisfied.

The use of multiple capacity options can provide substantial cost savings in several situations. In particular, it is desirable to consider topological implementations which include different line speeds in the following cases:

A. Non-Uniform Requirements

When different pairs of nodes have different throughput, delay and bandwidth requirements, it is often possible to satisfy such requirements in a cost effective manner, by providing higher capacity connections between selected node pairs.

B. Large Capacity Gaps

Even if node pair requirements are uniform, a non uniform channel capacity assignment can be desired when there exist large gaps between the available capacity options, or when line tariffs have an irregular cost structure. In such cases, for a uniform capacity implementation one has to choose between two capacity alternatives, where the higher alternative is typically too expensive, and the lower is not adequate to satisfy all of the requirements, even if highly connected topologies are considered in the attempt of improving network performance. A blend of the two options generally provides the best solution.

C. Large Scale Economies

If the tariff structure offers a volume discount in the cost per unit bandwidth of leased wide band channels, then substantial line savings can be obtained by using a few channels with speed higher than average, to accommodate high volume requirements between different network regions, and lower speed channels to satisfy regional requirements. In the limit, the presence of a strong volume discount and a large network size might justify a hierarchical network implementation, with a different capacity selection for each hierarchical level.

D. Inadequate Reliability

In the design of communications networks it often happens that topologies which are very satisfactory from throughput and delay point of view, are nevertheless inadequate from the reliability point of view. One well known technique for the improvement of network reliability consists of increasing network connectivity with the introduction of new links. In order to optimize the cost-reliability trade off, links of lower capacity are often introduced.

E. Network Growth

During the life of a data network it is likely that at some point in time, tariff changes or new communications offerings will make it desirable to use channel types and speeds different from the ones selected for the original design. Since it is often impossible for practical reasons to reoptimize in one shot the topological configuration using only the new, more advantageous offerings, a partial reconfiguration is in general performed, introducing the new services where most beneficial. A topological design which accounts for multiple capacity options is therefore required.

Although multiple capacity options can introduce substantial cost savings, as mentioned in the above cases, they nevertheless require more sophisticated routing and flow control procedures than the single option case. For example, in the uniform capacity case the routing program needs to consider only channel utilizations (or, equivalently, channel queue lengths), while a non-uniform capacity routing program must account for utilization, transmission rate, propagation delay and error characteristics relative to the specific capacity option implemented on the channel.

Another important aspect to be considered is file transfer bandwidth, i.e., the maximum data rate at which a file can be transfered through the network. An approximate evaluation of file transfer bandwidth can be done by assuming that the data travels along only one source to destination path, and that packets are sent one after the other like in a "pipeline", without waiting for RFMM's or buffer allocation confirmations etc. Under such assumptions, the uniform capacity network bandwidth is clearly equal to the value of channel capacity, while in the non-uniform capacity network implementation the bandwidth along a source to destination path is equal to the <u>minimum capacity</u> in the path. Recalling that the average source to destination single packet delay is related to the average value of capacity along a path, while the maximum bandwidth is related to the minimum value of capacity, it will be observed that the file transfer performance is typically superior in a uniform capacity network than in a non-uniform one, assuming that average delay performance is the same.

All the above issues must be considered in order to decide whether to use multiple capacity options, or how many options to use, for each network design application. During the network planning phase it is important, therefore, to have available an efficient algorithim that generates low cost, multiple capacity network topologies. Thus, the practical importance of the algorithm described in the sequal.

II. BACKGROUND

The C S algorithm for multiple capacity options is an extension of the C S algorithm for fixed capacities. This section summarizes the general concepts and properties of the C S approach; a more detailed discussion is found in [1].

A cut is a set of lines whose removal will disconnect the network. A cut is saturated if the traffic load in every line of the cut equals line capacity. There are in general a large number of cuts in a network. When traffic load grows, one of the cuts approaches saturation. This cut is the bottleneck to any further throughput increase; therfore, if a higher throughput is desired, the capacity of the cut must be increased by upgrading some of the line capacities in the cut and/or introducing new lines across the cut.

Conversely, it is intuitively obvious, and has been experimentally confirmed, that the reduction or elimination of low utilized links within each of the partitions separated by the saturated cut will in general result in only a marginal reduction of network throughput if the network is at least 2-connected.

The C S algorithm is based on the above stated concepts. The algorithm attempts to keep network throughput within specified bounds while iteratively reducing overall line cost and maintaining capacity, delay and reliability constraints.

The effectiveness of the algorithm is closely related to our ability of solving the three following steps:

A. Determination of the saturated cut set.

B. Optimal increase of cut set capacity (obtained with appropriate link insertion of upgrading).

C. Reduction or removal of links not in the cut set.

The three above steps are extensively discussed in a later section.

III. STATE-OF-THE-ART

The topological design problem for packet switching networks has not been given in the past the same attention that was given to other similar problems (e.g. the Telpack problem, centralized network problems etc.) This is in part because the packet-switching technology is relatively new, and also because the design of a packet-switching network requires as essential ingredients the solution of complex queueing, routing and reliability problems. Therefore, the literature is rather scarce on this subject.

Among the few contributions available we mention here the Branch Exchange Method (BXC) [2] and the Concave Branch Elimination (CBE) [3].

The BXC method starts from an initial feasible topology and performs at each step a simple topological transformation (branch exchange). If cost-throughput trade off is improved, the transformation is accepted; otherwise it is rejected. The procedure stops after a large number of local transformations is systematically explored. The BXC method does not have the capability of identifying at each step the subset of transformations which are most likely to produce performance improvements; therefore all transformations must be systematically explored. BXC is a very time consuming algorithm and its application is limited to small or medium size networks.

The CBE method can be applied whenever the discrete costs corresponding to the multiple capacity options can be reasonably approximated by concave curves. The method consists of starting from a fully connected topology, using concave costs and repeatedly applying a minimum cost routing algorithm [3] until a local minimum is reached. Typically, the algorithm eliminates uneconomical links, and strongly reduces the topology. Once a locally minimal topology is reached, the discrete capacity solution can be obtained from the continuous solution with an appropriate selection of capacity options. Since 2-connectivity is required, the algorithm is terminated whenever the next link removal violates this constraint; the last 2-connected solution is then assumed to be the local minimum. In order to obtain several local minima, and therefore several different topological solutions, the algorithm is applied to several randomly chosen initial flows.

The CBE algorithm has a better knowledge of the structure of the problem (network topology, cost-capacity volume discount etc.) than the BXC method and improves network performance at each interation. It is also less time consuming than BXC, since it explores only a selected subset of solutions (namely the local minima). However, it suffers the following limitations: it can only remove links (i.e., no links are added during the optimization procedure), and it requires smooth cost-capacity curves, without large gaps and irregularities.

In summary, the presently available methods are either computationally inefficient, or they are able to handle only very special line tariff structures. There is therefore the need for algorithms which are computationally efficient, and that can handle very general line cost structures. The C S method described in the sequal has been developed with the intent of meeting the two above objectives.

IV. THE CS METHOD FOR MULTIPLE CAPACITY OPTIONS

A. General

The CS method for multiple capacity options is based on the same concepts that inspired the development of the original CS method for fixed capacities. Some changes and some new features were required in the practical implementation of the algorithm, due to the presence of multiple capacity levels. The major changes are relative to:

1. The determination of the saturated cut;
2. The insertion or upgrading of links;
3. The deletion or reduction of links; and
4. The determination of starting topologies.

These issues are discussed in the following sections.

B. Determination of the Saturated Cut

The saturated cut is found using the following procedure. Network throughput is first increased, using a very efficient routing algorithm, until the delay constraint is satisfied with equality. The output of the program consists of the optimal values of cost, throughput and delay, and the value of the optimal flows in all links. Next, links are ordered according to their utilization, which is defined as the ratio of flow to capacity, and are removed starting from the most utilized one, while the network becomes disconnected. The minimum disconnecting set of links is defined as the saturated cut.

It was already observed in [1] that, for an appropriate determination of the saturated cut, the chains of nodes that carry prevalently transit traffic must be "collapsed" into a single equivalent link, before the above described removal operation is performed.

The criterium presented in [1] to determine whether a chain is collapsable or not, is rather heuristic and not sufficiently precise for the multiple capacity case, where the links in the same chain can have different capacity and different utilization values. Therefore, the following more accurate criterium was developed.

Let us assume that the traffic requirement matrix $R = [r_{ij}]$ is symmetric and therefore in each link the flow is the same in both directions; let f_1 and f_n be the terminal flows in the chain shown in Figure 1.

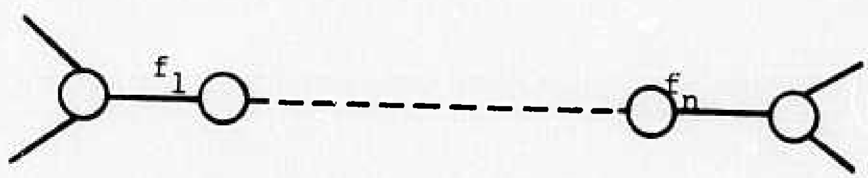


FIGURE 1: COLLAPSABLE CHAIN

Let $Q \triangleq \sum_{i \varepsilon I_c} \sum_{j \not\varepsilon I_c} r_{ij}$ (where I_c is the set of nodes in the chain) be the traffic originating from the nodes internal in the chain and transmitted to the external nodes (internal traffic). Letting S be the one way transit traffic in the chain, it is easily seen that:

$$S \simeq \frac{f_1 + f_n - Q}{2}$$

The chain is collapsed when the transit traffic S is predominant with respect to the internal traffic. More specifically, the chain is collapsed if

$$\max_{i=1,\ldots m} \left(\frac{f_i - S}{f_i} \right) \leq \alpha$$

, where α is an input parameter experimentally adjusted.

Although the above criterium is not completely fail-safe (e.g. is possible to construct pathological examples in which chains with predominantly internal traffic are declared collapsable by the test), the criterium has been found adequate for most network design applications.

C. Link Insertion or Upgrading

In order to select the link to be inserted or upgraded, the following ratio is computed for each existing or potential link connecting any two nodes separated by the cut set:

$$\rho_i = \frac{D_i' - D_i}{C_i' - C_i}$$

Where C_i and D_i are capacity and cost of the current option and C_i' and D_i' are the values corresponding to the next available option. If the link is new, $D_i = C_i = 0$; if the link cannot be expanded any further, $D_i' = \infty$. Special attention, and slightly more elaborate expressions are used in the case of chain upgrading.

After having computed all ratios f_i's, the link which minimizes the ratio (i.e. provides the lowest incremental cost per bandwidth) is introduced or upgraded to option C_i'.

The minimization of the above ratio represents the basic criterium for link insertion or upgrading. Such criterium can be improved and extended to account for the effect of a new link introduction on reliability (e.g. a link which eliminates pendant nodes or long chains is a favorite candidate, and to weigh the capacity increase $C_i' - C_i$ across the cut set with the amount of capacity actually required to achieve the target throughput REMAX. These and several other versions of the upgrading criterium are presently being experimented.

D. Link Reduction and Removal

The criterium for link reduction (or removal is based on the maximization of the following ratio over the existing links:

$$\theta_i = \Delta D_i \left[\beta \Delta C_i + \Delta f_i \left(\alpha + (1-\alpha) \frac{\Delta f_i}{f_i} \right) \right]^{-1}$$

where:

C_i' and D_i' capacity and cost of lower option for link i.

$\Delta C_i = C_i - C_i'$: capacity reduction on link i.

$\Delta D_i = D_i - D_i'$: cost saving corresponding to capacity reduction for link i.

$f_i =$ current flow in link i

$\Delta f_i = \max\ (f_i - C_i', 0)$

α and β input controlled parameters varying between 0 and 1.

Ratio θ is calculated for all existing links except for the links in the cut set and the link which maximizes the ratio is reduced or removed. Parameters α and β are experimentally adjusted and vary between 0 and 1. If $\alpha=1$ and $\beta=0$ then $\theta_i = \Delta D_i/\Delta f_i$, and the ratio represents the cost saving per unit of flow that must be rerouted on other links, after the reduction of link i. By letting $\alpha=\beta=1$, one equally penalizes the loss of capacity and the need to reroute flow on link i. The proper selection of α and β depends on problem characteristics (cost structure, delay requirements etc.) and can be done only experimentally.

In the case of link removal, the additional criterium that the network must remain 2-connected after the removal is applied.

E. Starting Topologies

The determination of good starting topologies to initialize the C S method is more critical in the case of multiple options than in the single option case. In fact, the multiple option algorithm is "more local" than the single one, in the sense that it tends to upgrade or reduce (rather than introduce or remove) links and therefore it does not produce such drastic topological changes as the fixed capacity option algorithm. Clearly, the locality can be corrected by properly adjusting the criteria for link insertion and removal; however, some degree of locality will always remain. Therefore, it is important to start from potentially good topologies.

An effective selection of starting topologies is offered by single option C S method solutions providing throughput and delay performance close to the designed requirements. The multiple option CS method applied to such topologies can thus be interpreted as a refinement procedure on an initial fixed capacity solution.

F. The Algorithm

The CS algorithm generates low cost, 2-connected topologies with throughput performance ranging between two specified throughput levels REMIN and REMAX. An iteration of the algorithm consists of the following fundamental steps:

1. Solution of an optimal routing problem for the current topology and capacity assignment, and determination of the saturated cut.

2. Topology and/or capacity modification, in order to improve network cost-effectiveness and to drive the solutions to the desired throughput range. In particular the algorithm performs one of the following operations:

   a. Increase only, (i.e. links are upgraded or added) if the current throughput level RE<REMIN;

   b. Reduce only, (i.e. links are reduced or deleted) if RE>REMAX;

   c. Perturbation (i.e. one link reduction (or deletion) and one link upgrading (or insertion) are performed simultaneously) if REMIN<RE<REMAX.

3. Acceptance test. If the new solution is dominated by previous solutions, or it is not cost-effective, or it is a repeat solution, the algorithm goes back to step B and selects an alternative modification. Otherwise it returns to step A.

The algorithm is initialized by providing an initial starting topology, and produces a new network configuration at each iteration.

V. AN APPLICATION

As an application of the multiple capacity CS algorithm we consider the design of a 26 node ARPANET like topology using 3 capacity options (9.6, 19.2 and 50 kb/s), with throughput requirement ranging between REMIN = 100 kb/s and REMAX = 200 kb/s, and with delay requirement $T_{max} \leq .250$ sec. Traffic demands are symmetric and uniform over all node pairs.

The following line and modem costs are assumed for the above mentioned capacity options:

Cap kb/s	Fixed cost $/mo	Line Cost $/mile x mo
9.6	493	.42
19.2	850	2.50
19.2 (using biplexers)	1500	.84
50	850	5.00

Biplexers are used to implement our 19.2 kb/s circuits from two 9.6 kb/s circuits in parallel, whenever this alternative results more economical than the standard 19.2 AT&T offering.

Several starting solutions were considered. Among them we mention an all 50 kb/s configuration shown in Figure 2; and the same topology as in Figure 2 but with all 9.6 kb/s links. The 50 kb/s solution is not very cost-effective considering our throughput requirements. The 9.6 kb/s solution is, on the other hand, inadequate for the .250 sec delay requirement (See Figure 3).

Starting from the above configurations several solutions were generated. The most representative ones are shown in the throughput vs. cost curve plotted in Figure 3. A typical solution is shown in Figure 4: notice that most of the short connections are implemented with 50 kb/s circuits, while the medium and long links use 9.6 and 19.2 kb/s circuits. This behavior is due to the particular cost structure considered. Figure 5 and Figure 6 show two CS solutions which have approximately the same cost and throughput performance but have different connectivity. The first solution is more attractive from the high bandwidth point of view since it provides a 50 kb/s route across country; the second is more attractive for reliability.

Notice that most CS solutions are regularly aligned along a curve which represents the cost-throughput trend for the 26 node network using the given capacity options. The same trend is shown by the lower bound cost solutions obtained by solving the topological problem using concave instead of discrete costs (See Figure 7) and relaxing the 2-connectivity constraint. The solid curve in Figure 3 connects such lower bound solutions.

Finally, the Concave Branch Elimination (CBE) method described in Section 2 was applied to the problem and two solutions are shown in Figure 3. Notice that the CBE solutions are well in line with the CS solutions, although their topology typically shows a higher degree of connectivity. This fact allows us to conjecture that both CS and CBE solutions are near optimal.

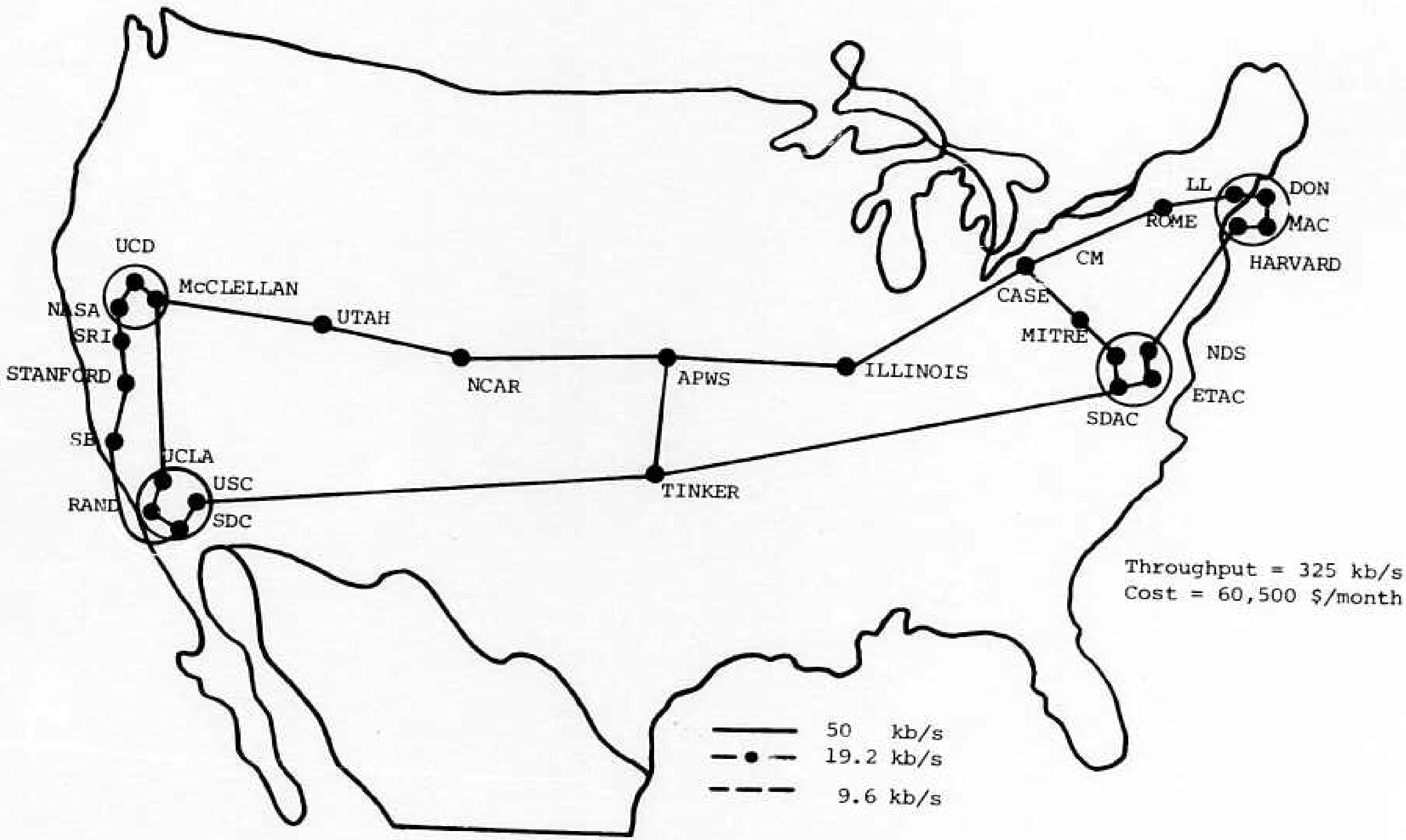


FIGURE 2 : INITIAL 50 KB/S CONFIGURATION

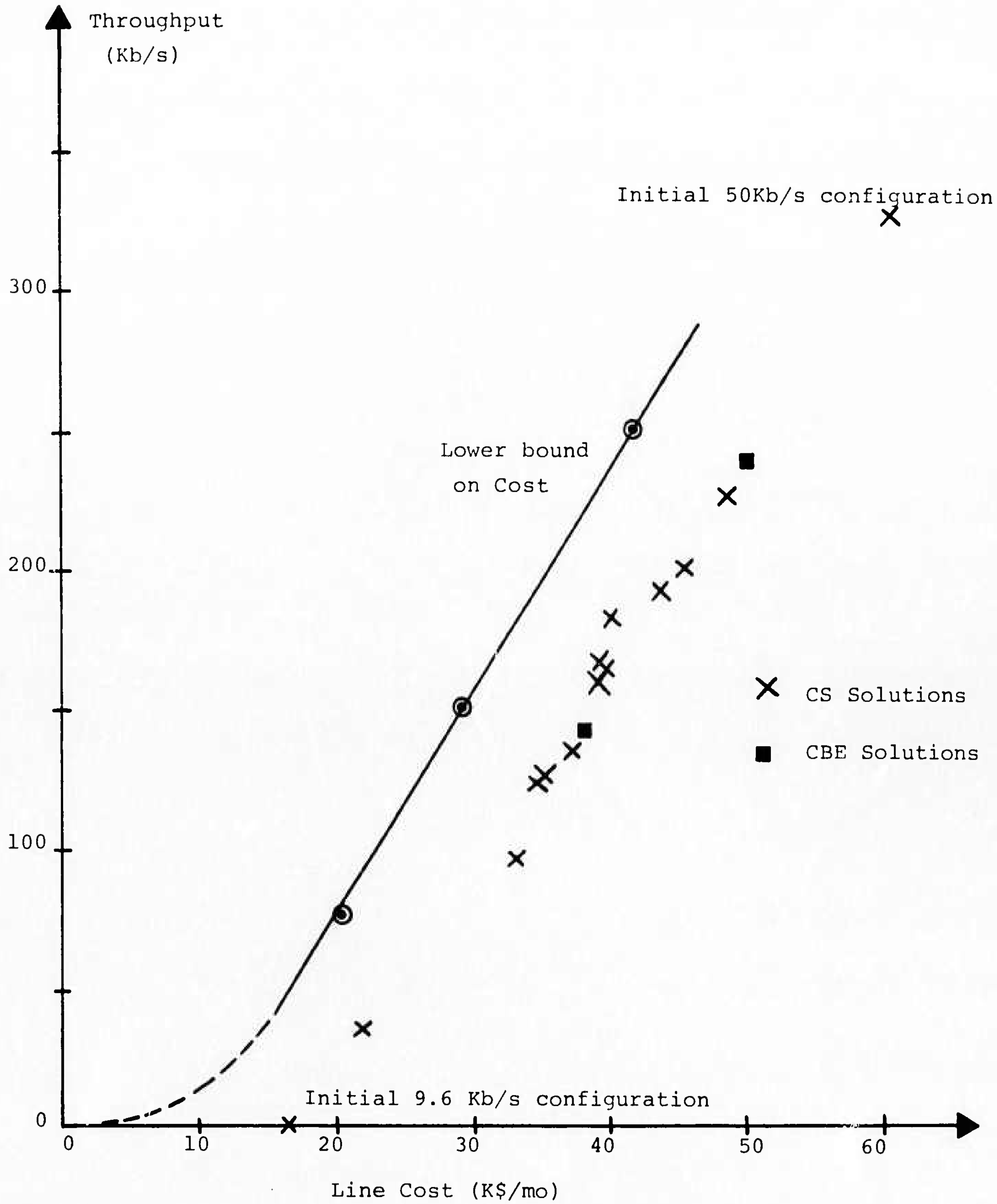


FIGURE 3 : THROUGHPUT VS. COST PERFORMANCE

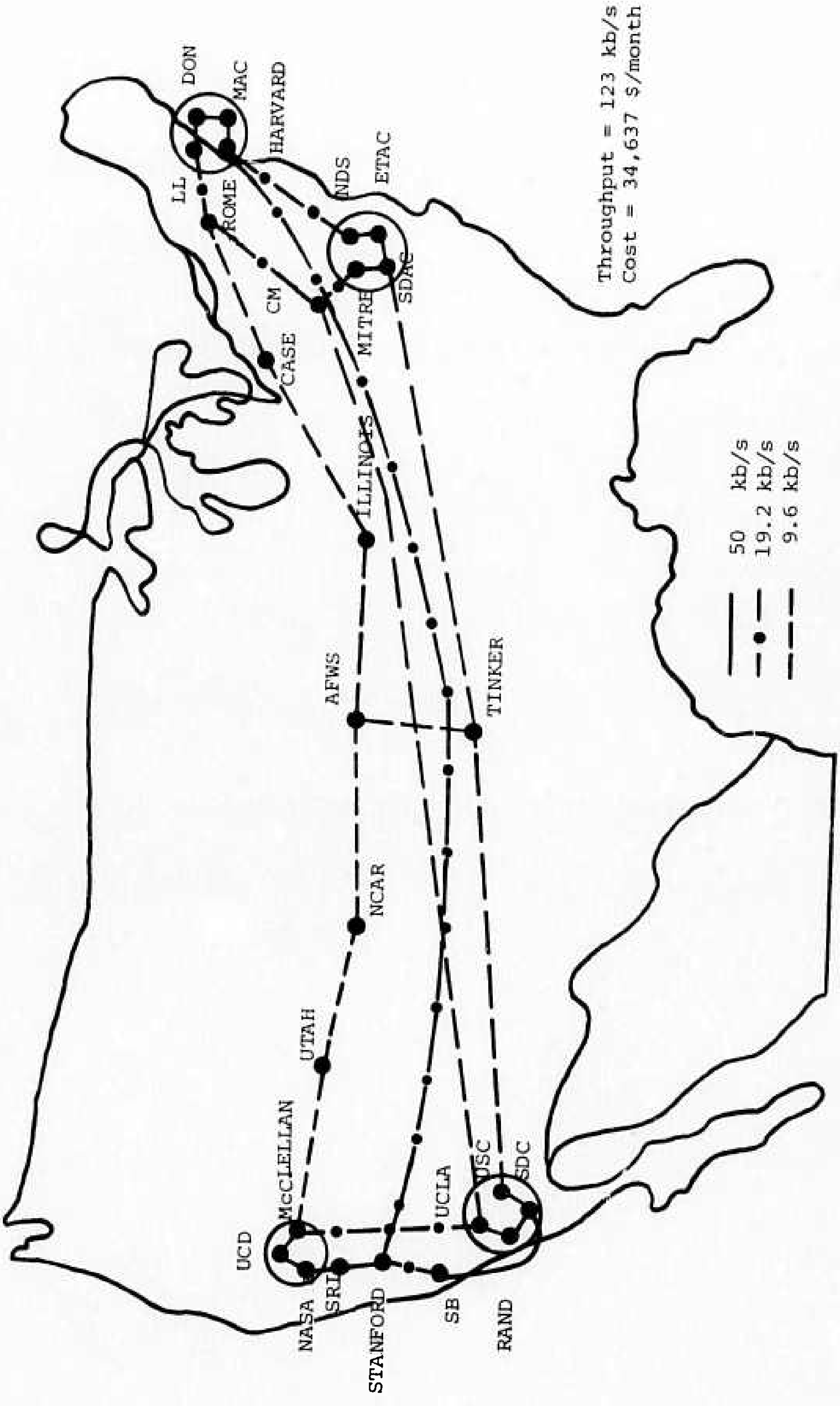


FIGURE 4 : TOPOLOGY OBTAINED WITH THE C-S METHOD

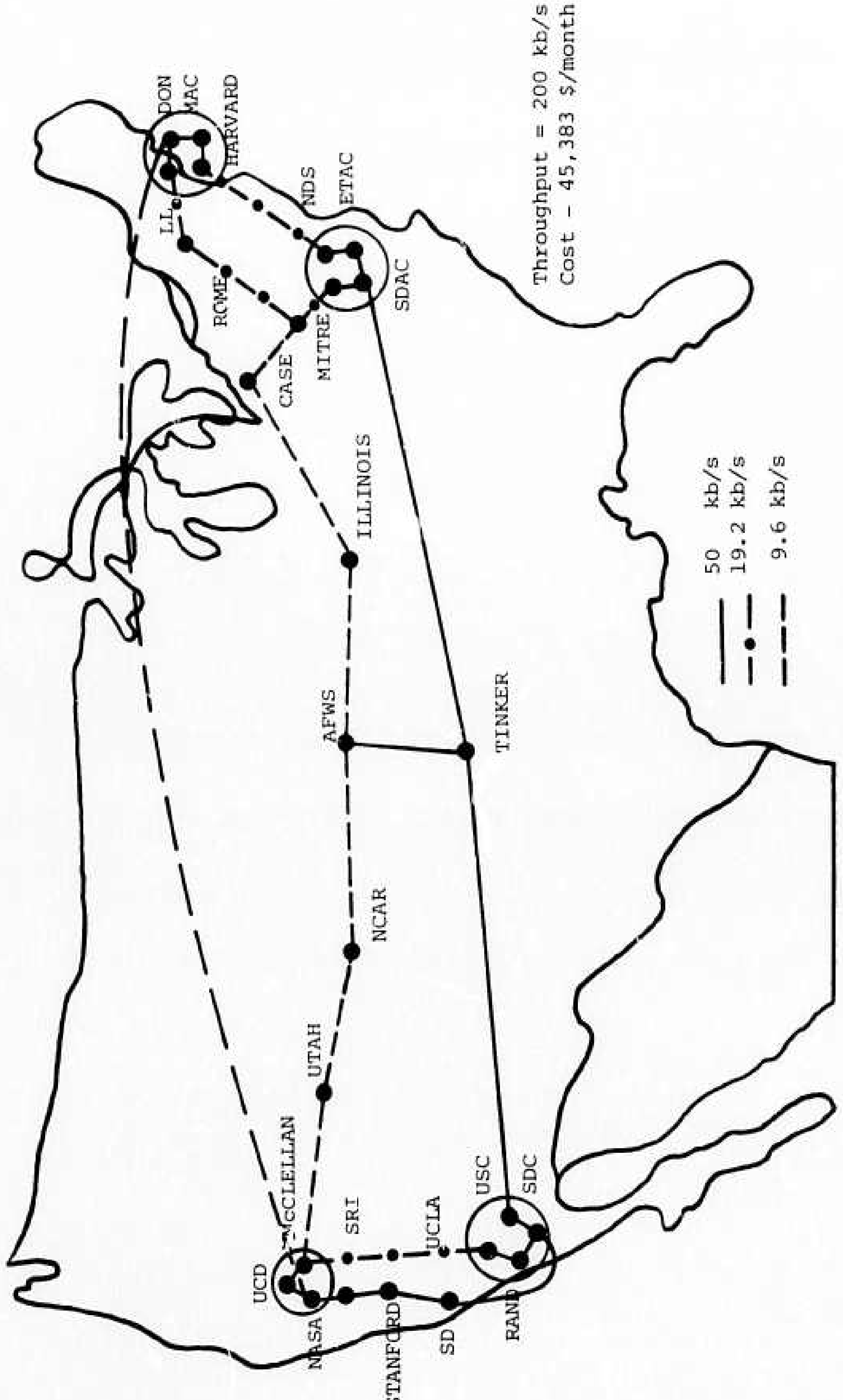


FIGURE 5 : TOPOLOGY OBTAINED WITH THE C-S METHOD

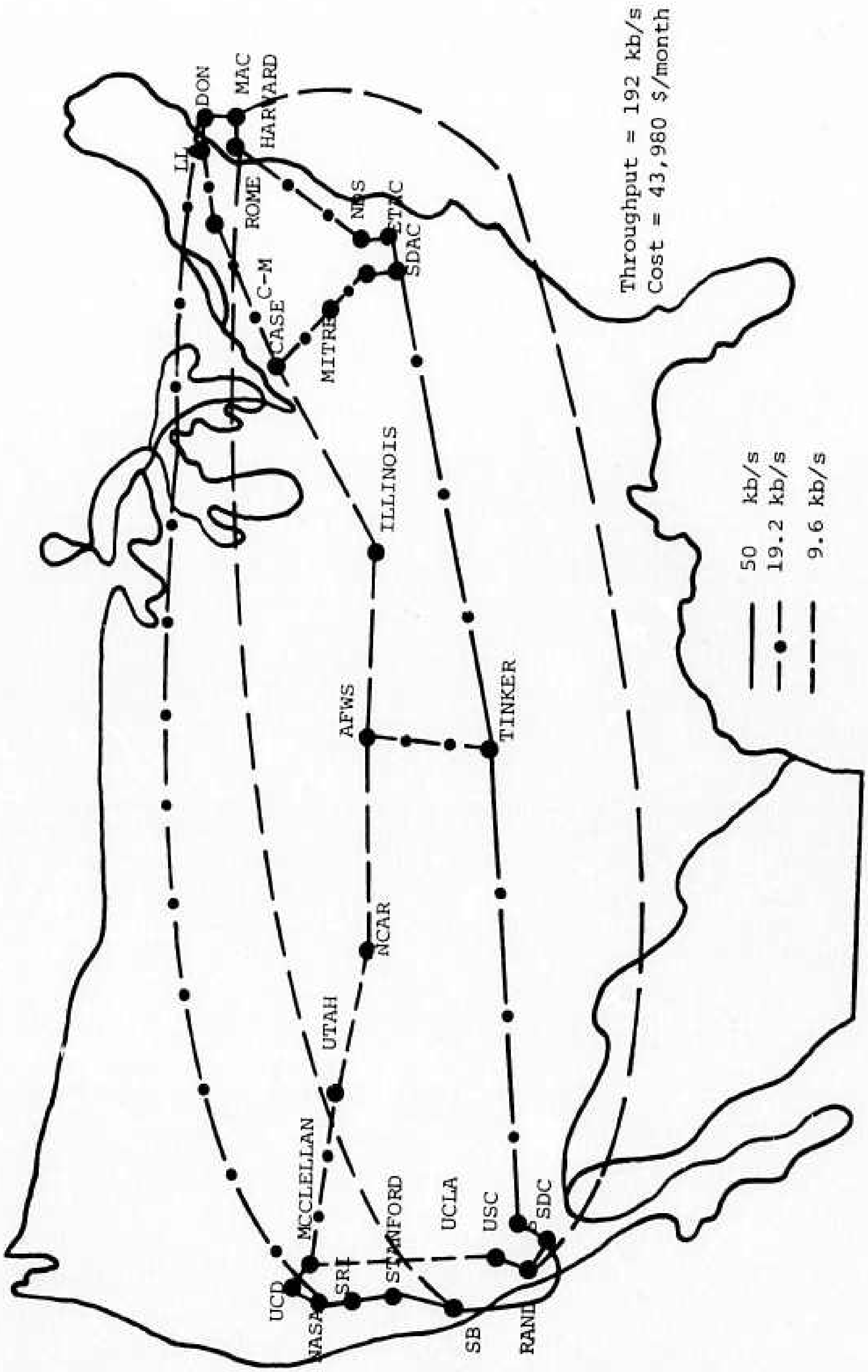


FIGURE 6 : TOPOLOGY OBTAINED WITH THE C-S METHOD

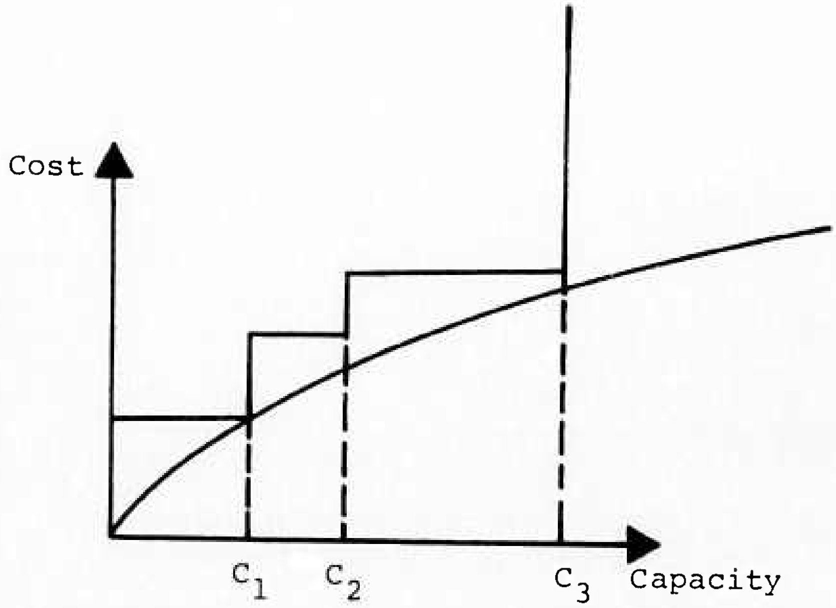


FIGURE 7 : LOWER BOUND TO ACTUAL LINK COST

VI. DIRECTIONS FOR FUTURE RESEARCH

The CS algorithm is an important contribution to the topological design of packet-switched networks, in that it is computationally very efficient and applicable to very general cost structures. However, more research is required in this and other areas of network design. In the sequel we report some directions for future investigation.

Present topological design techniques are based on static performance criteria (e.g. average flow and delay) and do not take into adequate consideration the impact of topology and channel selections on some of the important aspects of network operation (e.g. adaptive routing, flow control, buffer overflow, traffic congestion etc.) It is important therefore to further investigate the relationship between topological characteristics (such as connectivity, use of different line capacities, etc.) and dynamic performance criteria (e.g. network controllability, file transfer bandwidth, etc.) and to include dynamic performance considerations in the design phase by using appropriate constraints and design criteria.

Since the CS algorithm is based on a variety of heuristic steps, it is clearly possible to improve the algorithm by refining some of the steps. Therefore, research should be pursued on issues such as: development of efficient procedures for multiple (rather than single) insertion, deletion, upgrading and reduction of links at each CS iteration; development of more efficient techniques to predict the cost-throughput effectiveness of each topological transformation, and thus select the most effective transformation; inclusion of more precise reliability concepts in the design criteria, etc.

While experimenting the CS algorithm with multiple capacity options we noticed that, although the algorithm is very general and can deal with any type of line tariffs, nevertheless it was giving better results with some tariff structures than others. This fact suggests that it is possible to improve the performance of the algorithm by adjusting the heuristics to the particular cost structure of the problem. Clearly, it is not desirable to have around several design programs, one for each cost structure. The designer should identify the steps that are most sensitive to tariff changes, and should implement the computer program in a modularized fashion, so that only a few modules will be modified when tariffs change [4].

Another important direction for future research is the design of large networks, with hundreds or thousands of nodes. Such networks cannot be directly designed using traditional algorithms, because of the prohibitive computer time and memory requirement. The typical approach consists of partitioning the network into hierarchical levels and applying traditional design techniques within each partition. Since a large network will contain several partitions, each requiring a separate topological design, there is the need for very efficient design algorithms. Therefore, the CS algorithm can be considered an important contribution to large net design. The next important steps are: the development of good criteria for node partitioning; the implementation of an efficient data base to store the input parameters and intermediate design results; and the evaluation of global network performance. Preliminary results in some of the above areas are already available, and are reported in the First and Second semi-annual NAC reports [1] and [5]), but more research needs to be done.

Since most design algorithms are based on heuristics, it is conceivable that such algorithms can be greatly enhanced with the aid of interactive graphics. The implementation of analysis and design programs supported by interactive graphics can be useful in two ways:

1. It can assist in the <u>development</u> of better heuristics, since it allows immediate appraisal of the

affect of changes in parameter values and design criteria on algorithm performance;

2. It offers to an experienced designer the possibility of monitoring the network design process iteration by iteration, and of correcting eventual inefficiencies of the algorithm.

Considerable research efforts have been dedicated at NAC to the development of interactive graphics software for network analysis and design applications. The results are described in other sections of this report and clearly indicate that the development of interactive design programs supported by a graphic terminal is another very important direction for future research in the area of network design.

REFERENCES

1. Network Analysis Corporation, "The Practical Impact of Computer Advances on the Analysis and Design of Large Scale Networks," Second Semiannual Technical Report, December 1973.

2. Frank, H., I.T. Frisch, and W. Chou, "Topoloogical Considerations in the Design of the ARPA Computer Network," AFIPS Conference Proceedings, Vol. 36, SJCC, Atlantic City, N.J., 1970, pp. 581-587.

3. Gerla, M., "The Design of Store-and-Forward Networks for Computer Communications," Engineering Report #7319, School of Engineering and Applied Sciences, UCLA, Los Angeles, California, January 1973.

4. Gerla, M., "New Line Tariffs and Their Impact on Network Design," NCC Conference Proceedings, Chicago, May 1974, pp. 577-582.

5. Network Analysis Corporation, "The Practical Impact of Recent Computer Advances on the Analysis and Design of Large Scale Networks," First Semiannual Technical Report, May 1973.

TERMINAL ORIENTED NETWORK COST AND PERFORMANCE - PART 3
THE ACCESS FACILITY LOCATION PROBLEM

I. INTRODUCTION

In any network with a large number of widely dispersed "users" accessing a limited number of "resources", the strategy for access will play a large part in determining the cost and performance of the network. The "users" may include not only time-sharing terminals, but also terminals used for message transfers, remote automatic sensing devices (such as might be found in an environment monitoring situation), manned sensing stations, and several others. The "resources" may be as sophisticated as many heterogeneous computers tied together in a packet switching high level subnet, such as in the ARPANET, or as simple as a single computer processing data received from automatic remote sensing devices. An almost endless number of "user" and "resource" combinations, both covering and extending the range described above, appear possible. Effective, economical "user" access will depend on both the development of hardware to facilitate access, and the development of network and topology design techniques to effectively utilize such hardware. This chapter continues the study of this problem.

There are several ways to provide access, including land lines (i.e. ordinary telecommunication channels as derived from cable or LOS microwave transmission systems) and packet radio techniques. In this chapter, advances in the design techniques for land line approaches are reported; advances in packet ratio techniques are reported in other chapters. Cost effective land line access will depend on an effective line layout algorithm to connect "users" with access facilities, development of hardware to serve as access facilities, and an effective facility location algorithm to determine both the number and location of access facilities. The line layout problem for a given set of access facility locations has been effectively dealt with in Part 1 in Semiannual Report #1. Advances in hardware serving as access facilities have been discussed in Part 2 in Semiannual Report #2. In this part the development of an effective facility location algorithm is reported.

The facility location problem can be formulated in many ways, and can appear at several different levels within the same network design problem. For a network such as the ARPANET, the resources (hosts) are distributed over a large geographical region, and the users (terminals) may connect to them directly, or access them through the packet switching subnet by being connected to an access facility (such as a TIP or ANTS), as shown in Figure 1. If the resource locations and user locations are fixed, as shown in Figure 2, the problem becomes one of locating the access facilities (TIP or ANTS) to minimize the cost, subject to capacity, performance, and reliability constraints. When the locations of these facilities (TIP or ANTS) are fixed, the problem may appear at the level of using concentrators or multiplexers as "access facilities" for connecting the users to these new "resources," (i.e. the fixed TIP or ANTS locations). Because the facility location problem can be posed for various levels of the same design problem, with devices performing different "functions" depending on context, we will pose the problem in terms of a generic access facility, called a TACOM (TIP, ANTS, concentrator or multiplexer), and a generic resource to which users and/or TACOMs are to be connected, called a RESCOP (resource connection point). We will also use the term "node" as a generic replacement for "user". Thus, the general problem is one of locating TACOMs to most economically allow connection of a given set of nodes to a given set of RESCOPs.

The most basic formulation of this problem is when there is only one RESCOP location, and the nodes may be connected directly to the RESCOP, or through TACOMs, which are in turn connected directly to the RESCOP. The recent advances reported here are, for simplicity, presented in the context of this basic formulation. After presentation of the basic algorithm and discussion of its performance characteristics, we consider its application to the more general problem with multiple RESCOP locations.

In Section II a simple description of the problem is developed. In Section III, related problems and solution-techniques are described, and in Section IV previous approaches to this problem are considered. In Section V, a simplified description of our approach is given, and in Section VI a detailed description of the algorithm is presented. Results of experiments to determine the performance of the algorithm are given in Section VII, and extensions to more general problems are given in Section VIII.

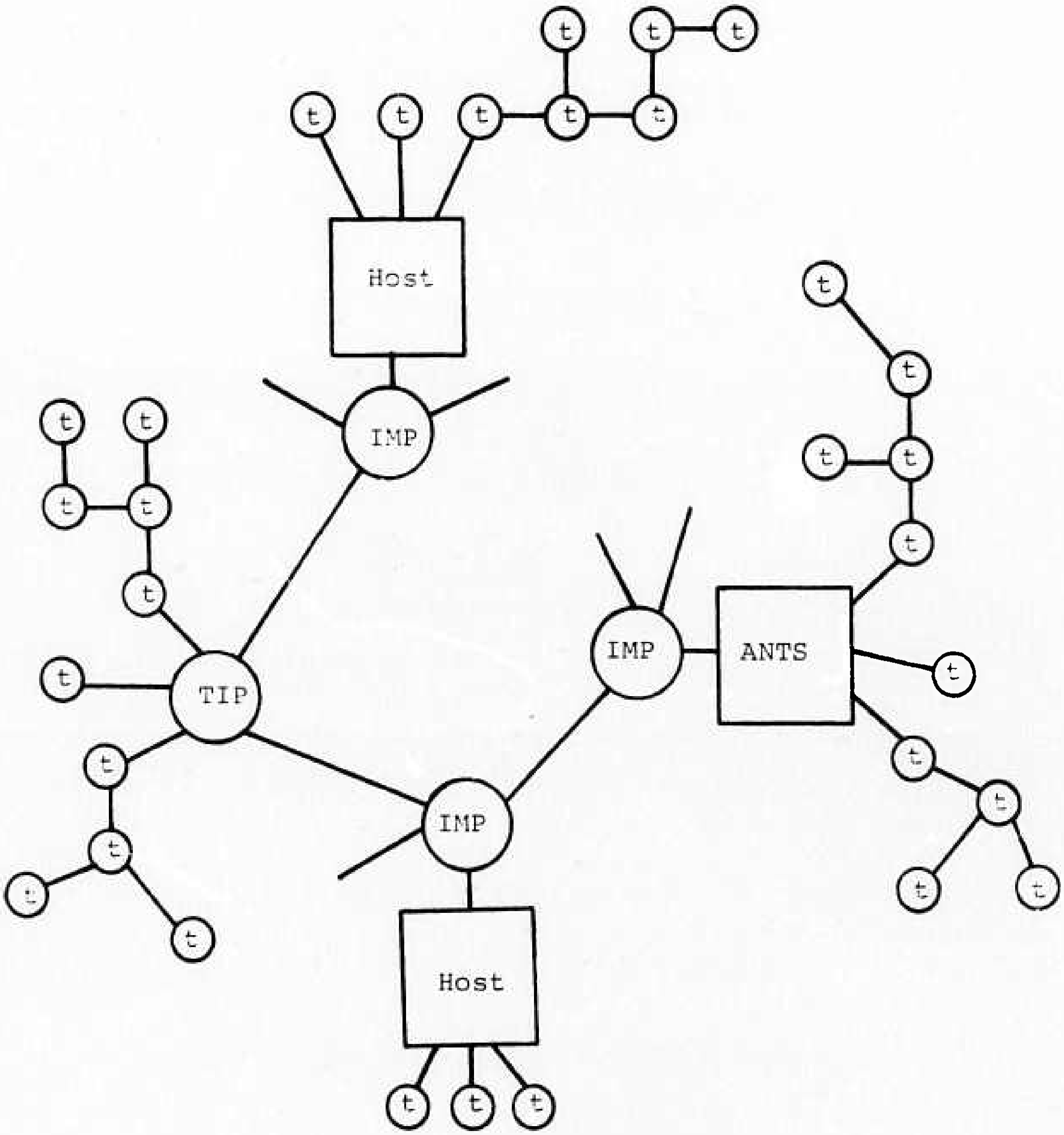


FIGURE 1: ACCESS ALTERNATIVES FOR ARPANET

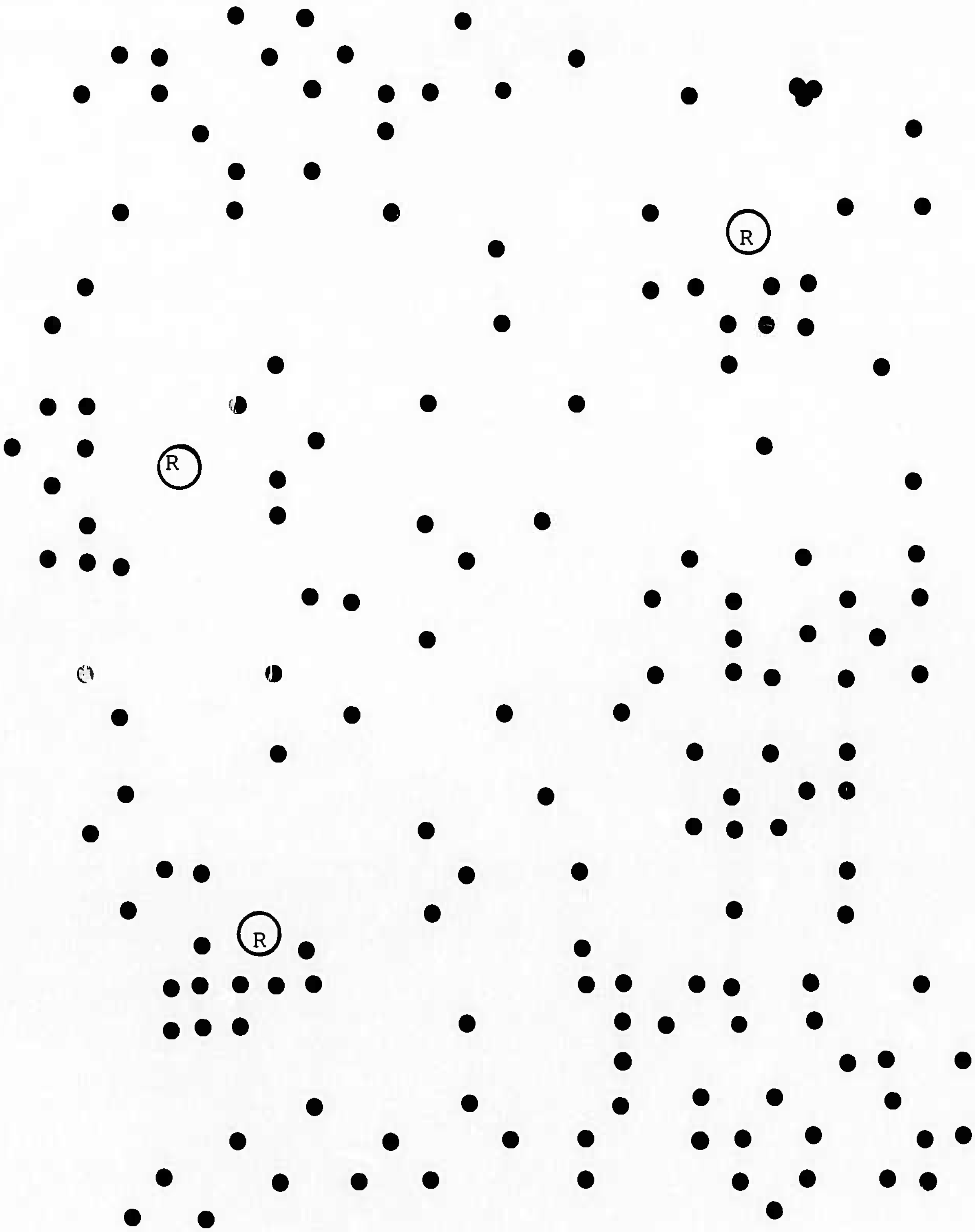


FIGURE 2: GENERAL PROBLEM FORMULATION

II. BASIC PROBLEM STATEMENT

The basic problem may be posed as that of a geographically distributed set of nodes that must be connected to a RESCOP (Resource connection point). The most primitive topology for such a network is shown in Figure 3 (a). Each node is connected directly to the RESCOP. There is no topological design effort associated with this network. It's cost is easily expressed as

$$COST_{STAR} = \sum_{i=2}^{N} d_1(i)$$

where d_i (j) is the cost of joining node j to node i, and the RESCOP has been made node 1.

A more sophisticated design alternate is shown in Figure 3 (b). In this case, TACOMS (TIP, ANTS, concentrators, or multiplexers) have been used to save cost. Here the topological design effort is directed at selection of number, and location of TACOMs, and assignment of nodes to the TACOMs, so as to minimize cost. The TACOMS have a cost, D, and a cost of being connected to the RESCOP when located at node i, $d_1^c(i)$, which may be different than the node connection cost (due to the possibile bandwidth requirements difference). Consider TACOMs at the two nodes k and ℓ, and let B_k and B_ℓ be the set of nodes which are assigned to k and ℓ, respectively, with A the set of all nodes. The cost of the star-TACOM network may then be expressed as

$$\begin{aligned} COST_{STAR\text{-}TACOM} = & \left[\sum_{i \in B_k} d_k(i) + D + d_1^c(k)\right] \\ & + \left[\sum_{i \in B_\ell} d_\ell(i) + D + d_1^c(\ell)\right] \\ & + \sum_{i \in (A-B_k-B_\ell)} d_1(i) \end{aligned}$$

Let C be the set of TACOM sites, with $C \subset A$. The design problem may then be expressed as:

Select C and B_j, $j \varepsilon C$ so as to minimize

$$COST_{STAR-TACOM} = \sum_{j \varepsilon C} [\sum_{i \varepsilon B_j} d_j (i) + D + d_1{}^c (j)] + \sum_{i \varepsilon (A - \bigcup_{j \varepsilon C} B_j)} d_1 (i)$$

When nodes do not have to be connected directly to the host or a TACOM, but rather can share a line (such as in a polling situation), considerable cost savings can result from proper line layout, as shown in Figure 3(c). If there are no constraints on the number of nodes which can share a line, then the optimal topology is simply a minimum spanning tree. When constraints are present, the design problem becomes more interesting. This problem was effectively formulated and dealt with in Part 1 in Semiannual Report #1.

A third, more general, and often most effective, design alternative is the combined use of multidrop lines and TACOMs, as shown in Figure 3 (d). In this case, the problem is one of total network design, including both selection of TACOM locations and line layout. The other design alternatives may be considered as subsets of this alternative.

It is this general problem that is considered here. There are many possible formulations of this problem, depending on the selected constraints and cost functions. We present our basic results in terms of the simplest of these formulations, and then in a later section consider the many possible extensions. This formulation is given below:

Given: A - set of nodes $i = 2,\ldots N$

node 1 - RESCOP site

H - set of possible TACOM sites

w_{max} - line capacity

c_{max} - TACOM capacity

$d_i(j)$ - cost of connecting node j to node i

charge - cost of TACOM

FIND: Low cost network design subject to the constraints:

A) No line may have more than w_{max} nodes.

B) No TACOM may serve more than c_{max} nodes.

In this formulation the set of possible TACOM sites, H, is defined separately from the set of nodes, A. In most practical problems the possible TACOM sites are limited to a subset of the nodes selected on considerations of maintenance, rental space, access by trained company personnel, security, etc. However, it is quite feasible for situations to occur where the possible TACOM sites are in fact disjoint from the nodes (such as in a commercial time sharing operation where TACOMs must be at company provided locations but all terminals are at customer locations), partially overlap with the set of nodes, are a proper subset of the nodes, are the same as the nodes, or have the nodes as a proper subset. For simplicity, we have chosen to present the algorithm for the case where the nodes are the possible TACOM sites, thus dealing with only one set, A. In a later section the algorithm will be also shown to easily handle the other cases noted above.

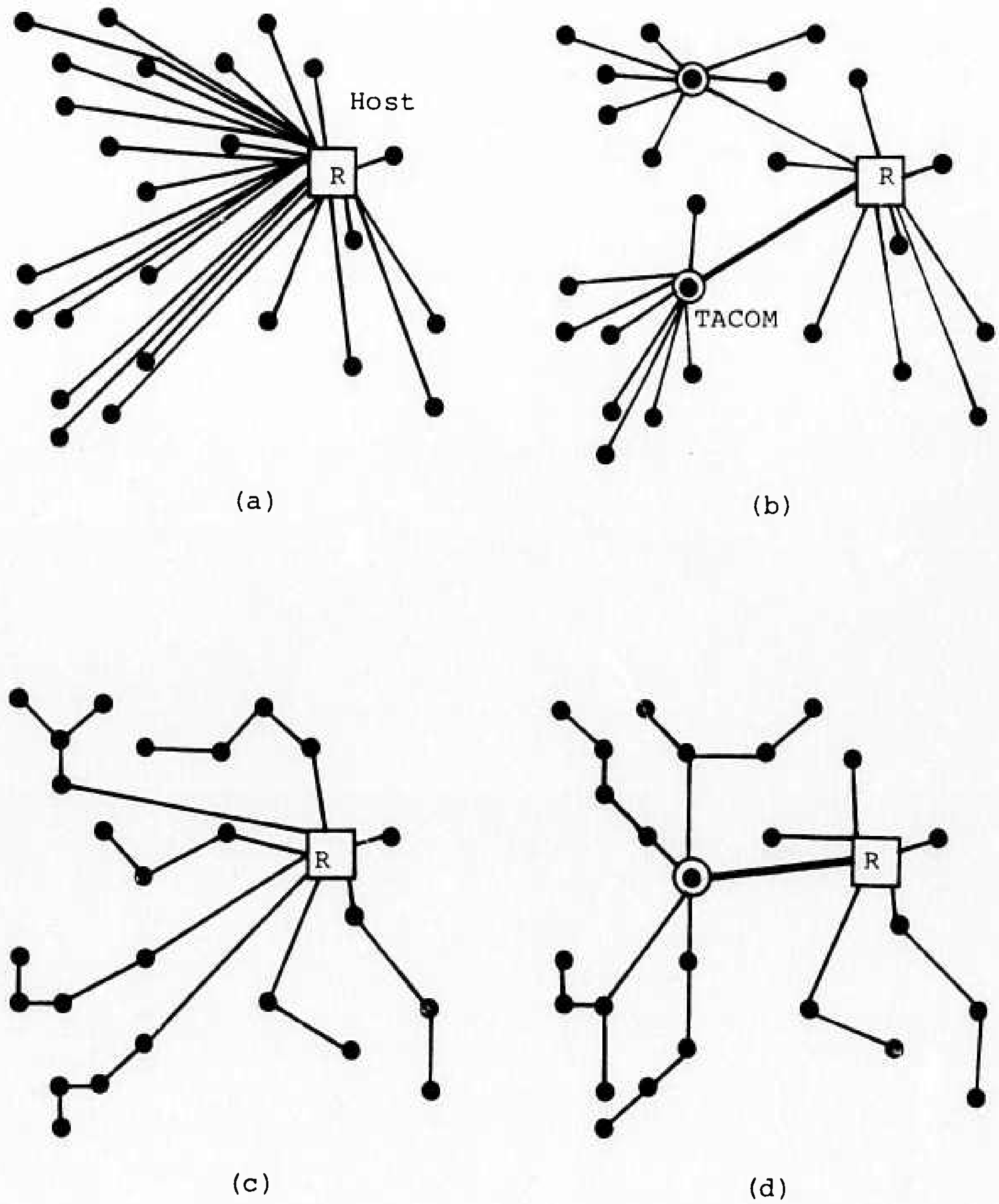


FIGURE 3: BASIC NETWORK TOPOLOGY PROBLEM

III. RELATED PROBLEMS AND SOLUTION TECHNIQUES

There are many problems that are related to the TACOM location problem, including warehouse location problems, clustering problems, and partitioning problems. In this section we discuss these problems, their solution techniques, and the applicability of these techniques to the TACOM location problem.

A. Warehouse Location Problems

The warehouse location problem may be briefly defined as the determination of the number, location, and capacity of source sites in order to minimize the cost of satisfying a set of shipping requirements under a given cost matrix [6]. Efroymson and Ray present a BranchBound algorithm solution technique for this problem when it is formulated in a manner analogous to the TACOM location problem with a pointtopoint connection constraint [8]. Relative to the general TACOM location problem, this approach has two drawbacks: it does not easily handle the multidrop line case, and, for large problems, the computational requirements are prohibitive.

The computational requirements for obtaining exact solutions to these problems have given rise to many heuristic approaches. Among the more successful is the "Add" algorithm [21]. In this approach a star network with all "customers" directly connected to a central warehouse is assumed to start with. Each possible warehouse location is then evaluated by determining the cost reduction which would be achieved by placing a warehouse at the location. The location giving the greatest reduction is then selected for the first placement of a warehouse. With the new warehouse in place, the process is repeated for the next location. When no further cost reduction is achieved, the process halts. Relative to the TACOM location problem, this approach also has the drawback of not handling the multidrop line case efficiently; evaluating each site with a complete line layout would be too computationally costly. In addition, each time a site is selected, it becomes permanent, but there is little reason to think that the location of the best single TACOM is also one of the best locations for two TACOMS, etc.

The "Drop" algorithm [10] is basically the reverse of the "Add" algorithm. Warehouses are initially assumed to be at all possible locations, and "customers" are assigned to multiple warehouses. The procedure is to then eliminate the warehouse whose elimination most reduces the cost. The process is repeated until no further reduction is possible. The difficulties here are directly analogous to those of the "Add" algorithm.

Other, less promising, formulations of the warehouse location problem and related solution techniques are contained in [26], [13], [9], [14].

B. Clustering Problems

Basic aspects of the TACOM location problem can be posed in a clustering context. The clustering problem may be thought of as detecting inherent separations between subsets of a given point set, where the separations may be in terms of several different measures. In the TACOM location problem, one might expect some measure to be available such that nodes clustered under this measure would most appropriately share a common line, or be connected to the same TACOM. Many clustering techniques exist [3], [23], [15], [1], [22]; however, few appear to suggest measures which may be applicable to the TACOM location problem.

Among the more promising are those which attempt to cluster points on a plane [28], [17]. Zahn, [28] attempts to identify gestalt clusters (twodimensional point sets naturally perceived as separate groupings) by connecting the points with a minimum spanning tree, and deleting relative long branches to form components, or clusters. Jarvis and Patnik [17] offer a similarity measure based on shared near neighbors to generate "nonglobular" clusters. This technique and that of Zahn have much in common. However, neither of the approaches appear to offer more than insight to the complexity of the TACOM location problem.

C. Partitioning Problems

The TACOM location problem may be viewed as that of partitioning a set of elements in a way that optimizes some objective function defined on the set of all partitions. The partitioning may be in terms of nodes sharing a common line, or in terms of nodes connected to the same TACOM, and the objective function is simply network cost. There are several formulations to the partitioning problem, in terms of set theory [12], [24], and graph theory [19], [20], [18]. Perhaps the most interesting of these formulations, from the TACOM location perspective, is the formulation by Roach [24] in which the objective function is to minimize the maximum within-cluster distance. Although the solution technique offered is one of forming reduced subproblems, and is somewhat improved over integer programming, it is still far too computationally complex to be of interest in the TACOM location problem.

IV. PREVIOUS APPROACHES

The TACOM location problem has received considerable attention in the data communication network literature [7], [2], [11], [25], [5], [16], [4], [27]. The approaches have almost all been heuristic in nature, and most have dealt with only the less general version of the problem where only point-to-point connections are permitted. For this simpler problem, two attractive approaches seem to be a direct adaptation of the "Add" algorithm of the warehouse location problem, and a "graceful" drop algorithm based on link removal rather than TACOM removal [2]. The former seems to be computationally considerably more efficient, but the latter to yield, in general, slightly better results. An approach based on combinatorial optimization over certain selected subsets of the network has been offered by Greenburg [16], and has the attraction of partitioning the main problem into subproblems in an iterative manner, and reoptimizing the partitions based on global considerations. The original set of node locations is partitioned by first locating the pair of nodes closest together as a kernal of a partition element. A point whose distance from the kernal gives a cost of connection to the kernal greater than the cost of locating a TACOM at the point is called an opposing point. The kernal is then iteratively enlarged by adding the point closest to the kernal which is closer to the kernal than to any opposing point. The heuristic employed here is that such points are more likely to be optimally connected to the same TACOM as the points in the kernal than to be connected to a TACOM at one of the opposing points. When no additional nodes can be added, the enlarged kernal becomes a partition member and is removed from further consideration in this initial partitioning phase. A new kernal is then selected, and the process repeated to find a second partition element. The procedure is continued until no further partitioning is achieved.

The optimal location of TACOMs within each member set of the partition is achieved by exhaustive combinatorial enumeration. However, optimality over each member set of the partition, does not necessarily give optimality over the entire set. Hence, an iterative scheme of repartitioning and reoptimizing is next pursued. Nodes are optimally assigned to the TACOMs originally found (by assigning each node to its closest TACOM), thereby creating a new partition of single TACOM sets. The optimal TACOM location within each set is then found by exhaustive search. Pairwise combinations of the sets are then optimized for one, two, or three TACOMs, again by combinatoric enumeration. If different TACOM locations result in the combined set, a new partitioning is done by optimally assigning nodes to TACOMs. The process is repeated until no pairwise revisions occur.

The difficulties here begin with the strategy for the partitioning. There is no control over the size of the partition sets, with a possible result being the original set itself. For a set of N possible TACOM locations, combinatoric enumeration of all possible locations takes 2^N-1 operations, where N is the number of locations. Each combination must have the nodes optimally assigned and its cost evaluated. Clearly, for large sets in the partition this is computationally prohibitive. The iterative process of pairwise combination of partition elements and optimizing TACOM locations over the combined pair is attractive, but the enumeration form of the optimization is again costly. All of the above difficulties are compounded considerably when extension to multidrop lines is considered.

The general TACOM location problem has been attacked by Woo and Tang [27], with an algorithm that approximates the problem with a simpler point-to-point problem, in which the point-to-point cost of connecting two nodes is a weighted average of the direct connection cost and the shared cost of connecting through the minimum spanning tree. After the problem simplification, the "Add" algorithm is applied to make site selections. Recall that in the "Add" algorithm, all candidate sites are evaluated for savings, the best selected and, with assigned nodes, removed from further consideration. The procedure is iterated until no new site is found to give positive savings. A vigorous proof of the following theorem which justifies the termination condition for the "Add" algorithm has been reported [27].

Theorem

Let L, L_1, and L_{12} be optimal network assignments corresponding respectively to TACOM sets $(j_o, j_1 \ldots, j_k)$, $(j_o, \ldots, j_i, j_{i+1})$, and $(j_o, \ldots, j_i, j_{i+1}, j_{i+2})$, and let C (L), C ($L_1$), and C ($L_{12}$) be their respective costs. Then $C(L) - C(L_1) \geq C(L_1) - C(L_{12})$.

Note that this theorem does not say that the network cost is a convex function of the number of TACOMs, but rather that the cost is convex over the iterations of adding a new TACOM to an existing set of TACOMs without reoptimization of the existing TACOM locations. In fact, counterexamples are easily produced in which the cost is not convex over the number of TACOMs when the given number of TACOMS are optimally located in each case.

The Average Tree-Direct (ATD) algorithm described above appears to be the best algorithm for the general TACOM location problem that is currently in the literature. We will use it as the basis for comparative evaluation of the algorithm we present next.

V. DESCRIPTION OF GENERAL APPROACH

There are many possible formulations for the TACOM location problem, and many possible approaches to finding acceptable solutions. We present here an approach to the following formulation of the problem:

Given:

- Set of nodes,
- Particular node which is RESCOP
- Constraint on number of nodes which may share a line,
- Constraint on number of nodes a TACOM may serve,
- Cost of connecting nodes,
- Cost of TACOM

Find:

- A low cost feasible design in which TACOMS may be used.

The object of this formulation is a total network design. The approach is heuristic, and consequently, no promise is made of optimality; only feasibility in terms of the given constraints. The constraints used in this formulation were chosen because they are often appropriate in reality, and are particularly simple. The approach is also reasonable for for various other constraints. These will be discussed later. TACOMS are used only where they appear to be beneficial, and consequently, for some node arrangements, designs will result which use no TACOMS.

The general approach is characterized by the following four steps.

1. Simplify the problem to a point-to-point problem by replacing clusters of nodes by single "center-of-mass" (COM) nodes.

2. Partition the reduced set of COM nodes by applying an add algorithm, resulting in one of the COM nodes selected as a TACOM site.

3. Select one of the original nodes as a real TACOM site in each partition by examining the original nodes closest to the COM node selected in the add algorithm, and selecting the best.

4. Apply a line-layout algorithm to each partition, with its selected TACOM site serving as the central node.

A simplified flow chart for the approach we use is shown in Figure 4. We will interpret this flow chart in terms of the four steps given above.

A. Simplification by Clustering

Simplification is achieved when the problem is reduced in size and converted to a point-to-point formulation. To accomplish this, clusters of nodes are replaced by single nodes. The clusters are intended to reflect natural groupings of nodes that can be most appropriately approximated by single nodes at their center of mass. The clusters are limited in size by the line constraint, and thus also reflect possible groupings of nodes to share a line.

The clusters are formed by "rolling snowballs" in a rather "balanced" fashion. First, the two nodes closest together are selected [2]. If these two nodes can be put in the same cluster (i.e., their being joined does not violate the line constraint) [3], they are replaced by a single node at their center-of-mass [4], called a COM node. The "weight" of the nodes are simple the number of nodes contained in the cluster they represent, with initially all weights equal to one.

If the two nodes can not be merged, the next closest together "pair" will be considered. As "pairs" of nodes are identified which can not be merged, they are removed from further consideration in the clustering process [5], although the individual nodes may reappear as members of other "pairs". The clustering process continues until no "pair" of nodes can be effectively merged [6]. An example of the clustering process is shown in Figure 5. At this point, the original set of nodes has been replaced by the set of nodes representing the clusters. This set is smaller in number by a factor slightly less than the line constraint, and the relative costs of connecting the nodes in a cluster to different sites can be approximated by the point-to-point costs of connecting their representative node. Thus, the problem is reduced in size and converted to a point-to-point form. An example of the reduced set, with associated weights for the nodes, is shown in Figure 6.

B. Partitioning

The add algorithm examines the benefit of placing a TACOM at each COM node. The benefit is determined by iteratively associating with each COM node the other COM nodes which give the greatest cost benefit by being connected to the TACOM instead of the RESCOP, subject to the TACOM capacity constraint. After the assignment of COM nodes to the TACOM, a heuristic estimate, explained in detail later, is made of the cost benefit.

The estimate incorporates the weights of the nodes, and is different than the simple cost gain of the COM as a TACOM in a point-to-point case. The COM which, as a TACOM site, has the greatest heuristic benefit estimate is selected as the best [7]. The simple point-to-point cost gain of this COM node is then checked [8]. If the cost gain is positive, the selected COM node and all those COM nodes assigned to the selected location in the add algorithm are partitioned from the remaining nodes to form a separate subproblem [9], as shown in Figure 6. If the cost gain is not positive, it is concluded that the best TACOM site is not cost-effective, and all the remaining COM nodes are then assigned to the RESCOP [13]. This forms a last partition element to be treated as a subproblem.

C. Local Optimization

The output of the partition process is a particular subset of COM nodes identified as deserving a TACOM, and a particular member of the subset identified as the appropriate site for the TACOM. However, the COM nodes are representatives of clusters, and in reality some particular original node site must be selected for the TACOM. To select this site, each of the k nodes closest to the selected COM node are evaluated, as shown in Figure 7, with $k = 3$.

The same measure is used in the evaluation as is used in the COM node selection. The node with the greatest heuristic estimate of benefit is then selected as the actual TACOM site [10]. This gives a partition of the nodes into a subproblem complete with an actual node site chosen for the TACOM.

D. Line Layout

In order to apply a line layout algorithm to the subproblem formed above, it is first necessary to replace all the COM nodes by the actual nodes they represent. This gives a partition of actual nodes with an actual node selected as the TACOM site. A line layout algorithm is then applied to the partition [11], giving a result as shown in Figure 8. The nodes in this partition are then removed from further consideration. If no nodes are left [12], the design is complete [14]. If nodes remain, the process is repeated [15]. The process is repeated until all nodes are assigned to TACOMS or until no no additional TACOM is estimated to be beneficial [8]. In this latter case, the remaining nodes are assigned to the RESCOP [13], and the line layout algorithm is applied [11] to complete the design, as shown in Figure 8.

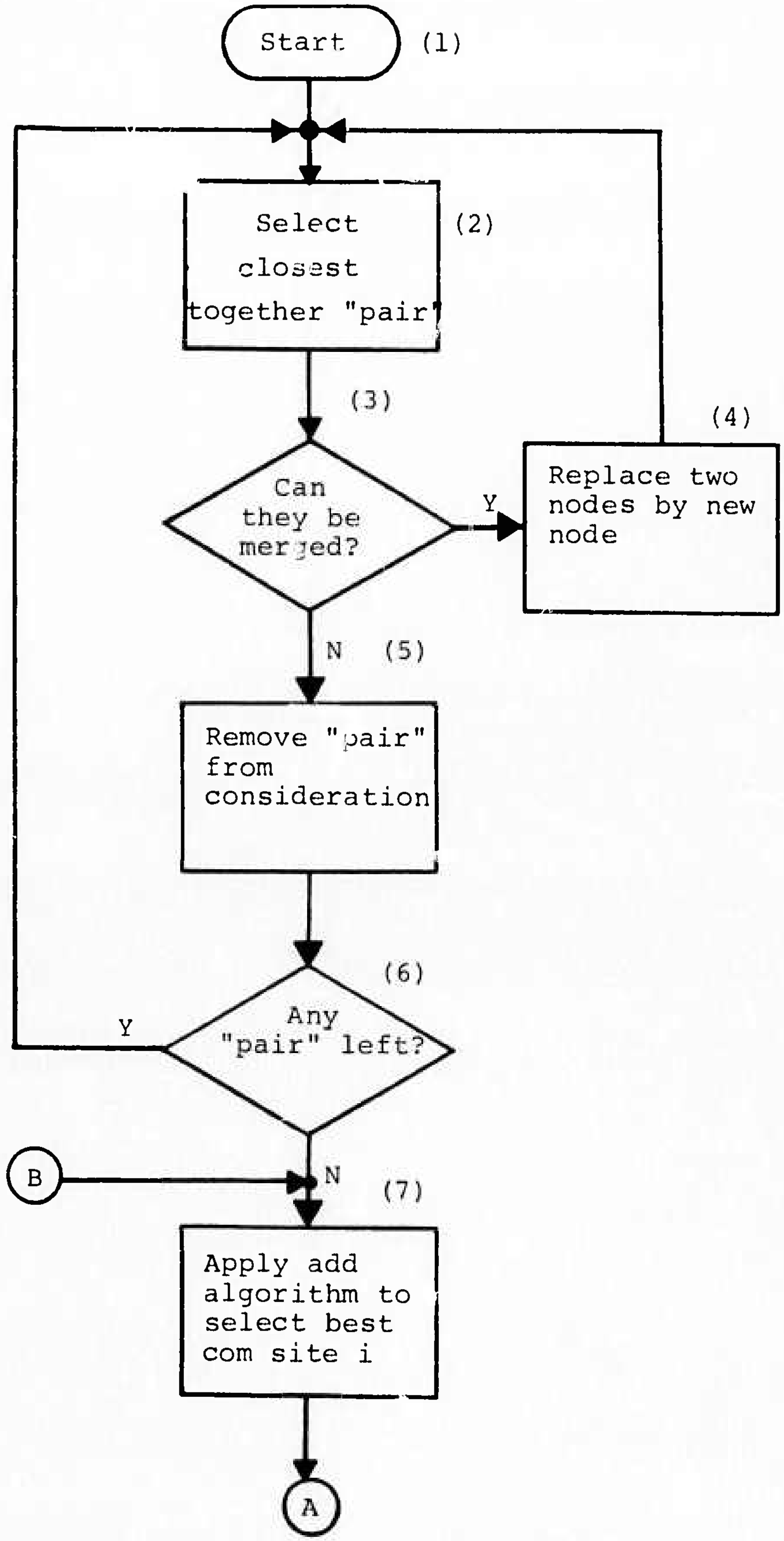
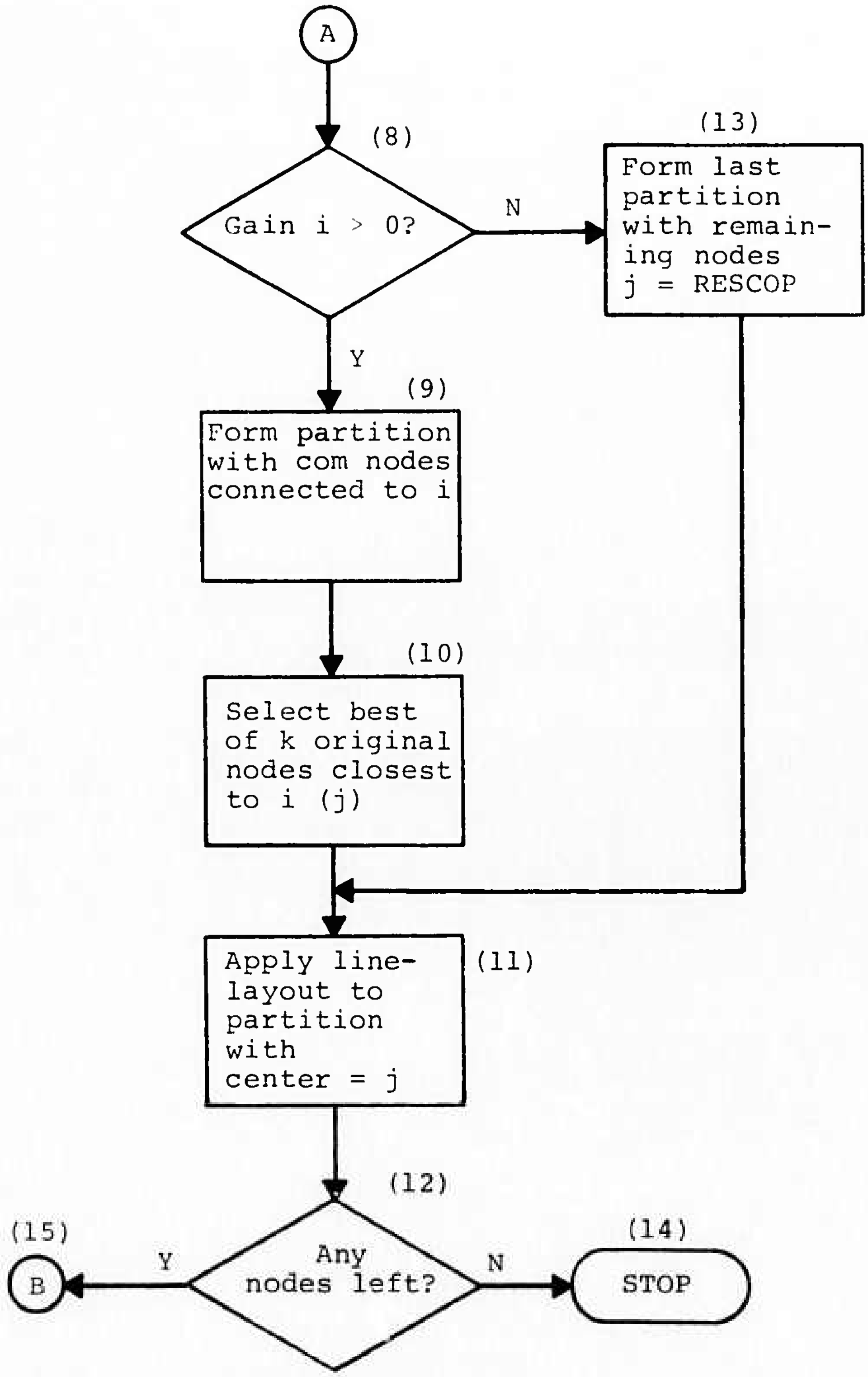


FIGURE 4: FLOW CHART OF APPROACH



(FIGURE 4 CONTINUED)

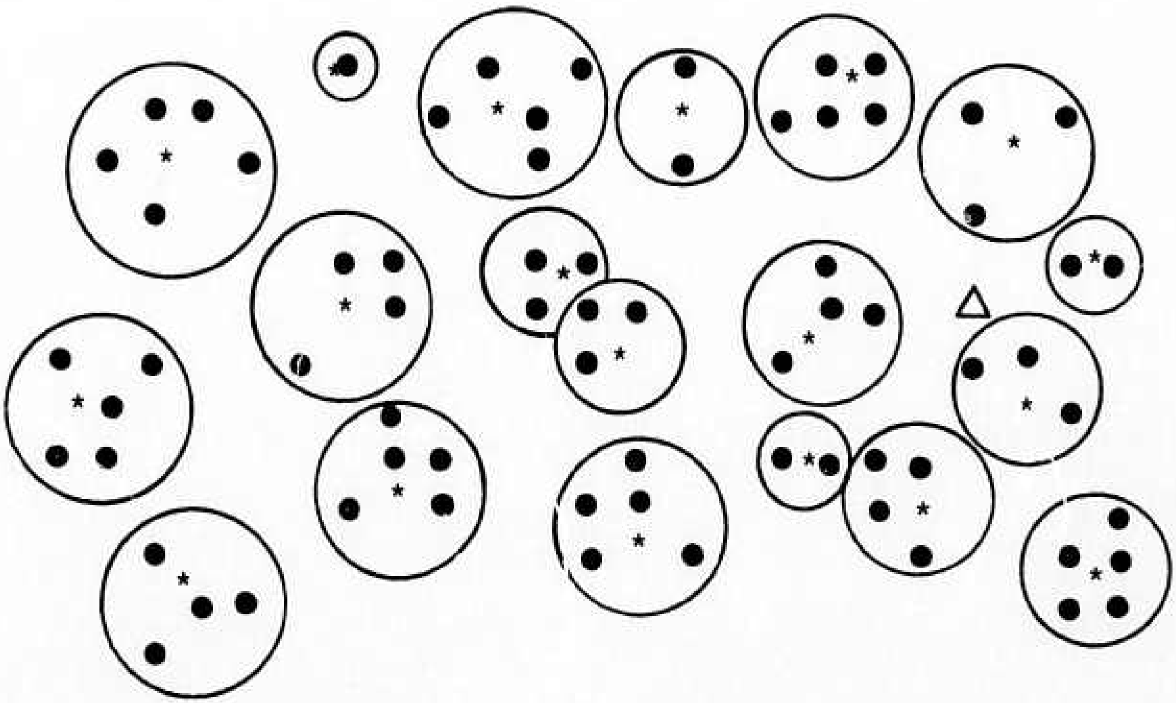


FIGURE 5: SIMPLIFICATION BY CLUSTERING

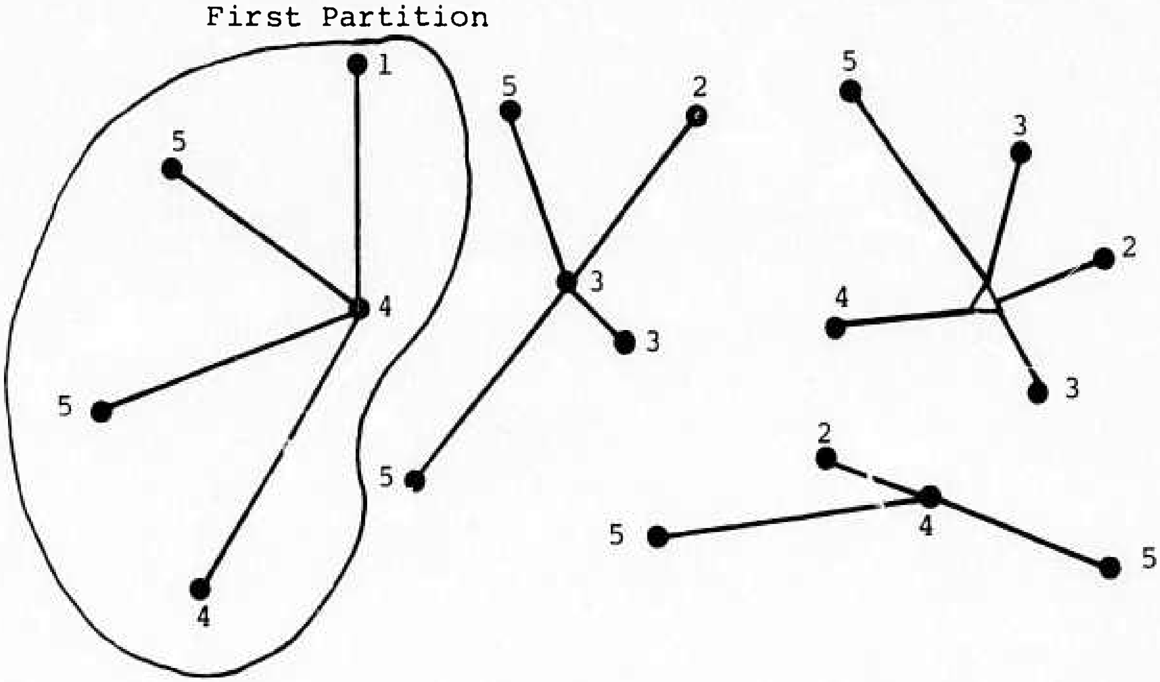


FIGURE 6: PARTITION BY ADD ALGORITHM

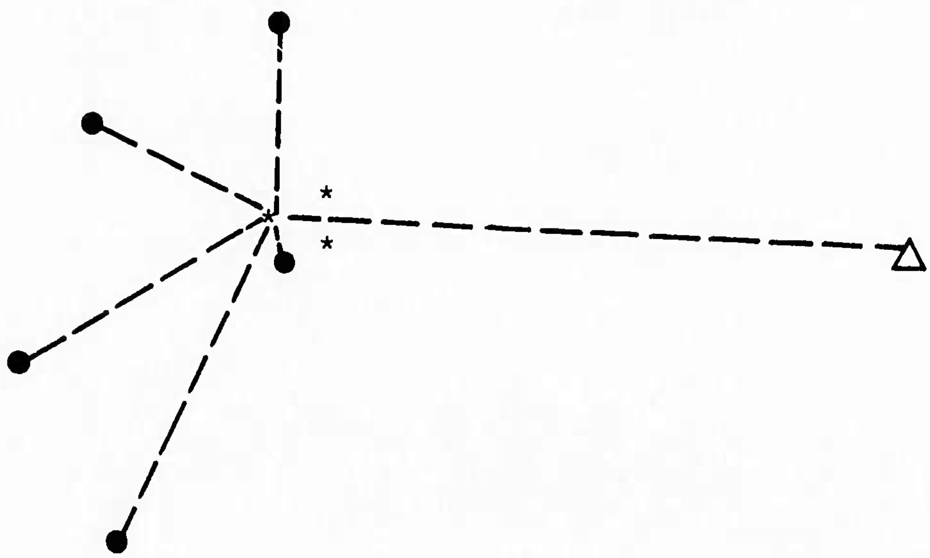


FIGURE 7: LOCAL OPTIMIZATION

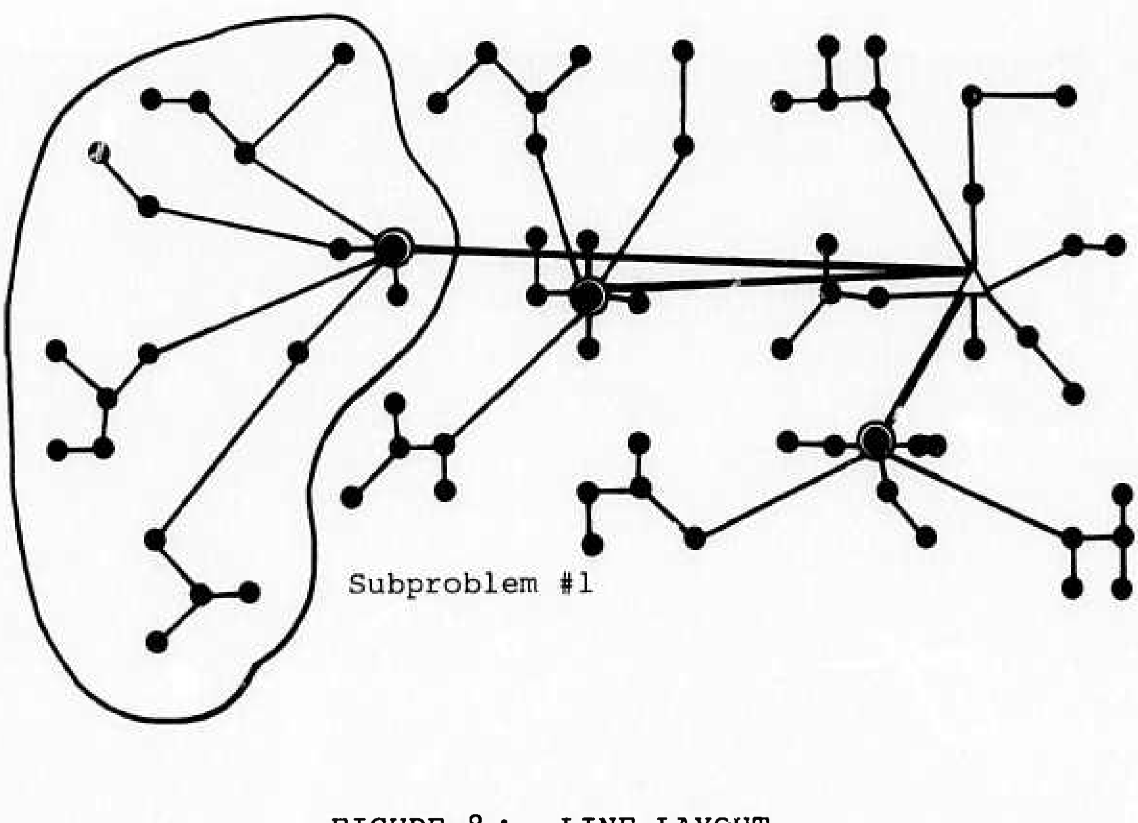


FIGURE 8: LINE LAYOUT

VI. THE CENTER-OF-MASS ALGORITHM

In this section, we give a precise description of the algorithm outlined above. The algorithm is formally stated at the end of the section. The first part of the section is devoted to interpretating the formal statement.

The formal statement begins with a list of the relevant definitons, parameters, constants, constraints, and cost function. Note that the constraints are of the simple capacity variety, and the cost function for connecting two nodes is simply the Euclidean distance between the nodes. It will be clear after examination of the algorithm that other constraints and cost functions are equally usable.

In the initialization phase (Step 0), A is formed as the set of N - 1 nodes of interest, where the RESCOP (node 1) is not included. The RESCOP is not considered eligible for a TACOM as nodes may connect to it directly. The set D0 is the set of all nearest neighbor distances. For each node i a list L_i is kept of the real nodes represented by the node. Initially, the list for each node contains only the node itself. After two nodes merge, the list for the new node will contain the members of the lists of the two nodes that merged, thereby keeping track of the represented nodes. The weight assigned to each node is initially set equal to one. Thus, the constraints are simply on the number of nodes per line and the number of nodes per TACOM. For each node i, its nearest neighbor j is represented as n_i. The average "nearest neighbor distance", d_{avg}, is used to set a maximum distance d_{max} over which nodes may be merged. The parameter α determines this maximum distance. The reason for establishing such a maximum distance is based on the clustering objective and procedure, as discussed below.

The clustering objectve is to replace natural groupings of nodes ~~by a single~~ node. The procedure is to iteratively merge the closest feasible pair of nodes. As this procedure nears completion, most nodes are representative of near capacity clusters, and consequently, cannot be merged. Thus, the shortest distance between feasible nodes may become quite large. Merging of these nodes would not reflect the objective of natural groupings. The maximum distance is designed to prevent such mergings.

After determining a maximum allowable distance, the set D is formed as the set of allowable nearest neighbor distances.

The merging of two nodes is accomplished in Step 1 of the algorithm. Prior to entering this step, it is known that all nodes with their nearest neighbor distance contained in D can be feasibly merged with their nearest neighbor. When Step 1 is entered after the initialization process, this is certainly true, or else the problem is point-to-point in form, and not multipoint. Following Step 1 is an update procedure designed to ensure that this is true when the step is reentered.

The first task in the merge process is selection of the minimum nearest neighbor distance. Then a new node is formed with its location at the center-of-mass of the two nodes being merged, where the weights used in the center-of-mass calculation are simply the weights assigned to each node. The list of real nodes represented by the new node is formed by merging the lists of the two old nodes. The old nodes are then removed from further consideration by deleting them from the set A. The new node will be added to the set A later. The weight of the new node is simply the sum of the weights of the old nodes. Note that with the initial weights all set equal to one, the weight is simply the number of real nodes represented by the new node.

After a new node is formed, its nearest neighbor must be found, and all nodes which had one of the old nodes as a nearest neighbor must also have new nearest neighbors found. This is accomplished in the update procedure (Step 2). D1 is the subset of D containing all the distances for which the nearest neighbors are not one of the two nodes just merged in the previous step. A2 is the set of all nodes whose nearest neighbors were one of the two nodes just merged, plus the new node; i.e., the set of all nodes for which new nearest neighbors are to be found. The new node is added to the set A, making it again the set of all nodes which are candidates for merging. The nearest feasible neighbor distances are then found for the nodes which need new distances, i.e., the members of A2. This is the set D2. D3 is the subset of D2 containing distances less than the maximum allowed distance. For each node i, which has a new feasible, allowable distance neighbor j, n_i is defined as j. The set D of all allowable nearest neighbor distances is then formed by combining the unchanged distances, D1, with the new distances D3. Note that all members of D are defined for pairs of nodes which are feasible to merge. If D is not empty, then the merging process is continued. If D is empty, no further merging is possible, and the add process is entered (Step 3).

The first task of the add algorithm is the iterative examination of each node for the possible benefit of locating a TACOM at its site. This task is accomplished in Step 3. The savings achieved by connecting a node j to a TACOM at i versus the RESCOP (node 1) is defined as s_{ij}. For each node i, nodes are iteratively associated with the node in order of decreasing savings, with the maximum savings node associated first. The iterations continue until no further savings are possible, or until the capacity constraint prevents any further associations. When the iterative process is terminated, the benefit of placing a TACOM at the node site is evaluated on the basis of the associated nodes. Thus, for each node, i, the iterative process is initiated by forming Λ as the set of candidate nodes for connection to i, and B_i as the list of nodes actually to be associated (initially empty). T is the running sum of weights of the associated nodes, used to check the capacity constraint.

Part A) is the iterative process of association. If all candidates have been examined, then the evaluation, Part B), is commenced. If there are still unchecked candidates, select the one with the greatest savings. If the savings are not positive, no further savings are possible, and commence the evaluation. Otherwise, remove the selected node from the candidate set to ensure that it is not selected a second time, and then determine if the association is consistent with the constraint. If not, go on to the next candidate. If the association is feasible, add the node to the association list, and add its weight to the running sum. If the sum is equal to the capacity, no further associations are possible, and the evaluation process is commenced. If the sum is less than the capacity, additional associations may be made, and thus, return to consider the next candidate.

Part B) is the evaluation process. First, the point-to-point savings obtained by placing a TACOM at the node are determined. This is simply the sum of all the individual savings found for the associated nodes, minus the cost of connecting the TACOM to the RESCOP and the cost of the TACOM. Then a relative multipoint benefit is calculated. This is a heuristic measure of the node as a site for a TACOM considering the multipoint nature of the problem. The measure is calculated as the sum of the weighted savings minus an emphasized cost of connecting the TACOM to the RESCOP. The weights serve to move TACOMS towards the bigger clusters, and the emphasis parameter, γ, serves to move TACOMs towards the RESCOP.

After evaluating each node as a possible TACOM site, the one with the greatest heuristic measure of benefit is selected (Step 4). If the point-to-point savings for this site are not positive, then it is predicted that an additional TACOM will not be cost-effective, and the final stage, Step 5, is then entered. This stopping condition has its foundation in the theorem stated earlier which characterizes the costs resulting from the intermediate steps of the "Add" algorithm as a convex function of the addition of new TACOMS. If the selected site is found to be cost-effective on a point-to-point basis, then the process of forming a subproblem is commenced.

First, the set of center-of-mass nodes associated with the selected site are removed from further consideration in the main problem. Then a partition element, P, is formed as the set of real nodes represented by all the center-of-mass nodes associated with the selected TACOM site. Since a real node must be selected for the actual TACOM site, a local optimization procedure is used to evaluate possible real node sites. The set K is formed as the k real nodes closest to the center-of-mass node selected as the TACOM site. A heuristic measure, z_ℓ of relative cost for each of these nodes is then evaluated, and the node with the minimum cost measure is selected. Thus, a partition element of real nodes and selected TACOM site has been formed as a subproblem for the line-layout algorithm. If additional center-of-mass nodes remain, the next iteration of the add algorithm is commenced. Otherwise, we are done.

When center-of-mass nodes still exist in the consideration set, A, but TACOMS are predicted to not be cost-effective, then the remaining nodes are associated with the RESCOP site. The line-layout for the real nodes represented by these center-of-mass nodes is the only remaining problem, and is handled in Step 5.

We note that in Step 2 all nodes which have their nearest neighbor distance more than the allowed maximum are excluded from further consideration by deleting their distance from the set D. In fact, the merging of two nodes may eventually lead to a new nearest neighbor with acceptable distance, as shown in Figure 9. However, this appears to be sufficiently rare and inconsequential to warrant exclusion in favor of reducing the computational burden. For completeness, an alternate Step 2, which includes this case, is shown.

COM ALGORITHM

Definitions:

A = Set of nodes.

L_i = Set of nodes associated with $i \varepsilon A$.

x_i, y_i = Coordinates of node i.

$d_i(j)$ = Cost of connecting j to i.

w_i = Weight assigned to node i.

Parameters:

γ - Used to emphasize proximity of possible TACOM site to RESCOP.

α - Used to limit distance over which nodes may be merged.

k - Number of real nodes nearest to the merged node which are to be examined as possible TACOM sites.

Constants:

charge - Cost of TACOM.

w_{max} - Line capacity.

c_{max} - TACOM capacity

node 1 - RESCOP

Constraints:

A) For a line shared by the nodes $i = 1, 2, \ldots, M$ to be feasible;

$$\sum_{i=1}^{M} w_i \leq w_{max}$$

B) For a TACOM serving the nodes $i = 1, 2, \ldots, M$ to be feasible;

$$\sum_{i=1}^{M} w_i \leq c_{max}$$

Cost Function:

$$d_i(j) = ||i\,,j|| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

STEP 0: (Initialization)

$$A = \{i \mid i = 2, 3, \ldots, N\}$$

$$D0 = \{d_i(j) \mid d_i(j) = \min_{\substack{\ell \varepsilon A \\ \ell \neq i}} d_i(\ell),\ i = 2, 3, \ldots, N\}$$

for $i = 2, 3, \ldots, N$

$$L_i = \{i\}$$

$$w_i = 1$$

for each $d_i(j) \ \varepsilon\ D0, n_i = j$

$$d_{avg} = \frac{1}{N-1} \sum_{D0} d_i(j)$$

$$d_{max} = \alpha * d_{avg}$$

$$D = \{d_i(j) \mid d_i(j) \ \varepsilon\ D0,\ d_i(j) \leq d_{max}\}$$

STEP 1: (Merge)

$$d_\ell(n_\ell) = \min_D d_i(n_i)$$

Form a new node k with;

$$x_k = \frac{(w_\ell * x_\ell + w_{n_\ell} * x_{n_\ell})}{w_\ell + w_{n_\ell}}$$

$$y_k = \frac{(w_\ell * y_\ell + w_{n_\ell} * y_{n_\ell})}{w_\ell + w_{n_\ell}}$$

$$L_k = L_\ell \cup L_{n_\ell}$$

$$A = A - \{\ell, n_\ell\}$$

$$w_k = w_\ell + w_{n_\ell}$$

STEP 2: (Update)

$D1 = D - \{d_i(n_i) \mid n_i \in \{\ell, n_\ell\}\}$

$A2 = \{i \mid i \in A,\ n_i \in \{\ell, n_\ell\}\} \cup \{k\}$

$A = A \cup \{k\}$

$$D2 = \{d_i(j) \mid i \in A2,\ d_i(j) = \min_{\substack{\ell \in A \\ \ell \neq i \\ w_i + w_\ell \leq w_{max}}} d_i(\ell)\}$$

$D3 = \{d_i(j) \mid d_i(j) \in D2,\ d_i(j) \leq d_{max}\}$

For all i,j such that $d_i(j) \in D3$, $n_i = j$

$D = D1 \cup D3$

If $D \neq \emptyset$, then go to Step 3.

STEP 3: (Evaluate Each Site)

Let $s_{ij} = d_j(1) - d_j(i)$

For each i;

$\Lambda = A$

$B_i = \emptyset$

$T = 0$

A) If $\Lambda = \emptyset$, then go to B)

Select ℓ such that $s_{i\ell} = \max_{j \varepsilon \Lambda} s_{ij}$

If $s_{i\ell} \leq 0$, then go to B)

$\Lambda = \Lambda - \{\ell\}$

If $w_\ell + T > C_{max}$, then go to A)

$B_i = B_i \cup \{\ell\}$

$T = T + w_\ell$

If $T = C_{max}$, then go to B)

Go to A)

B) $s_i = \sum_{j \varepsilon B_i} s_{ij} - d_1(i) - \text{charge}$

$r_i = \sum_{j \varepsilon B_i} s_{ij} * w_i - d_1(i) * \gamma$

STEP 4: (Select Best)

$$r_\ell = \max_{i \varepsilon A} r_i$$

If $S_\ell \leq 0$, then go to Step 5.

$$A = A - B_\ell$$

$$P = \bigcup_{j \varepsilon B_\ell} L_j$$

$$K = \{i_n \mid i_n \varepsilon P,\ n = 1, \ldots, k,\ d_\ell(i_n) \leq d_\ell(i_m),\ i_m \not\varepsilon K\}$$

$$Z_\ell = \min_{i \varepsilon k} \left[\sum_{j \varepsilon B_\ell} d_i(j) * w_j + d_1(i) * \gamma \right]$$

$$i_c = \ell$$

Do line layout on P with i_c = TACOM site.

If $A \neq \emptyset$, then go to Step 3.

ELSE STOP.

<u>STEP 5</u>: (Finish)

$$P = \bigcup_{j \in A} L_j$$

$$i_c = 1$$

Do line layout on P with i_c = RECOP site.

STOP.

ALTERNATE STEP 2: (Alternate Update)

$A = A \cup \{k\}$

$$D1 = \{d_i(j) \mid i \in A,\ d_i(j) = \min_{\substack{\ell \in A \\ i \neq \ell}} d_i(\ell),\ w_i + w_\ell \leq w_{max}\}$$

$$D = \{d_i(j) \mid d_i(j) \in D1,\ d_i(j) \leq d_{max}\}$$

If $D = \emptyset$, then go to Step 3.

Else for each i,j such that $d_i(j) \in D$, $n_i = j$.

Go to Step 1.

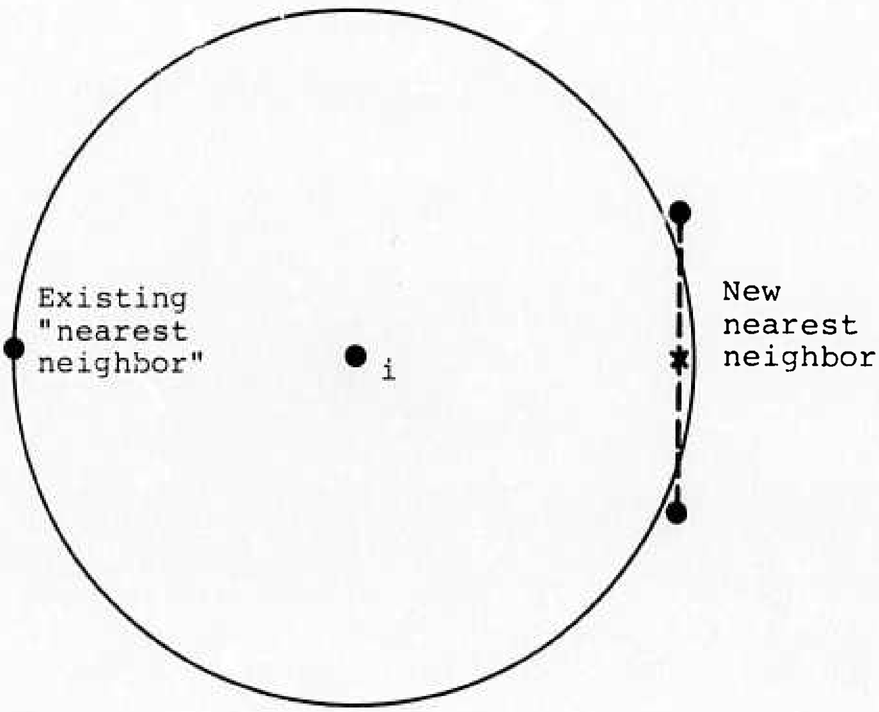


FIGURE 9: NEW NEAREST NEIGHBOR

VII. PERFORMANCE RESULTS

The COM Algorithm is a heuristic approach to a rather complex problem. In order to evaluate its performance, we have implemented the algorithm and applied it to a number of problems with randomly positioned nodes. The results of these experiments are reported below. For all experiments, the implementation was in Fortran, and the experiments conducted on a CDC 6600 computer. Each of the problems were of the simple type described earlier, with constraints on the number of nodes per line and nodes per TACOM, and one preset RESCOP site to which all nodes must be connected, either through TACOMs or directly through their multipoint lines. The cost function was a telpak rate of $.50 per mile and $40 per drop as a monthly charge. In each experiment, the values chosen for the program parameters were held fixed for all cases; no attempt was made at "fine tuning".

In order to have a comparison basis for the evaluation, we implemented the best other heuristic algorithm for this problem that we could find in the literature. This was the approach of converting the problem to a point-to-point problem by forming a cost matrix based on the average of the cost of connecting two nodes through the minimum spanning tree and the cost of connecting them directly, as reported by Woo and Tang [27]. The implementation of this algorithm (which we call the Average Tree-Direct (ATD) algorithm), was also in Fortran on a CDC 6600, and used the same line layout procedure used in the COM implementation. In order to ensure that our implementation of this algorithm was reasonable for a comparison base, we applied the algorithm to a problem of four hundred nodes distributed in a random manner based on population densities (the distribution technique will be explained in detail later). With TACOMs having a cost of $25000 per month and a capacity of 100 nodes, and the RESCOP site located in Atlanta, the resulting design is shown in Figure 10. We feel the problem is quite comparable to the example Woo and Tang reported. The running time on the CDC 6600 was approximately 160 seconds, whereas the time they reported for running on a 360/91 was approximately 60 seconds. However, in our implementation we used a true combination of weighted tree distance and direct distance, whereas in their example it appears the combination was approximated with only a weighted direct distance. Given also that their design resulted in only three TACOMs, and our's four, thus requiring more time in the add algorithm phase, and the relative computing power of the two machines for such problems being slightly more than 2:1 (in favor of the 91), the implementation appears reasonably comparable In the ATD algorithm the cost of connecting two nodes i and j is approximated as

$$cost_{ij} + \alpha t_{ij} + \beta d_{ij}$$

where d_{ij} is the direct connection cost, t_{ij} is the tree cost, and α and β are parameters with $0 \leq \alpha, \beta \leq 1$. The tree cost is defined as follows:

Let $\{b_h\}$ be the set of links (or branches) in the minimal spanning tree; for every b_h define $B_j\ (b_h)$ as the set of nodes disconnected from j upon removal of b_h from the minimal spanning tree, and let $|B_j\ (b_h)|$ be the number of such nodes. Then

$$t_{ij} = \sum_{b_k \epsilon \pi_i} \frac{d_k}{|B_j(b_k)|}$$

where d_k is the cost of link b_k, and π_i is the unique path from j to i consisting of a sequence of branches b_k.

The values for the parameters α and β used in our implementation were determined by optimizing for a pilot problem, and then held fixed during all the experiments. Our experience with the pilot problem supported the observation by Woo and Tang that the design results are rather insensitive to variations in the parameters around the optimum values. We now present descriptions and results of the evaluation experiments.

A. Uniformly Randomly Distributed Nodes

The first series of experiments were performed on problems where the nodes were uniformly randomly distributed over a 2000 by 3000 mile rectangle. The algorithm was applied to problems of 50, 100, 200, and 400 nodes. The node distributions for these problems are shown in Figures 11, 12, 13 and 14. The minimum spanning trees are shown connecting the nodes. Three RESCOP sites were considered, labeled as 1, 2, and 3 in the figures.

The first experiment used constraints of four nodes per line and twenty nodes per TACOM; with a TACOM cost of $200 per month. Problems of 50, 100, and 200 nodes were considered, with the cost results shown in Table 1. In eight out of the nine comparisons, the COM algorithm had the lower cost, with an average improvement of 3.1% ((ATD-DOM)ATD). A typical COM design is shown in Figure 13.

In the second experiment, the same nine problems were considered, but with a TACOM cost of $1500/month. The results are shown in Table 2. The COM algorithm produced the lower cost designs in the same eight out of nine cases as before, with an average improvement of 3.8%. A typical COM design for this experiment is shown in Figure 14. The same problem is shown as before; but note the reduction in number of TACOMS due to the increase in their cost.

In the third experiment, the constraints were changed to 10 nodes per line and 50 nodes per TACOM. The TACOM cost was fixed at $1500, and problems of 100, 200, and 400 nodes were considered. The results are shown in Table 3. In all cases the COM algorithm produces the lower cost designs, with an average improvement of

7.2%. A typical design is shown in Figure 15. The average costs of the designs produced by the COM algorithm as a function of the number of nodes is shown in Figure 18. Note that, as would be expected, the larger capacity constraints not only give lower cost designs,but also a less rapidly growing cost-curve.

B. Randomly Distributed Based on Population

In most real problems, the nodes will not be uniformly distributed over a nice rectangular region. In order to pose a more realistic problem, nodes were located throughout the United States in a random manner based on population density. A weight of 1000 was divided among the 238 most populated cities (i.e., those with populations greater than 50,000) in proportion to their population. A rectangular region was determined for each city, or collection of cities, to reflect the feasibility of the region to support a population segment with access to urban facilities. Thus, consideration was given to natural geographical boundaries, such as mountains, lakes and coast lines, to major roads in the area, to the number of nearby smaller communities, and to the natural pattern of urbanization between relatively close major population centers. Using this approach, 123 regions were defined, with varying sizes of approximately 70 miles square.

Once a number of nodes has been allocated to a region in proportion to population, the geographic positions of the nodes within the region are uniformly randomly distributed. With a large number of nodes, it is reasonable to anticipate that some of them may be located at points with no discernable geographic significance. Therefore, a fraction of the nodes were located at random in a large geographic segment; east of Denver, west of Pittsburgh, north of Austin, and south of Mikwaukee. In the two experiments reported below, problems of 400 nodes were considered, with 5% distributed in the large region.

In each experiment, the RESCOP site was A lanta, the TELCOM cost was $2500 per month, and the line constraint was ten nodes per line. With a TACOM constraint of 100 nodes per TACOM, the ATD algorithm produced the design shown in Figure 10, with a cost of $41, 508.50/month, and the COM algorithm produced the design shown in Figure 19, with a cost of $39,418.50/month. In this case the COM algorithm produced a design lower in cost by 5%.

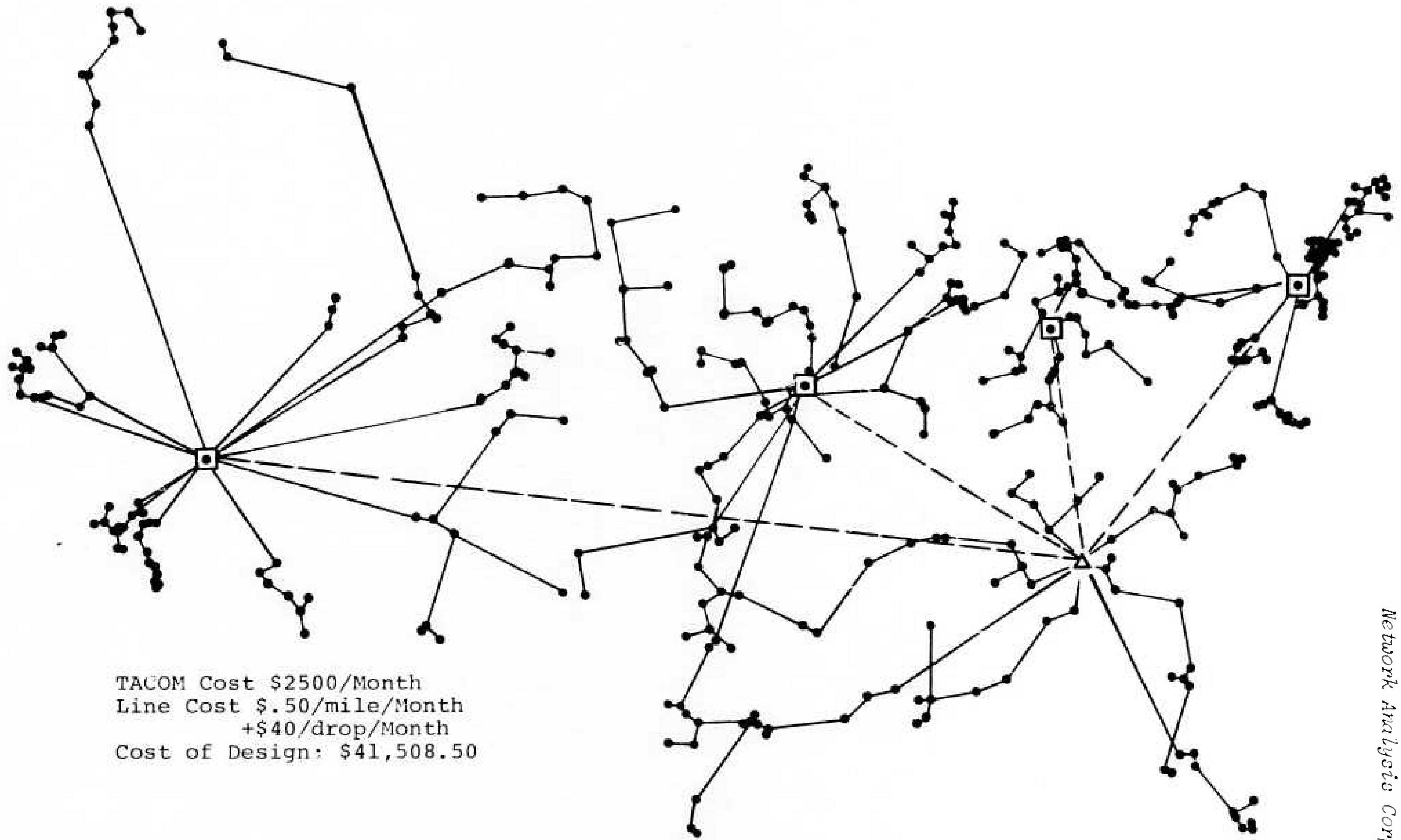
As a final experiment, we applied the algorithms to the same problems as described above, but with no constraint on the TACOM capacities. The designs they produced are shown in Figures 20 and 21. As can be seen from the figures, the designs are very similar. The COM algorithm produced a lower cost design by only .50%. This result, coupled with the others, suggests that the COM algorithm is perhaps more sensitive to TACOM capacity. In fact, in the 27 comparison cases examined, the COM algorithm had fewer TACOMS in 18 cases, and the same number in the remaining nine cases. On the average, it used 21% fewer TACOMS $[(N_{ATD}-N_{COM})/N_{ATD}]$.

C. <u>Computation Time</u>

A significant factor for a design algorithm is its efficiency, i.e., the computer time it requires. In order to appraise this attribute of the COM algorithm, the execution time for the algorithm was measured on several problems. The times are only for the TACOM selection portion of the problem, and not the line layout portion. For comparison purposes, the execution times for the ATD algorithm were also measured in the same way. The same basic strategies for efficiency were used in each implementation. For the problems with constraints of 10 nodes per line and 50 nodes per TACOM, and TACOM cost of $1500 per month, the average results are shown in Table 4. Curves portraying these results are shown in Figure 22. It would appear that the COM algorithm is substantially more efficient. To quantify this comparison, the two curves are shown on a log-log scale in Figure 23. From these curves, the execution time of the COM algorithm for these problems may be approximated by the function $t = (2 \times 10^{-4})N^2$, and the ATD al orithm by the function $t = (10^{-4})N^{2.5}$.

The execution time was measured in the same way for the problems using the population model. With a constraint of 100 nodes per TACOM, and TACOM cost of $2500 per month, each algorithm used fewer TACOMs in its design than in the above uniformly distributed problems, and the execution time dropped accordingly. Thus, for the COM algorithm, the time was 32.8 seconds, and for the ATD algorithm the time was 161.8 seconds. With no constraints on the TACOMs, again fewer TACOMs were found, and again the times were reduced, to 32.0 for the COM algorithm, and 121.6 for the ATD algorithm. Note that the decrease in the COM time is much less dramatic than for the ATD algorithm. This can easily be interpreted as due to the time spent in the add algorithm phase. With its simplification of the problem to a reduced number of nodes, the COM algorithm spends much less of its time in the add phase than does the ATD algorithm, which considers all the nodes in this phase. Thus reduction of this part of the problem with greater TACOM capacity will have a much more dramatic effect on the ATD algorithm than on the COM algorithm. Note that this last case with the add phase used to select only two TACOMs will be one of the worst comparative cases for the COM algorithm.

The basic execution time advantage for t   OM algorithm results from its problem simplification strategy. The merging process can be implemented with only slightly more complexity than a basic kruskal minimum spanning tree algorithm. Howerver, the ATD algorithm involves considerable computation to determine the equivalent tree cost of connecting each pair of nodes in addition to generation of a minimum spanning time. Furthermore, as noted above, the results of the simplification is not only conversion to a point-to-point problem, as is true for both algorithms, but also reduction of the number of nodes to be used in the "Add" phase for the COM algorithm, whereas the ATD algorithm has no such reduction.



TACOM Cost $2500/Month
Line Cost $.50/mile/Month
+$40/drop/Month
Cost of Design: $41,508.50

FIGURE 10: AVERAGE OF TREE COST AND DIRECT COST ALGORITHM APPLIED TO 400 NODES DISTRIBUTED IN A RANDOM MANNER BASED ON POPULATION.

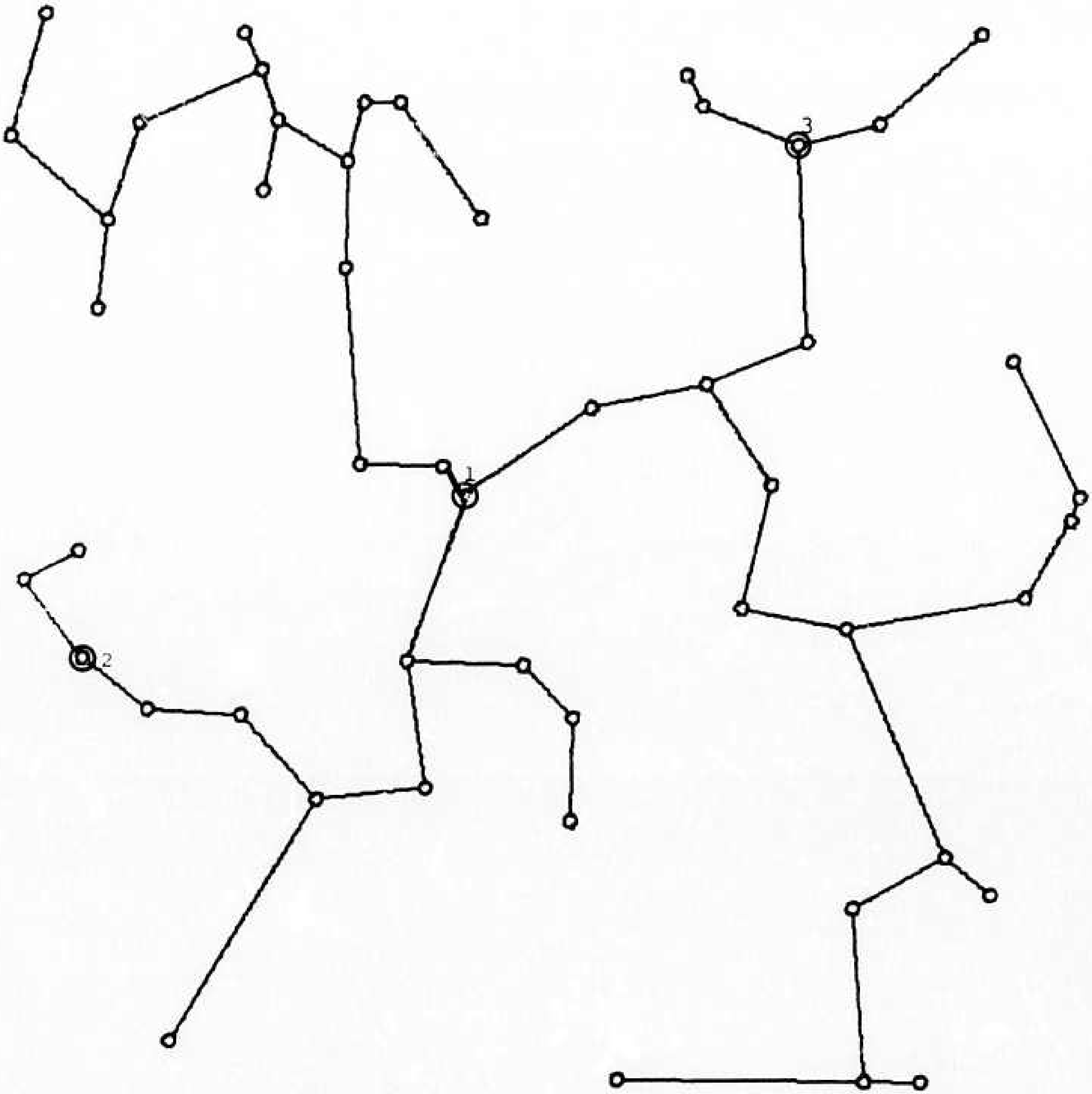


FIGURE 11: 50 NODES RANDOMLY DISTRIBUTED ON THEIR MST

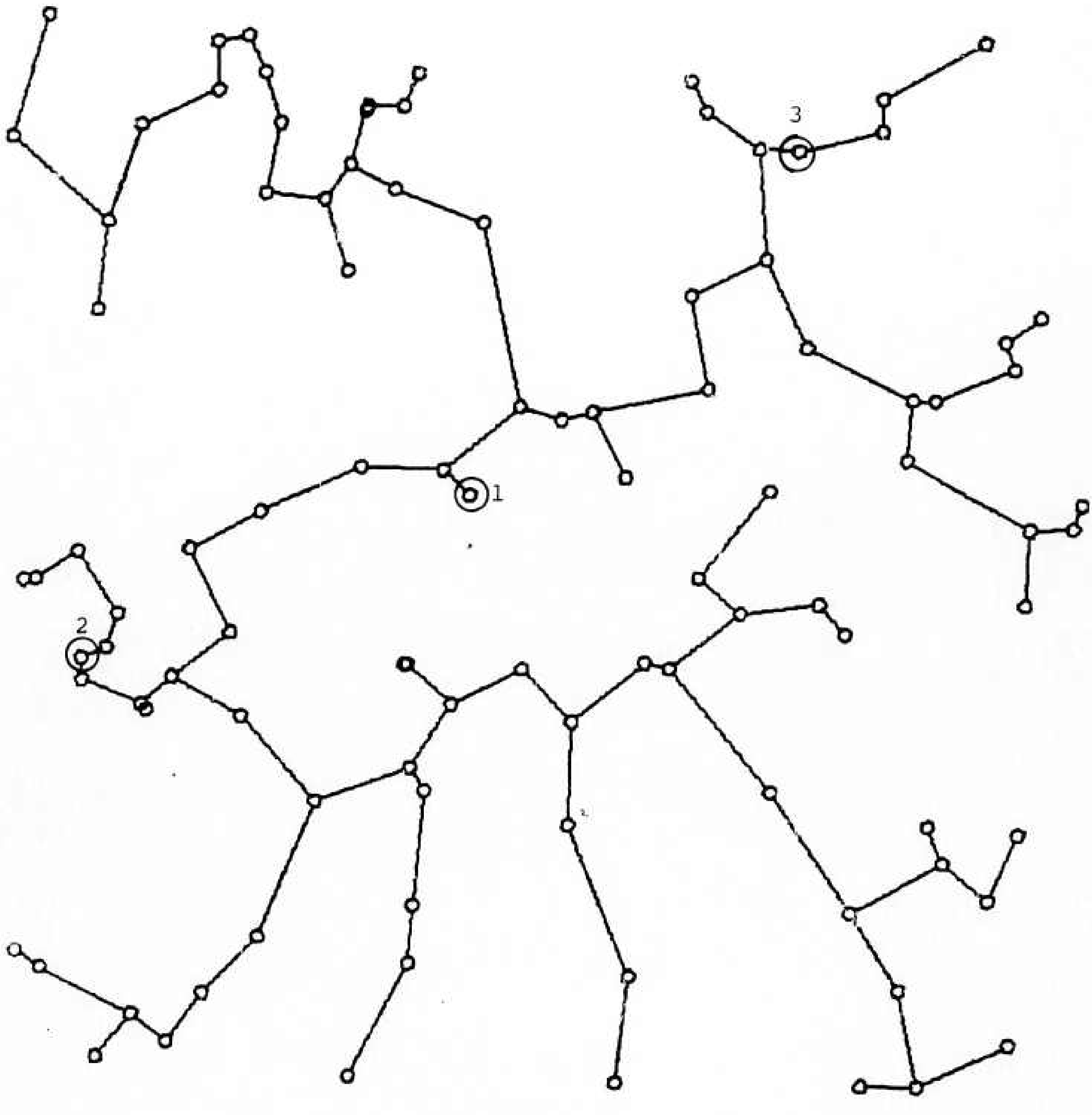


FIGURE 12 : 100 NODES RANDOMLY DISTRIBUTED
AND THEIR MST

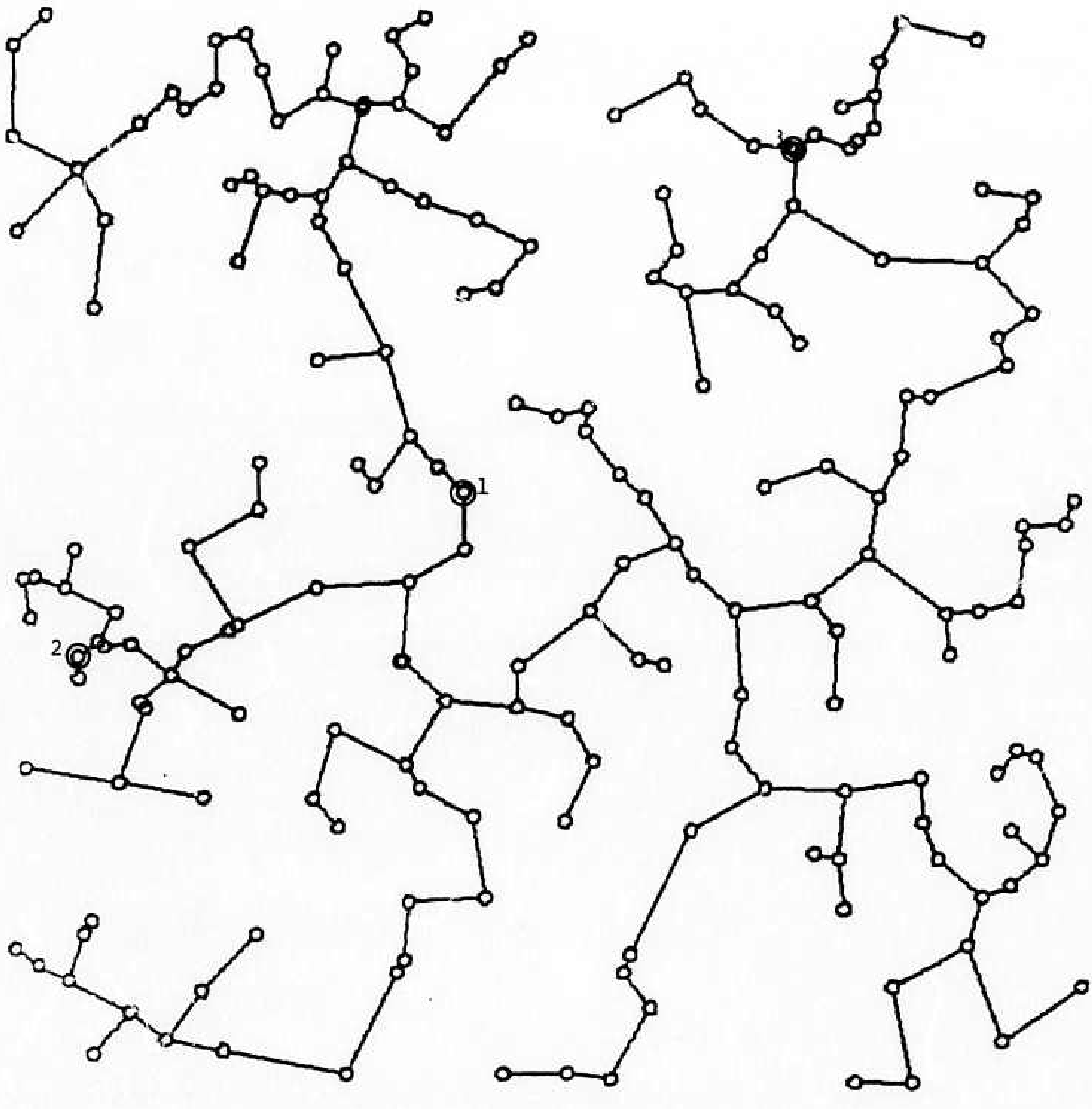


FIGURE 13 : 200 NODES RANDOMLY DISTRIBUTED ON THEIR MST

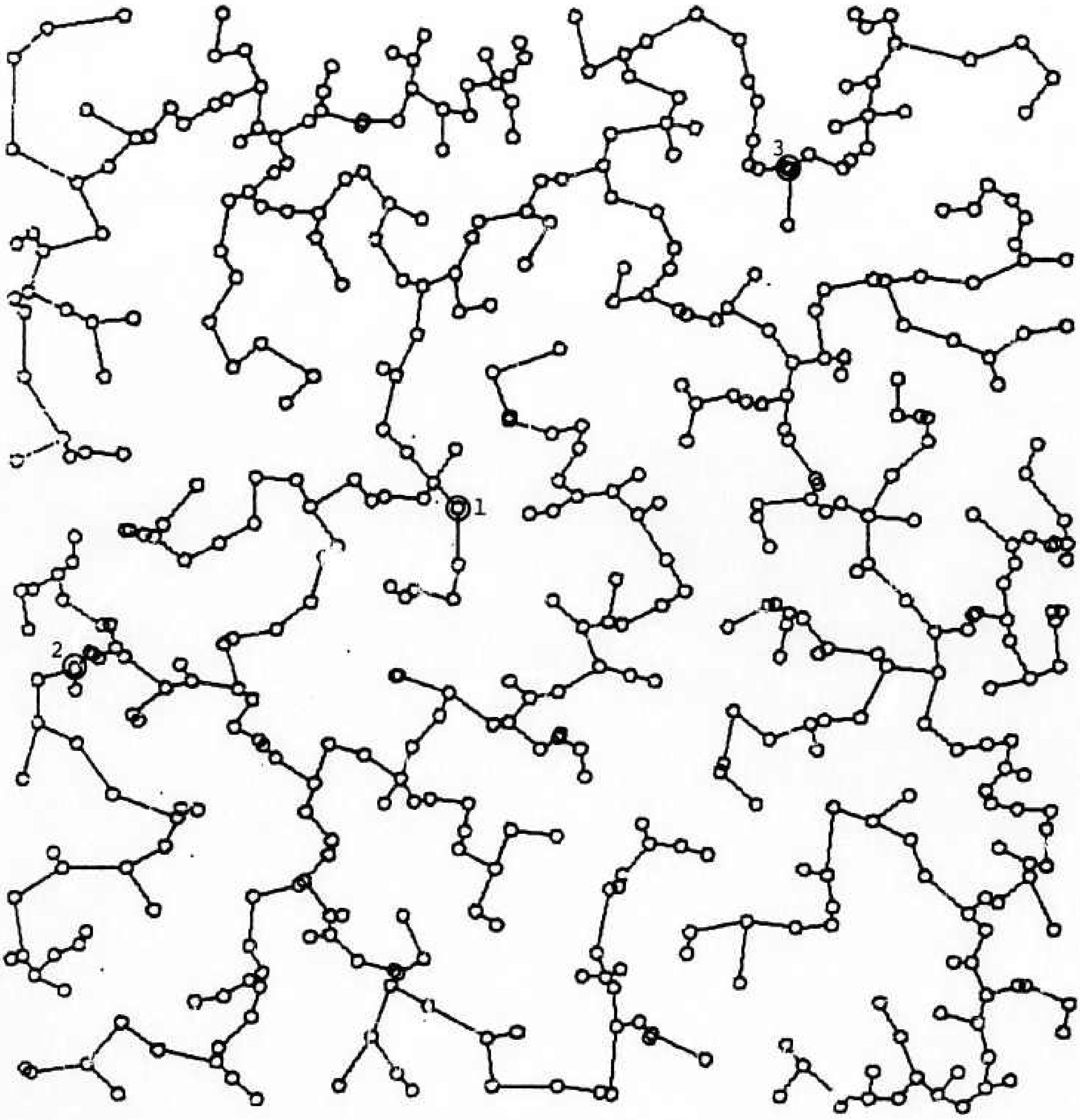
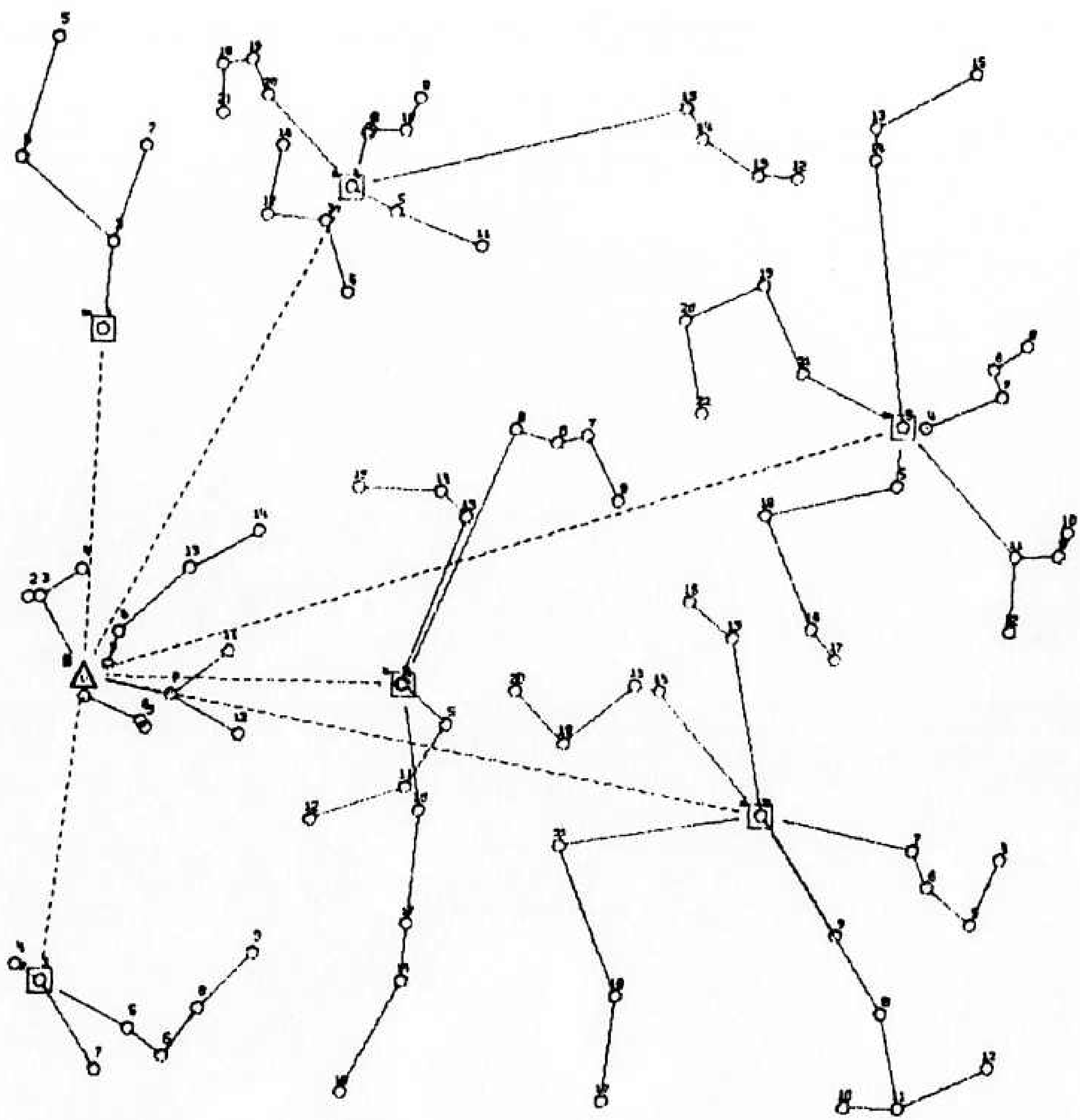


FIGURE 14 : 400 NODES RANDOMLY DISTRIBUTED AND THEIR MST



4 nodes / line
20 nodes / TACOM
$200/TACOM
COST $28909/month

FIGURE 15 : TYPICAL COM DESIGN EXPERIMENT # 1; 100 NODES

4 nodes/line
20 nodes/TACOM
$1500/TACOM
Cost: $25888/month

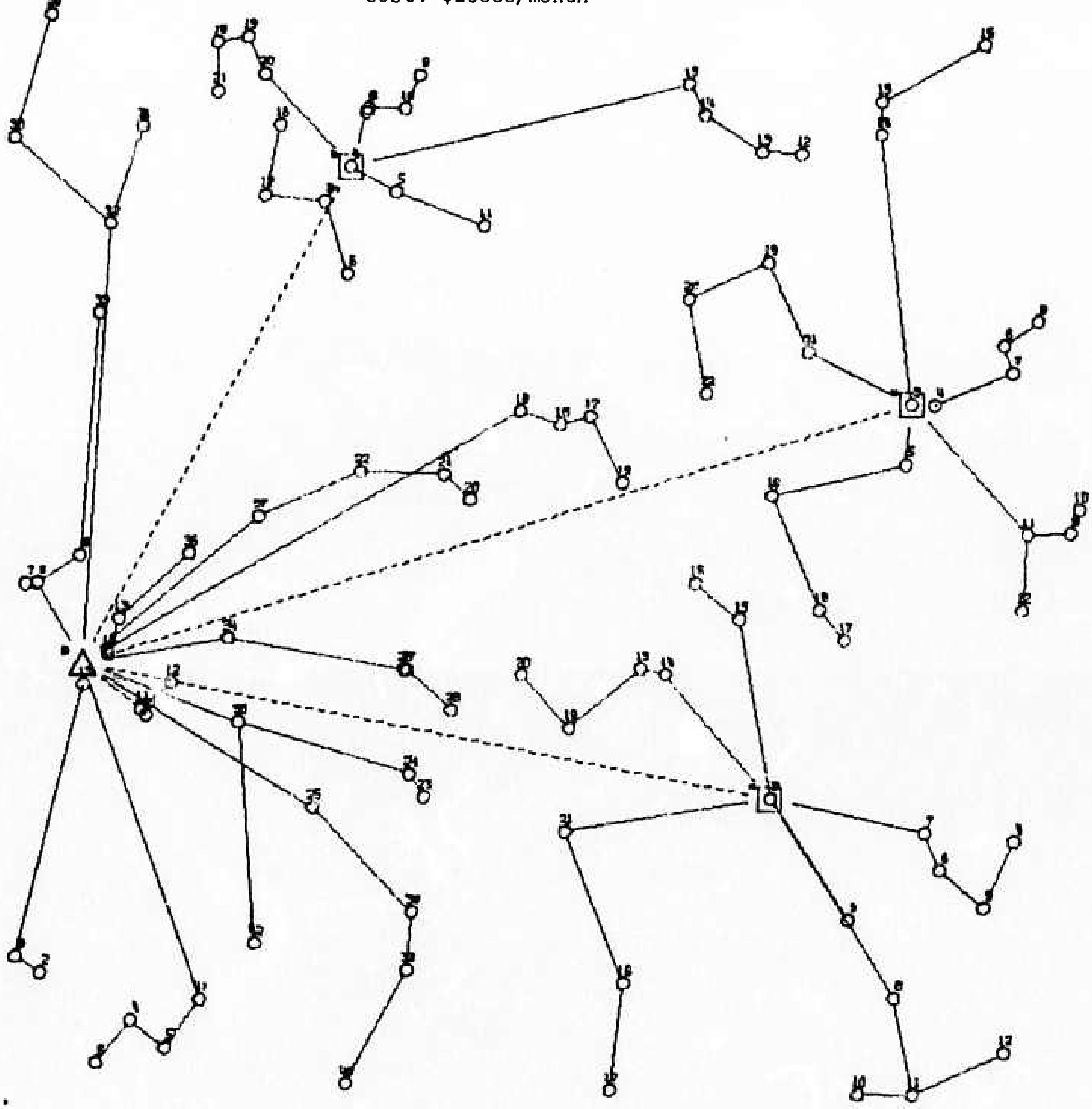
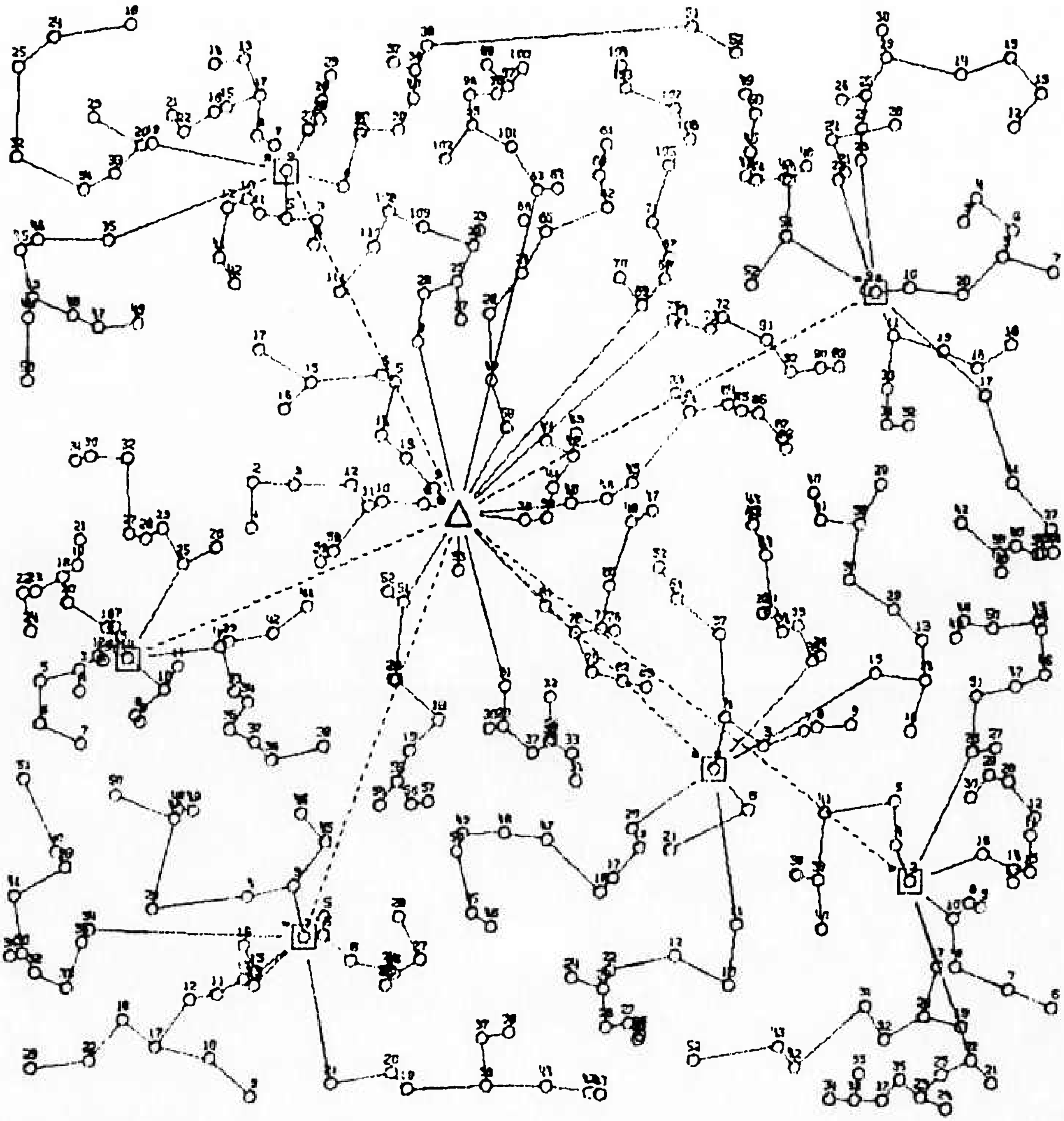


FIGURE 16 : TYPICAL COM DESIGN EXPERIMENT #2; 100 NODES



10 Node/line
50 Nodes/TACOM
$1500/TACOM
Design Cost $55406/month

FIGURE 17 : TYPICAL COM DESIGN
EXPERIMENT #3; 400 NODES

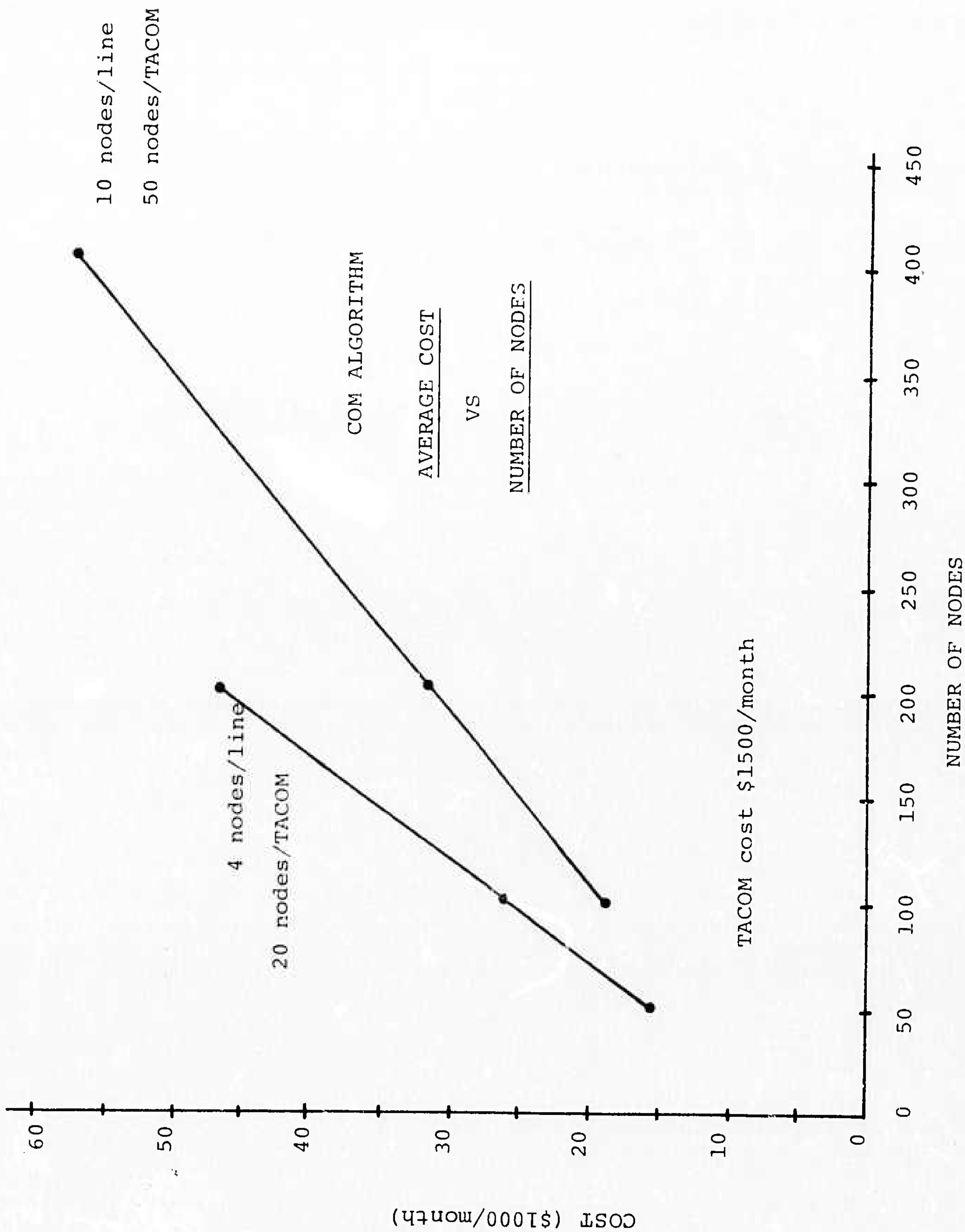
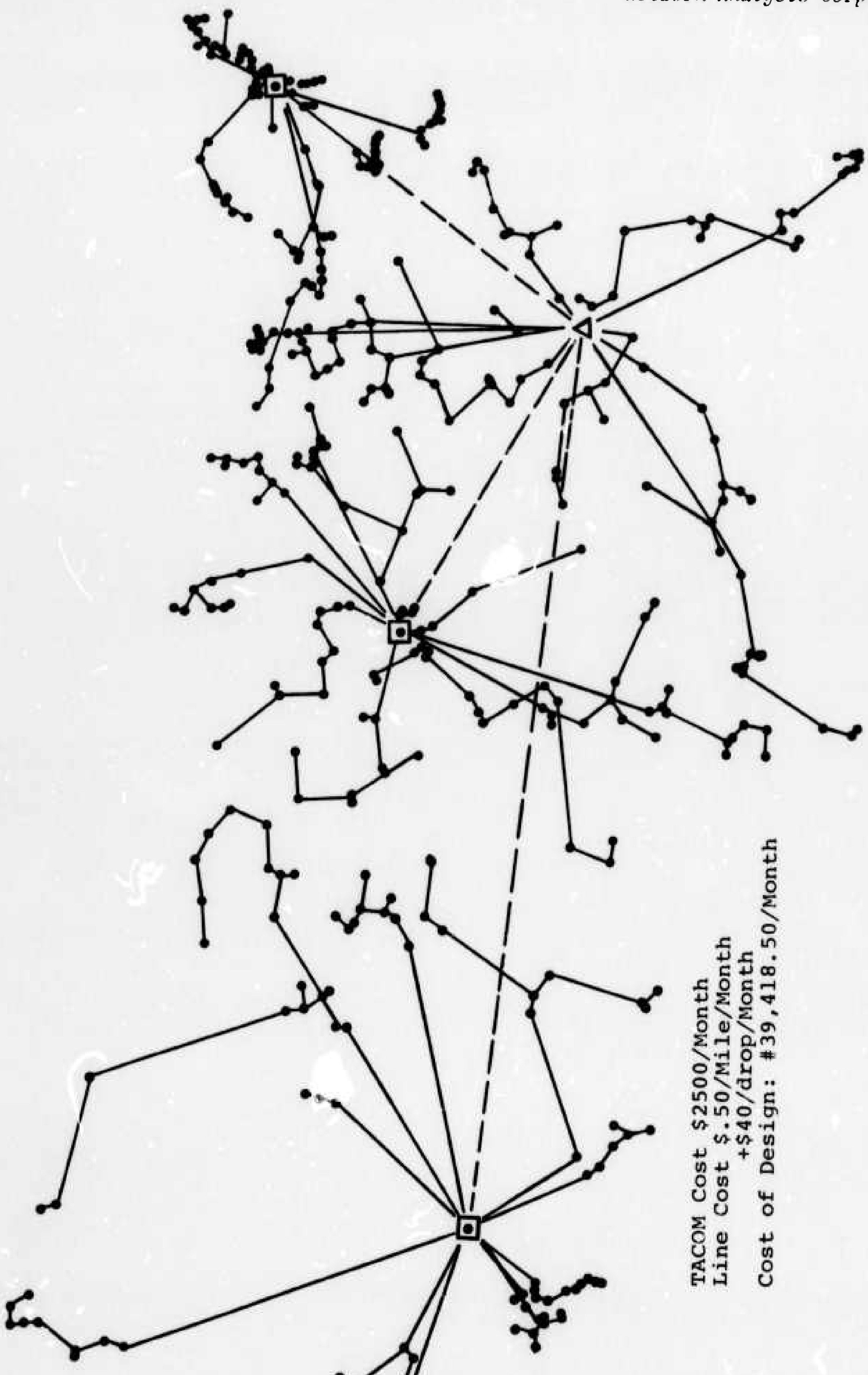
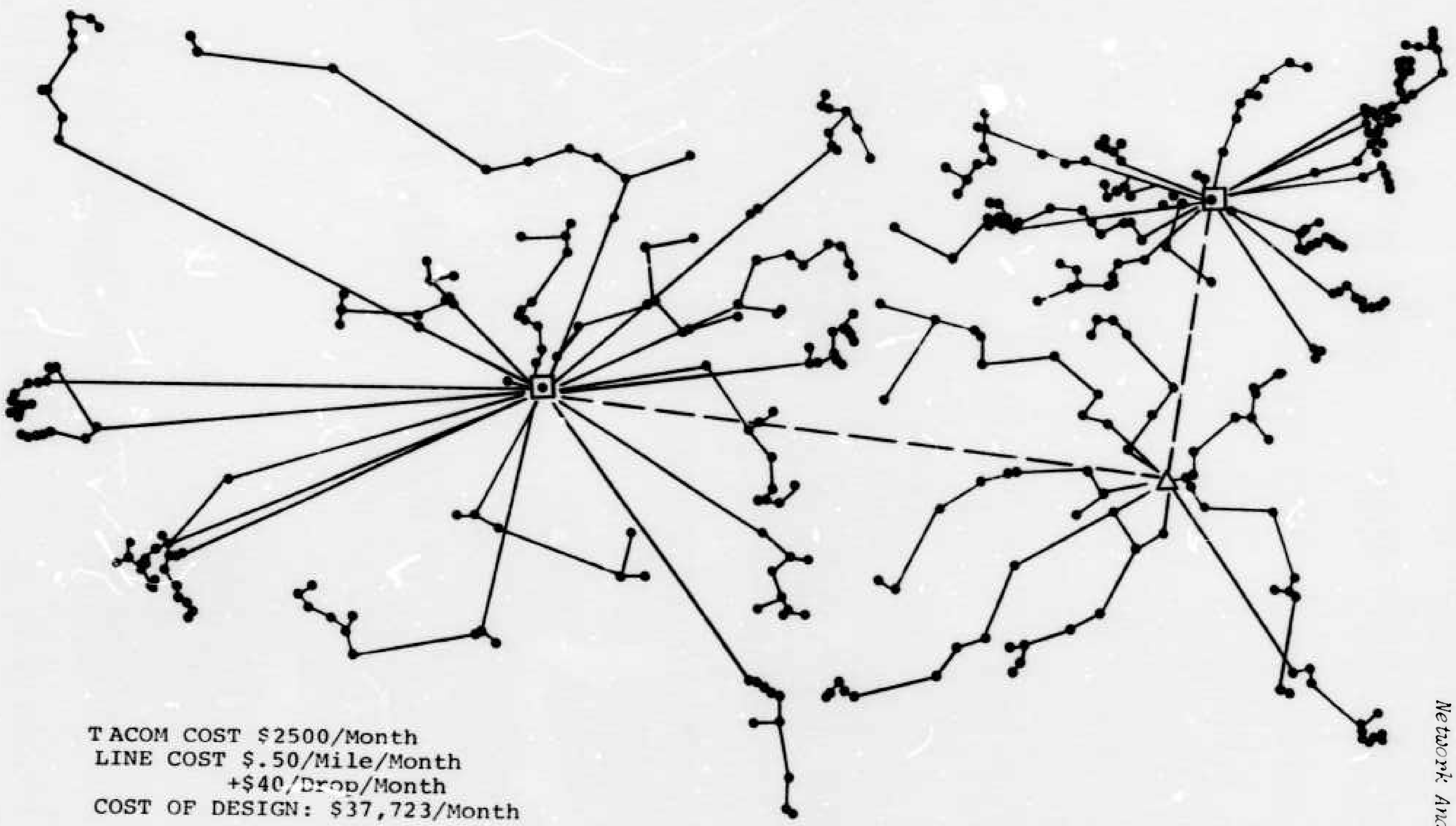


FIGURE 18 : COM ALGORITHM COST PERFORMANCE



TACOM Cost $2500/Month
Line Cost $.50/Mile/Month
+$40/drop/Month
Cost of Design: #39,418.50/Month

FIGURE 19: COM ALGORITHM APPLIED TO 400 NODES DISTRIBUTED IN A RANDOM MANNER BASED ON POPULATION



TACOM COST $2500/Month
LINE COST $.50/Mile/Month
+$40/Drop/Month
COST OF DESIGN: $37,723/Month

FIGURE 20: COM ALGORITHM APPLIED TO 400 NODES DISTRIBUTED IN A RANDOM MANNER BASED ON POPULATION, AND WITH NO CONSTRAINT ON TACOM CAPACITY.

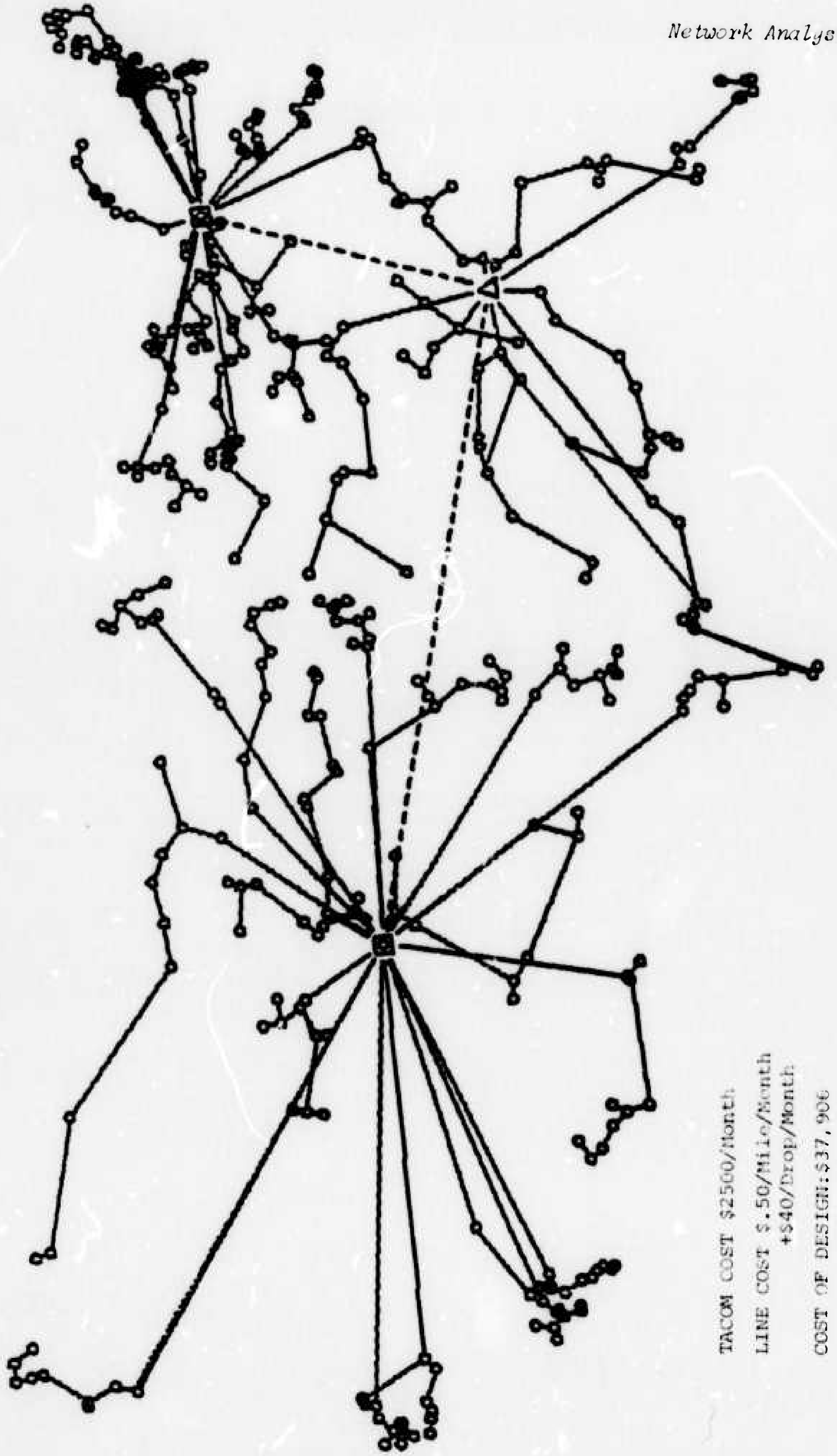


FIGURE 21: ATD ALGORITHM APPLIED TO 400 NODES DISTRIBUTED IN A RANDOM MANNER BASED ON POPULATION, AND WITH NO CONSTRAINT ON TACOM CAPACITY.

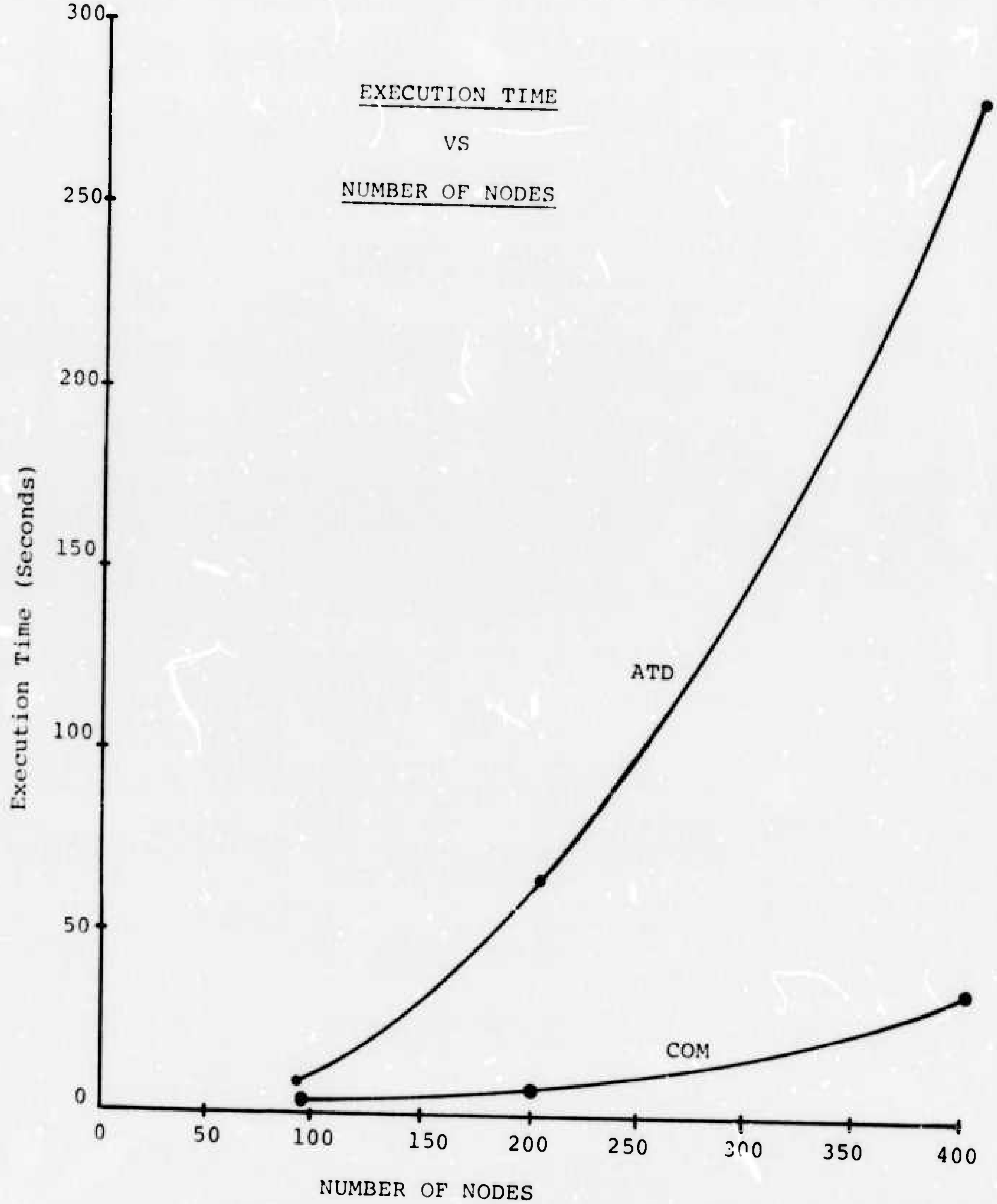


FIGURE 22: EXECUTION TIME of COM and ATD ALGORITHMS

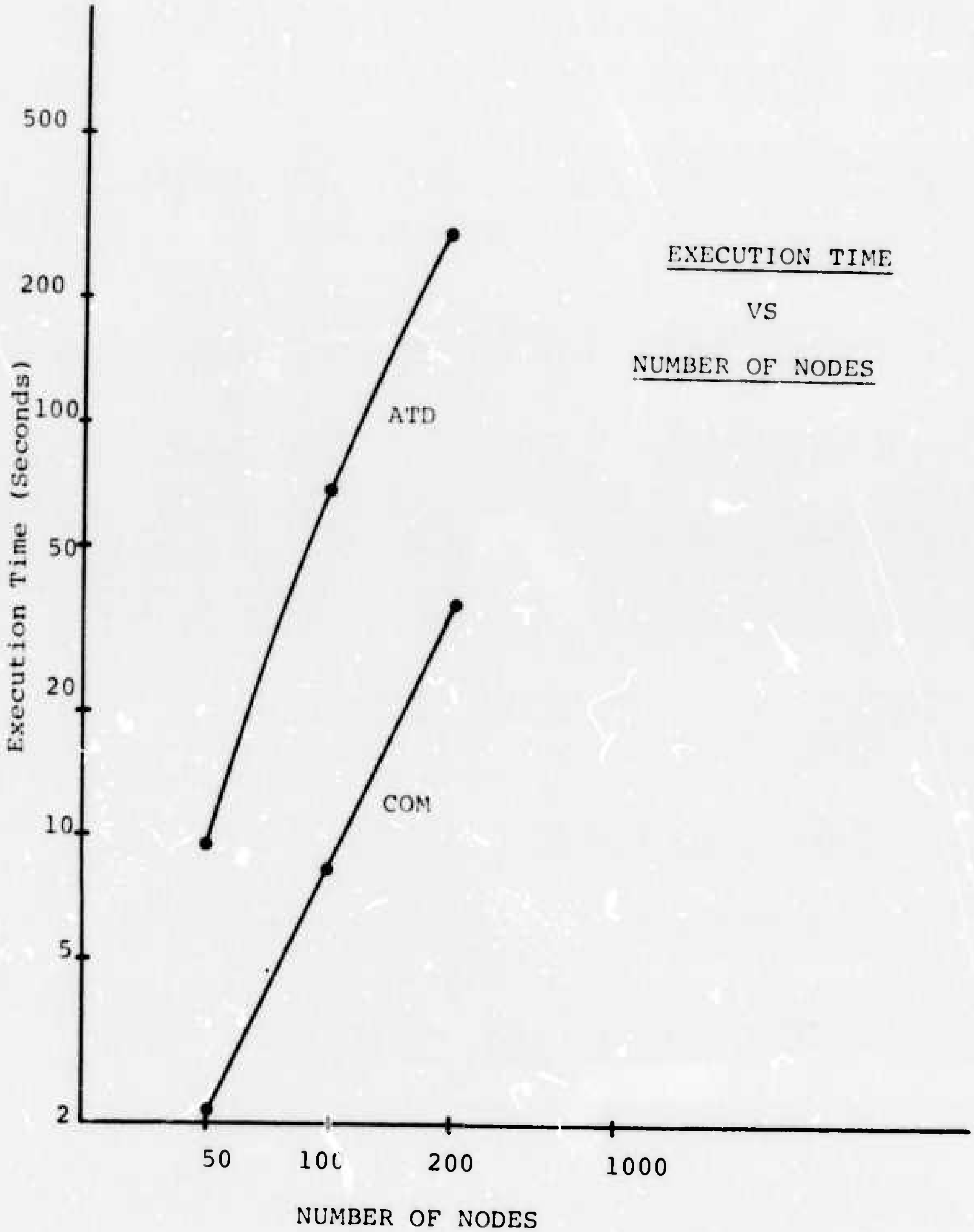


FIGURE 23: EXECUTION TIMES of the COM and ATD ALGORITHMS (LOG-LOG SCALES)

EXPERIMENT #1

4 nodes/line
20 nodes/TACOM
$200/TACOM

ALGORITHM	RESCOP	NUMBER OF NODES (COST IN DOLLARS)		
		50	100	200
COM	1	12754	20348	34353
ATD	1	13409	21332	34443
COM	2	13373	20809	37411
ATD	2	13215	22745	38011
COM	3	12885	21371	36845
ATD	3	13503	22000	37556
Average % Improvement		2.8%	5.3%	1.24%

TABLE 1: PERFORMANCE OF COM ALGORITHM EXPERIMENT # 1.

EXPERIMENT #2

4 nodes/line
20 nodes/TACOM
$1500/TACOM

ALGORITHM	RESCOP	NUMBER OF NODES (COST IN DOLLARS) 50	100	200
COM	1	14005	25846	44275
ATD	1	15385	25955	45794
COM	2	16136	25888	48005
ATD	2	15757	27366	48160
COM	3	15485	26601	47337
ATD	3	17403	27205	47821
Average % Improvement		5.9%	3.9%	1.55%

TABLE 2' PERFORMANCE OF COM ALGORITHM EXPERIMENT # 2.

EXPERIMENT #3

10 nodes/line
50 nodes/TACOM
$1500/TACOM

ALGORITHM	RESCOP	NUMBER OF NODES (COST IN DOLLARS)		
		100	200	400
COM	1	18393	32023	55406
ATD	1	20881	33258	57438
COM	2	18765	32107	58520
ATD	2	21419	33816	
COM	3	18706	32184	5696
ATD	3	22148	34046	
Average % Improvement		13.3%	4.7%	3.5%

TABLE 3: PERFORMANCE OF COM ALGORITHM EXPERIMENT # 3.

10 nodes/line
50 nodes/TACOM
TACOM COST: $1500/Month

ALGORITHM	NUMBER OF NODES (EXECUTION TIME-SECONDS)		
	100	200	400
COM	2.1	8.2	36
ATD	9.4	67.7	275

TABLE 4: AVERAGE EXECUTION TIMES FOR ALGORITHMS.

VIII. GENERALIZATIONS AND EXTENSIONS

The TACOM location problem has been posed in its most basic form, and an algorithm, with associated performance results, has been presented which appears to be an effective approach to the problem. Several generalizations and extensions of the problem and algorithm are now considered.

A. Line Constraints

The line constraint used in the basic formulation was simply a limit on the number of nodes which may share a common line. This constraint is quite realistic when the traffic is uniformly distributed among all the nodes. The appropriate maximum number to ensure acceptable performance can usually be determined quite easily by either analytical or simulation techniques. However, when the traffic level for different terminals varies considerably, a simple number constraint may not be appropriate. When based on the average traffic, the constraint may yield cases of unacceptable performance, and when based on the maximum traffic, it will be inefficient. A constraint we have found effective for such problems is a weighted sum of traffic and number of terminals. In this case, let w_i be the average traffic associated with node i, and m be the number of terminals sharing the same line. The constraint has the form:

$$k_1 \sum_i w_i + k_2 m \leq w_{max}$$

The proper values for the constants k_1, k_2, and w_{max} will depend on the performance requirements and traffic measure used, and may be determined by analytical or simulation techniques.

To incorporate such a constraint in the COM algorithm requires only a simple modification; two numbers are kept for each node instead of one. The first number is the combined traffic, and the second, the number of original nodes represented by the node. The feasibility constraint is evaluated on the basis of both these numbers, in accordance with the above formula. The center-of-mass is determined only on the basis of the second number, i.e., the number of original nodes represented by the node. This is the same procedure as used in the initial statement of the algorithm, and reflects the intention for the clustering process to identify natural geographical groupings of nodes.

In general, the nodes may each possess a number of attributes which must be consistent with a set of constraints for feasibility. This can be formulated as each node having an attribute vector, $\underline{v}_1$, and for feasible merging, a constraint function must be satisfied.

$$C\,(\underline{v}_i,\ \underline{v}_j) = \begin{cases} 1 & i \text{ can be merged with } j \\ 0 & i \text{ can not be merged with } j \end{cases}$$

This general formulation can be handled in the algorithm in the same manner as described above.

B. TACOM Constraints

The TACOM constraint used in the initial algorithm statement was a limit on the number of nodes which could be served by a TACOM. This constraint may be interpreted in terms of a traffic performance relationship. Acceptable performance on the part of the TACOM may be directly related to the amount of traffic it must process, and if the traffic is uniformly distributed among the nodes, then this translates into a constraint on the number of nodes a TACOM may serve. However, when traffic is not uniformly distributed, it may be more appropriate to constrain the amount of traffic rather than the number of nodes. In fact a weighted sum of traffic and number of nodes, as in the line constraint case, is often the most realistic, as there may be an overhead dependent on number of nodes, and a processing load dependent on traffic. The algorithm can easily be modified to handle such a constraint. As in the line constraint case, two numbers are kept for each node; the first number is the combined traffic, and the second the number of original nodes represented by the node. During the "Add" phase of the algorithm, the TACOM feasibility constraint is evaluated on the basis of a weighted running sum of both numbers.

The constraint outlined above appears reasonable in form. However, to obtain appropriate constants for this constraint may be considerably more difficult than obtaining such constants for a similar line constraint, as the TACOM's performance may be intimately related to its particular hardware and/or software characteristics. Consequently, such constraints are usually developed with conservative considerations, and are thus not necessarily inviolate; that is, acceptable performance can often be achieved even though the constraint is violated. The impact of this consideration on the algorithm stems from the following observation: the algorithm creates subproblems for the line layout process by partitioning the nodes into subsets, each of which satisfies the TACOM constraint. If the line layout process were applied to the problem as a whole, with the derived TACOM sites, it may achieve a more economical result than if applied to each subset independently, simply because it has a larger domain over which to "optimize." However, unless a line layout procedure is used that is sensitive to the TACOM constraint, this may result in cases where the TACOM constraint may be violated. As noted above, this may be acceptable. For cases where this is acceptable, the algorithm can be easily modified to simply perform the line layout procedure after all TACOM sites have been selected.

In some cases, the TACOM is more restricted by its hardware line-connection limitations than by its traffic capacity. In this case, a constraint on the number of lines that may be connected to a TACOM is appropriate. Such a constraint should be incorporated in both the "Add" phase of the algorithm, by simply limiting the number of clusters which may be connected to the TACOM, and during the line layout phase. This constraint might reasonably be present along with a traffic constraint.

C. Possible TACOM Sites

In the general formulation of the TACOM location problem, the set of nodes, A, and the set of possible TACOM sites, H, may be considered as independently defined. It is quite feasible for situations to occur where the possible TACOM sites are in fact disjoint from the nodes, partially overlap with the nodes, are a proper subset of the nodes, are the same as the nodes, or have the nodes as a proper subset. For simplicity, the COM algorithm was presented in terms of the case where the possible TACOM sites were the same as the nodes, thus having to deal with only one set, A. We will now consider the other cases. In particular, three variations of the algorithm are presented, each of which appears most appropriate for a particular type of possible TACOM locations set.

1. Small Number of Possible TACOM Sites

Consider the case where the number of possible TACOM sites is much smaller than the number of nodes. The possible TACOM sites may be disjoint from the nodes, a proper subset of the nodes, or partially overlap the nodes (i.e. some possible TACOM sites are nodes, others are not). The portion of the algorithm where nodes are merged into single COM nodes representing clusters should remain the same. However, during the "Add" phase, rather than evaluating the COM nodes as possible TACOM sites, the actual possible sites (members of H) should be evaluated. In this way the "Add" phase will involve a reduced number of nodes, and will directly result in an actual TACOM site selection, thereby, eliminating the requirement for the local optimization phase. Thus, this variation capitalizes on the small number of possible TACOM sites to improve computational efficiency.

2. Large Number of Possible TACOM Sites

It is feasible for the number of possible TACOM sites to be comparable, or even much larger than the number of nodes. For example, if a major oil company wanted to extend its corporate domain into the commercial time-sharing world, the initial number of its customers may be far less than the number of its service stations that may be considered as possible TACOM sites. In the general form of this case, the possible TACOM sites may be disjoint from the nodes, contain the nodes as a subset, or partially overlap the nodes. The handling of this case is particularly simple. The only portion of the COM algorithm to be varied is the local optimization phase. There, rather than selecting an actual TACOM site from the k nodes in A closest to the COM site selected in the "Add" phase, the selection is made from the k nodes in H (the set of possible TACOM sites) closest to the COM site selected in the "Add" phases.

3. <u>Possible TACOM Sites Subset of Nodes</u>

In the most common practical problems, the possible TACOM sites are limited to a subset of the nodes selected on considerations of maintenance, rental space, access by trained company personnel, security, etc. If the subset is sufficiently small, this case should be handled as in section one above. Otherwise, minor modifications of the COM algorithm are appropriate. First, in the merge phase, COM nodes which represent at least one real node that is also a possible TACOM site should be flagged. Then, during the "Add" phase, only the flagged COM nodes are considered as possible TACOM sites. Finally, as in section two above, in the local optimization phase the k nodes closest to the COM node selected in the "Add" phase should be drawn from the set of possible TACOM sites rather than the set of all nodes. These modifications result, not only in satisfying the design constraint of TACOM sites being selected from the restricted set of nodes, but also improve efficiency by restricting attention during the "Add" phase to a reduced set of COM nodes (i.e. only those representing nodes which are also possible TACOM sites).

D. Multiple Capacity TACOMS

There may be several different models of TACOMS available with different capacities and costs. In this case, the total design problem requires selection of model as well as site and associated nodes. There are several possible approaches to this problem. We mention only one that is particularly simple, yet effective.

During the "Add" phase of the algorithm, each model of TACOM is evaluated at each site. This requires little increase in computational burden if the smallest capacity model is evaluated first, as its stopping point can then be used as the starting point for the next larger capacity model. The best performance result is then selected, including model. The process continues as usual until no further savings can be found.

E. Staging

Staging refers to the interconnection of TACOMS as shown in Figure 24. A smaller capacity TACOM is connected to a larger capacity TACOM which is then connected to the RESCOP. There are many possible approaches to this problem. Perhaps the simplest is to locate the large TACOMS first in the usual manner, and then to consider these RESCOPS for the problem of locating the small TACOMS.

F. RESCOP Variations

The basic problem formulation had a single RESCOP to which any number of nodes and-or TACOMS could be connected. There are many other formulations having different RESCOP characteristics. Several of these are considered below.

1. Constraint on Nodes Connected Directly to RESCOP

A frequent practical problem is the connection of terminals to a large time-sharing computer. When the computer is serving a large user population, the over-head required for direct connection may prohibit any direct connections of terminals to the mainframe. In this case, the problem is formulated in terms of a RESCOP to which no nodes may be connected.

The COM algorithm is easily modified to handle such a formulation. The cost of a TACOM is added to the cost of connecting a node to the RESCOP, and the termination condition for the "Add" phase is changed from "no savings achieved" to "no nodes left."

If the large computer has a front-end processor to connect to the TACOMS, the processor may also serve as a TACOM. In this case, the problem is formulated as a RESCOP with the first TACOM site preselected to be co-located with the RESCOP. The pre-selected TACOM may have the same or different node and/or line constraints as the other TACOMS. This problem is handled as a RESCOP to which no nodes may be connected, but with the first TACOM site preselected.

2. Multiple RESCOPS

In many problems there may be multiple RESCOPS. This may be the case initially, as in the problem of locating TIPS in the ARPANET, or as the result of staging TACOMS, as discussed earlier. To modify the algorithm for this case, all that is needed is to evaluate costs on the basis of connection to the closest RESCOP.

3. RESCOP Location

The COM algorithm can also be used as an approach to the general resource distribution problem. In this problem, the question is where to place RESCOPS to provide the most economical connection of all nodes to a RESCOP. The problem is different from the TACOM location problem in that there is no initial facility against which to trade connection costs.

The approach to this problem is very similar to the case of RESCOPS which permit no direct terminal connections. Each node is assigned an initial RESCOP connection cost, which is simply the cost of a RESCOP. Thus, in the "Add" phase, instead of comparing the cost of connecting to a selected site to the cost of connecting to a RESCOP, the comparison is made to the cost of having a RESCOP at its own site.

4. Connectivity Requirements

When the TACOM is intended to provide access to a packet switching subnet (such as provided by a TIP in the ARPANET), there may be requirements placed on its inter-connection to the subnet. In the ARPANET case, such a requirement for a TIP is at least two-connectivity, and acceptable impact on network performance and reliability. To evaluate these constraints for every site would be too costly. Consequently, an approach is taken of determining a list of the k most economical arrangements during the "Add" phase, and then processing the list to determine the first feasible arrangement. If the TACOM's can be inter-connected (such as TIP's connected to TIP's), then the selected TACOM is considered in place for the next iteration of the "Add" phase.

G. Very Large Networks and Further Time Reductions

The COM algorithm may be viewed as composed of a "Clustering" phase, followed by an "Add" phase, followed by a "line-layout" phase. The results reported above were obtained with a rather straight-forward implementation of the algorithm in which no particular effort was made to minimize execution time or storage requirements. However, when working with very large networks (5,000 - 50,000 nodes), such considerations are imperative. We give below a brief discussion of two basic techniques that can be applied to extend the algorithm for applicability to very large networks.

1. Sparsity

In each phase of the COM algorithm, various interconnections of nodes are considered. In general, such interconnections may be viewed as branches of a graph. When all interconnections are considered possible, and are thus examined, the graph is a complete graph. However, there are many interconnections which may easily be discarded from consideration. This corresponds to limiting consideration to a sparse graph. Thus, in determining closest nearest neighbors, it is quite efficient to place a grid over the network, assign each node to a box, and then for each node only examine nodes in the same box or adjacent boxes to determine the nearest neighbor. The box assignment can be made linear, and the nearest neighbor search can then be considerably reduced in complexity. Very distant nodes are naturally excluded from consideration. Such an approach can also be used in the "Add" and line-layout phases. Note that the sparsity does not imply inconsistency or inaccuracy in the results, i.e., true nearest neighbors will indeed be found. However, even greater efficiency can be obtained by coupling the sparsity with acceptable approximations, i.e., if no feasible neighbors are found in adjacent boxes, simply abandon the search, rather than considering the next ring of boxes.

2. Local Considerations

In very large networks, it is often quite reasonable to make decisions on the basis of local considerations rather than global considerations. Thus, in the "clustering" phase, it would appear quite reasonable to merge nearest neighbors on a box-by-box basis rather than choosing the two closest together nodes over the entire network.

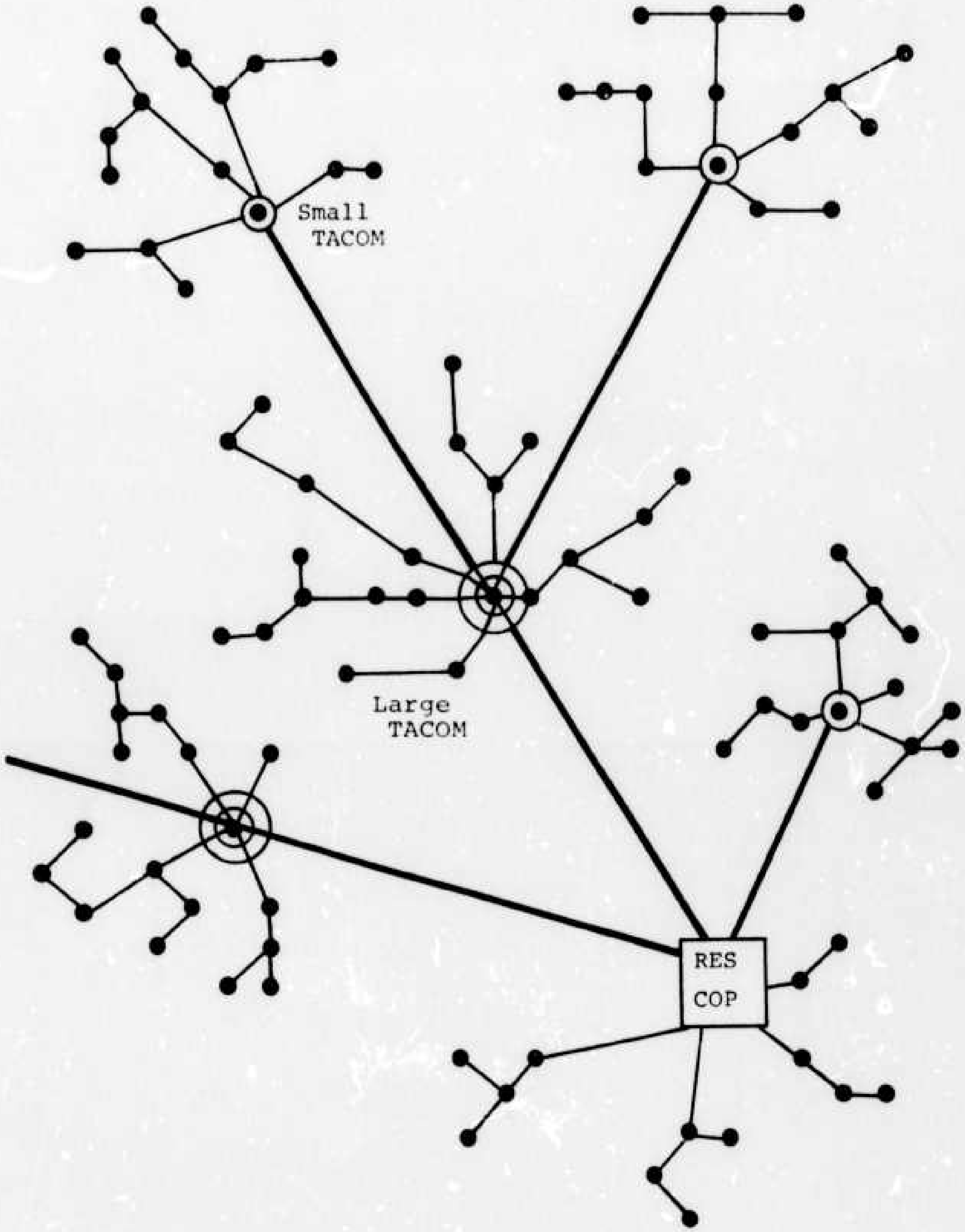


FIGURE 24: TACOM STAGING

IX. CONCLUSION

A new algorithm has been presented for the design of multidrop networks that may incorporate TACOMS (TIP, ANTS, concentrator, or multiplexer) to economically connect nodes (users) to RESCOPS (resource connection points). Experiments with the algorithm indicate that it is both effective and efficient. Extension of the basic algorithm to handle more general problems was shown to be easily accomplished.

Research is continuing on extending the concepts reported here to the integrated design of large (5,000 - 50,000 nodes), hierarchical networks with various levels of access facilities.

REFERENCES

1. Auguston, J. G. and J. Minker, "An Analysis of Some Graph Theoretic Cluster Techniques," Journal of the Association for Computing Machinery, Vol. 17, No. 4. (1970), pp. 571-558.

2. Bahl, L. R. and D. T. Tang, "Optimization of Concentrator Locations in Teleprocessing Networks," Proc. of the Symposium on Computer Communications Networks and Tele-traffic, Polytechnic Institute of Brooklyn, April 4-6, 1972.

3. Bonner, R. E., "On Some Clustering Techniques,", IBM Journal, Jan. 1964, pp. 22-32.

4. Cahit, J. and R. Cahit, "Topological Considerations in the Design of Optimum Teleprocessing Tree Networks," National Telecommunications Conference 73, Nov. 26-28, 1973, Atlanta, Georgia, pp. 37F1-37F7.

5. Chow, W. and A. Kershenbaum, "A Unified Algorithm for Designing Multidrop Teleprocessing Networks," Data Networks: Analysis and Design, Third Data Communications Symposium, St. Petersburg, Florida, Nov. 13-15, 1973, pp. 148-156.

6. Cooper, L., "Location-Allocation Problems," Operations Research, 1962, pp. 331-343.

7. Doll, D. R., "Topology and Transmission Rate Considerations in the Design of Centralized Computer-Communication Networks," IEEE Trans. on Commun. Tech., June 1971, pp. 339-344.

8. Efroymson, M. A. and T. L. Ray, "A Branch-Bound Algorithm for Plant Location," Operations Research, May-June, 1966.

REFERENCES (Cont'd.)

9. Elzinga, J. and D. W. Hearn, "Geometrical Solutions for Some Minimax Location Problems," Operations Research, Oct. 1971, pp. 379-394.

10. Feldman, E., F. A. Lehner, and T. L. Ray, "Warehouse Location Under Continuous Economies of Scale," Management Sci., Vol. 12, May 1966, pp. 670-684.

11. Frank, H. and W. Chou, "Topological Optimization of Computer Networks," Proceedings of the IEEE, Vol. 60, No. 11, Nov. 1972, pp. 1385-1397.

12. Garfinkel, R. S. and G. L. Nemhauser, "The Set-Partitioning Problem: Set Covering with Equality Constraints," Operations Research, Oct. 1968, pp. 848-856.

13. Goldman, A. J., "Optimal Center Location in Simple Networks," Operations Research, Aug. 1970, pp. 212-221.

14. Goldman, A. J., "Minimax Location of a Facility in A Network," Operations Research, Nov. 1971, pp. 407-418.

15. Gower, J. C. and G. J. S. Ross, "Minimum Spanning Trees and Single Link Cluster Analysis," Applied Statistics, Vol. 18, 1969, pp. 54-64.

16. Greenberg, D. A., "A New Approach for the Optimal Placement of Concentrators in a Remote Terminal Communications Network," National Telecommunications Conference, Nov. 26-28, 1973, Atlanta, Georgia, pp. 37D-1 - 37D-7.

REFERENCES (Cont'd.)

17. Jarvis, R. A. and E. A. Patrick, "Clustering Using a Similarity Measure Based on Shared Near Neighbors," IEEE Trans. on Computers, Vol. C-22, No. 11, Nov. 1973, pp. 1025-1034.

18. Jensen, P. A., "Optimum Network Partitioning," Operations Research, July-Aug. 1971, pp. 916-931.

19. Kerninghan, B. W. and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," Bell System Technical Journal, Feb. 1970, pp. 291-307.

20. Kerninghan, B. W., "Optimal Sequential Partitions of Graphs," Journal of the ACM, Vol. 18, No. 1, Jan. 1971, pp. 34-40.

21. Kuehn, A. A. and M. J. Hamburger, "A Heuristic Program for Locating Warehouses," Management Science, Vol. 9, July 1963, pp. 643-666.

22. Ling, R. F., "A Computer Generated Aid for Cluster Analysis," Comm. of the ACM, June 1973, Vol. 16, No. 6, pp. 355-365.

23. Patrick, E. A. and F. P. Fisher, II, "Cluster Mapping with Experimental Computer Graphics," IEEE Trans. on Computers, Vol. C-18, No. 11, Nov. 1969, pp. 987-991.

24. Roach, C., "An Optimization Algorithm for Cluster Analysis," Rand Corporation p. 4878, Aug. 1972.

25. Tang, D. T., "Network Optimization for Teleprocessing Systems," Proceedings of the Fifth Annual Southeastern Symposium on System Theory, North Carolina State University, Raleigh, North Carolina, Duke University, Durham, North Carolina, March 22-23, 1973.

REFERENCES (Cont'd.)

26. Teitz, M. B. and P. Bart, "Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph," Operations Research, July, 1967, pp. 955-961.

27. Woo, L. S. and D. T. Tang, "Optimization of Teleprocessing Networks with Concentrators," National Telecommunications Conference.

28. Zahn, C. T., "Graph Theoretical Methods for Detecting and Describing Gestalt Clusters," IEEE Trans. on Computers, Vol. C-20, No. 1, January 1971, pp. 68-86.

A BRANCH AND BOUND APPROACH TO TOPOLOGICAL NETWORK DESIGN

PART 1

I. INTRODUCTION

We are here concerned with the following topological design problem for a store-and-forward communications network. We wish to design a minimum cost 2-connected network that will accommodate a given amount of inter-node traffic, while keeping the total average delay that packets experience in route below a certain specified amount. It is assumed that all circuits have the same capacity $\bar{C}$. The precise mathematical formulation of the design problem is now presented.

Given: Requirement Matrix R

Cost-capacity functions $D_i = d_i(\bar{C}), \forall i$

Minimize: $D(A) = \sum_{i \varepsilon A} d_i(\bar{C})$ where A is the set of links which correspond to a given topology.

Over A, $\underset{\sim}{f}$

s.t.: (a) $\underset{\sim}{f}$ is a multicommodity flow satisfying the requirement matrix R.

(b) $\underset{\sim}{f} \leq \underset{\sim}{\bar{C}}$

(c) $T = \frac{1}{\gamma} \sum_{i \varepsilon A} f_i \left[\frac{1}{\bar{C} - f_i} \right] \leq T_{max}$

(d) The set A must correspond to a 2-connected topology.

At present, only suboptimal techniques are available for the solution of the design problem. One such technique is the Branch-Exchange (BXC) method, which consists of improving a given initial topology by means of a sequence of topological transformations, called branch exchanges [1]. BXC terminates when all possible transformations are explored.

A refinement of BXC is the Cut-Saturation (CS) method [2], which also uses an iterative approach and performs at each step the topological transformation that is most likely to yield cost-performance improvement.

A third technique is the Concave Branch Elimination method [3], which starts from a fully connected topology and eliminates uneconomical links, until a locally optimal configuration is achieved.

It should be emphasized that the solution obtained by employing the techniques described above, is conceivably a suboptimal solution to the originally posed problem. We were able to obtain lower bounds on the optimal solution, and to show that in most cases, the suboptimal cost is no more than 10 - 20% higher than the lower bound. But there is an indication that the suboptimal solutions are indeed much closer to optimum, say by less than 5%.

It is important therefore that we endeavor to develop a procedure that would yield the exact solution, in order to be able to evaluate the efficiency of the suboptimal techniques, and to determine whether it would pay to improve them. If the exact solution proves to be elusive, we at least hope to determine tighter lower bounds on the total cost of an optimally designed network. The lower bounds can be employed along with the upper bounds corresponding to current heuristic solutions, to effectively trap the optimal solution and reinforce the conviction that our heuristic solution is in fact a very good one.

In the sequel we describe a branch and bound algorithm for the exact solution of the topological problem. The algorithm is particularly attractive because, in addition to providing at the end the optimal solution, it provides also a lower bound which is increasing (and therefore becoming more precise) at each step of the computation.

The purpose of this section is to describe the algorithm, show its convergence to the optimum solution, and discuss its possible applications. The coding of the algorithm is now in progress; experimental results will be available in the near future. Such results will allow evaluation of the computational efficiency of both exact method and heuristic techniques.

II. THE LOWER BOUND

We propose to employ a modified branch and bound technique to effectively construct an optimal topological configuration by an exhaustive display of a proper subset of the set of all possible topological configurations.

A very important ingredient of every branch and bound algorithm is the determination of the lower bound. Lower bounds are generally obtained by solving subproblems which are in a sense more favorable than the original problem, either because some of the constraints are relaxed, or because the real costs are replaced by lower approximations.

In our case, the subproblem consists of determining the minimum cost routing on a partially specified topology. A partially specified topology contains: a set S_A of assigned links, a set S_U of undefined links; and a set S_E of excluded links. The union of the three sets $S = S_A \cup S_U \cup S_E$ is the set of all possible links (which are $\frac{1}{2}NN(NN-1)$ in a network with NN nodes).

The assigned links are links that have been definitively introduced in the network topology and therefore their cost is the real cost corresponding to leasing a circuit of given capacity between the end points. Although the value of capacity on the assigned links is fixed, we can represent the cost of such links as a function of capacity, as shown in Figure 1. The curve in Figure 1 indicates that the only admissible value of capacity for an assigned link is $\bar{C}$, since there is no saving for $C < \bar{C}$, and the cost becomes infinite for $C > \bar{C}$.

The undefined links are links for which it has not yet been decided whether they should be included or excluded from the topology.

Due to this uncertainty, their cost should be somehow proportional to the link utilization, and should be zero if the link carries zero flow. Therefore, the cost of an undefined link is assumed linear with respect to link capacity, and is shown in Figure 2. Notice that the cost of the undefined link equals the real cost for $C = \bar{C}$. Therefore, the undefined link cost is a lower bound to the real cost shown in Figure 1.

The excluded links are links that have been discarded during preceding branch and bound steps, and therefore, are no longer considered at this stage.

For the above mentioned partially specified topology, we are interested in solving the following subproblem:

$$\left[\begin{array}{l} \min\limits_{\underset{\sim}{f},\underset{\sim}{C}} \sum\limits_{i \in S_U} d_i(C_i) \\ \text{Such That} \begin{cases} T_A(\underset{\sim}{f}) \leq T_{MAX} \\ T_U(\underset{\sim}{f}) \leq T_{MAX} \\ \underset{\sim}{f} \text{ is a multicommodity flow corresponding to the requirement matrix } R. \end{cases} \end{array}\right. \qquad (1)$$

where:

(1) $d_i(C_i)$ is the linear cost function for the undefined link i.

(2) $T_A(\underset{\sim}{f}) = \frac{1}{\gamma} \sum\limits_{i \in S_A} \frac{f_i}{\bar{C}-f_i}$ is the total delay on the assigned links.

(3) $T_U(\underset{\sim}{f})$ is the total delay on the undefined links.

(4) $T_A + T_U$ is total delay for the partially specified network.

Let us consider the total cost of the partially specified topology D_P given by:

$$D_P = \sum_{k \varepsilon S_A} D_k + \sum_{i \varepsilon S_u} d_i(C_i) \quad (2)$$

It is clear that at optimality (i.e., after solving problem (1)), D_P represents a <u>lower</u> <u>bound</u> to the cost of <u>any</u> final topology derived from the partially specified one by <u>assigning</u> a subset of undefined links. In fact, in subproblem (1), the cost of each undefined link is a lower bound to the real cost of the link in the final topology. Furthermore, the delay constraint has been relaxed since the total delay is required to be $\leq 2T_{MAX}$ (instead of $\leq T_{MAX}$).

In order to solve problem (1), we first express the cost of the undefined links D_U as a function of the link flows. From [4], we have:

$$D_U = \sum_{i \varepsilon S_U} d_i(C_i) = \sum_{i \varepsilon S_U} d_i f_i + \frac{1}{\gamma T_{MAX}} \sum_{i,j \varepsilon S_U} \sqrt{d_i d_j} \sqrt{f_i f_j} \quad (3)$$

This expression relates D_U to the link flows in such a way that the delay constraint $T_U \leq T_{MAX}$ is satisfied.

Since the expression in (3) is concave in $\underset{\sim}{f}$, and we prefer a convex objective function, we further bound D_U as follows:

$$D_U \geq \sum_{i \varepsilon S_U} d_i f_i + \frac{1}{\gamma T_{MA}} \sum_{i,j \varepsilon S_U} \frac{\sqrt{d_i d_j}}{\bar{C}} f_i f_j \quad (4)$$

It can be easily shown that the r.h.s. of (4) is convex. By replacing (4) in the original formulation of the subproblem in (1), we obtain a convex multicommodity flow problem whose solution still provides a lower bound on the final cost.

Subproblem (1) can now be rewritten as follows:

$$\min_{\underset{\sim}{f}} \sum_{i \epsilon S_U} d_i f_i + \frac{1}{\gamma T_{MAX}} \sum_{i,j \epsilon S_U} \frac{\sqrt{d_i d_j}}{\bar{C}} f_i f_j \tag{5}$$

$$\text{s.t.} \begin{cases} T_A(f) = \frac{1}{\gamma} \sum_{R \epsilon S_A} \frac{f_i}{\bar{C} - f_i} \leq T_{MAX} \\ \underset{\sim}{f} \text{ is a multicommodity flow satisfying requirement matrix R.} \end{cases}$$

Since the objective function in (5) is convex and the constraint set is also convex, there exists only one local minimum, which is also the global minimum.

The solution of (5) can be carried out using the method of the Lagrange multiplexers. To do so, we rewrite (5) as follows:

$$\min_{\underset{\sim}{f}} L(\underset{\sim}{f}, \lambda) = \sum_{i \epsilon S_U} d_i f_i + \frac{1}{\gamma T_{MAX}} \sum_{i,j \epsilon S_U} \frac{\sqrt{d_i d_j}}{\bar{C}} f_i f_j$$

$$+ \lambda \left\{ \frac{1}{\gamma} \sum_{R \epsilon S_A} \frac{f_i}{\bar{C} - f_i} - T_{MAX} \right\} \tag{6}$$

s.t. $\underset{\sim}{f}$ is a multicommodity flow satisfying requirement matrix R.

The multiplier $\lambda \geq 0$ must be chosen in such a way that, for the optimal flow $\underset{\sim}{f}^*$, we have:

$$\lambda \left\{ \frac{1}{\gamma} \sum_{R \epsilon S_A} \frac{f_i^*}{\bar{C} - f_i^*} - T_{MAX} \right\} = 0 \tag{7}$$

A possible way to solve (6) is to select an arbitrary $\lambda°$, minimize $L(f,\lambda°)$ over $\underset{\sim}{f}$, then verify if Equation (7) is satisfied. If (7) is not satisfied, a different λ, say λ^1, is chosen and $L(\underset{\sim}{f},\lambda^1)$ is minimized, etc. We anticipate that the search on λ will not be a serious computational bottleneck since $\sigma(\lambda)$ = min (over $\underset{\sim}{f}$) $L(\underset{\sim}{f},\lambda)$ is monotonic. Furthermore, we don't need to solve Equation (6) with great accuracy, but rather can be satisfied with a lower bound to its optimum solution.

At this point, the topological problem for a partially specified topology has been reduced to the solution of an unconstrained, strictly convex, multicommodity flow problem, namely, the minimization of $L(\underset{\sim}{f}, \bar{\lambda})$ over $\underset{\sim}{f}$. Such a minimization is readily carried out with the Flow Deviation Method [5], an efficient tool for the solution of nonlinear flow problems.

We have shown how we can find a lower bound on the cost of all possible topologies deriving from a partially specified topology. The next section shows how we can use these results for the search for the globally optimal topology.

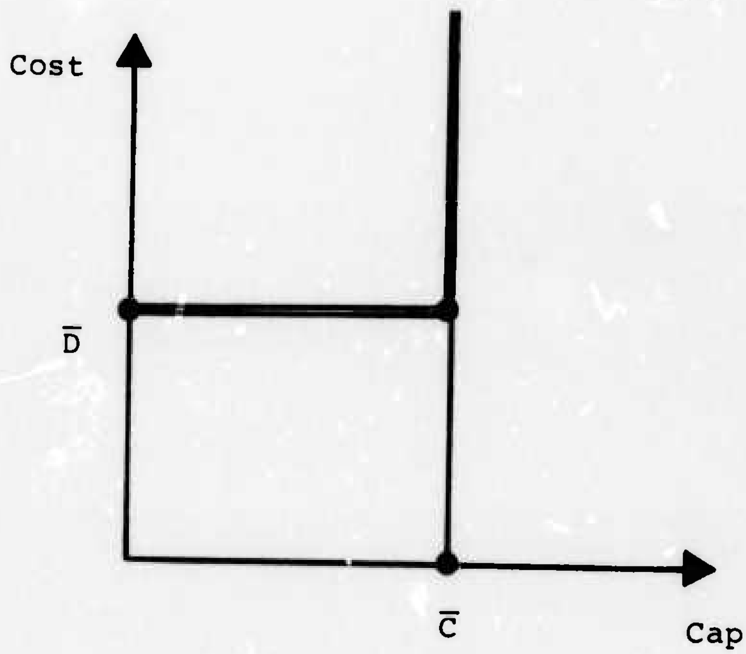


FIGURE 1: COST-CAPACITY FUNCTION FOR AN ASSIGNED LINK

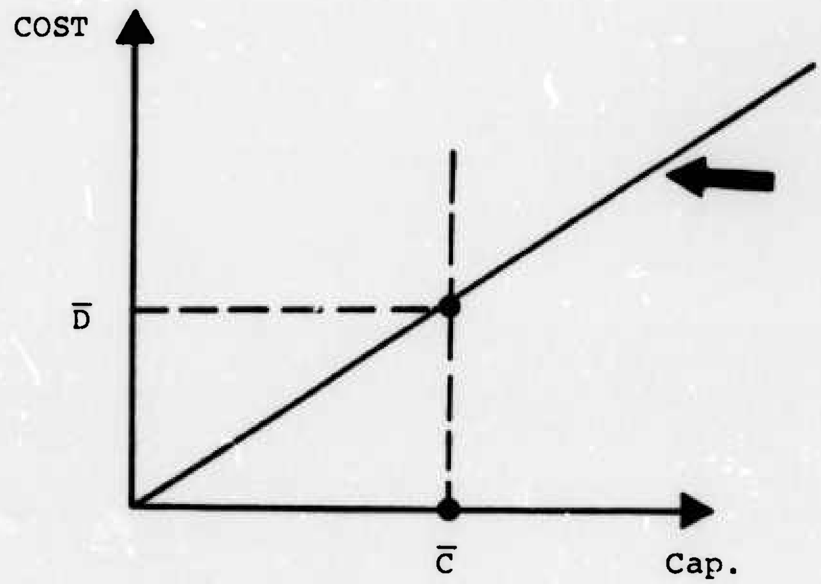


FIGURE 2 : COST VS. CAPACITY FOR UNDEFINED LINK

III. THE BRANCH AND BOUND CONCEPT

First, we describe the branch operation. Suppose we have a partially specified topology T_n to which are associated, three sets of links S_A, S_U, S_E defined above. From such a topology, we can "branch" to two new topologies by removing a link, say link i, (selected with some well defined criterion) from S_U and reassigning it to either S_A or S_E. The two new topologies T_{n+1} and T_{n+2} correspond to the following sets of links:

$$T_{n+1} = (S_A + \{i\},\ S_U - \{i\},\ S_E)$$

$$T_{n+2} = (S_A,\ S_U - \{i\},\ S_E + \{i\})$$

It is clear that if we start from the topology T_o, in which all links are undefined (i.e., $S_A = S_E = 0$), and proceed branching until all the final topologies have $S_U = 0$, we have generated all the $2^{\frac{1}{2} NN(NN-1)}$ graphs that can be constructed on NN nodes.

The branch and bound techniques is essentially an intelligent way of searching the tree of successive derivations and branches, without enumerating all the $2^{\frac{1}{2} NN(NN-1)}$ possible solutions. The key instrument that helps us in this search is the notion of lower bound, discussed in the previous section.

A typical step of a branch and bound procedure is now described. Suppose we have generated, by means of successive branch operations, a part of the tree; (see Figure 3) and suppose the incomplete tree has p "leaves" (i.e., terminal topologies), namely $T_n, T_{n+1}, \ldots, T_{n+p-1}$, with respective lower bounds LB_n, $LB_{n+1}, \ldots, LB_{n+p-1}$, computed by solving the associated subproblems (as discussed in Section 2). The next branch move is performed on topology T_{n+S} such that $LB_{n+S} \le LB_i \ \forall i = n,\ldots,n+p-1$.

To show that the procedure converges to the optimum solution, we first must show that the lower bound for the two topologies generated by a branch operation (successors) is greater or equal to the lower bound for the originating topology (predecessor). But, this is true since the two successor subproblems have more constraints than the predecessor subproblem, so the minimum for the former is generally higher than the minimum for the latter.

The branch and bound procedure terminates when the topology which minimizes the lower bound corresponds to a feasible topology (i.e., is such that the set S_A can satisfy all the requirements with $T \le T_{MAX}$ and corresponds to a 2-connected topology). Such a topology is optimal because its lower bound corresponds to the actual cost, and is lower or equal than the lower bound for any other infeasible topology, and therefore, (because of the nondecreasing property of lower bounds), lower or equal to the cost of any other feasible topology.

To summarize, an iteration of the branch and bound procedure contains the following steps:

A. Determine, between all the topologies so far generated, the topology which minimizes the lower bound. If such a topology is feasible, STOP: optimum has been found.

B. Perform a branch operation on such topology.

C. Compute the lower bounds for the two successors and check for feasibility. Go to A.

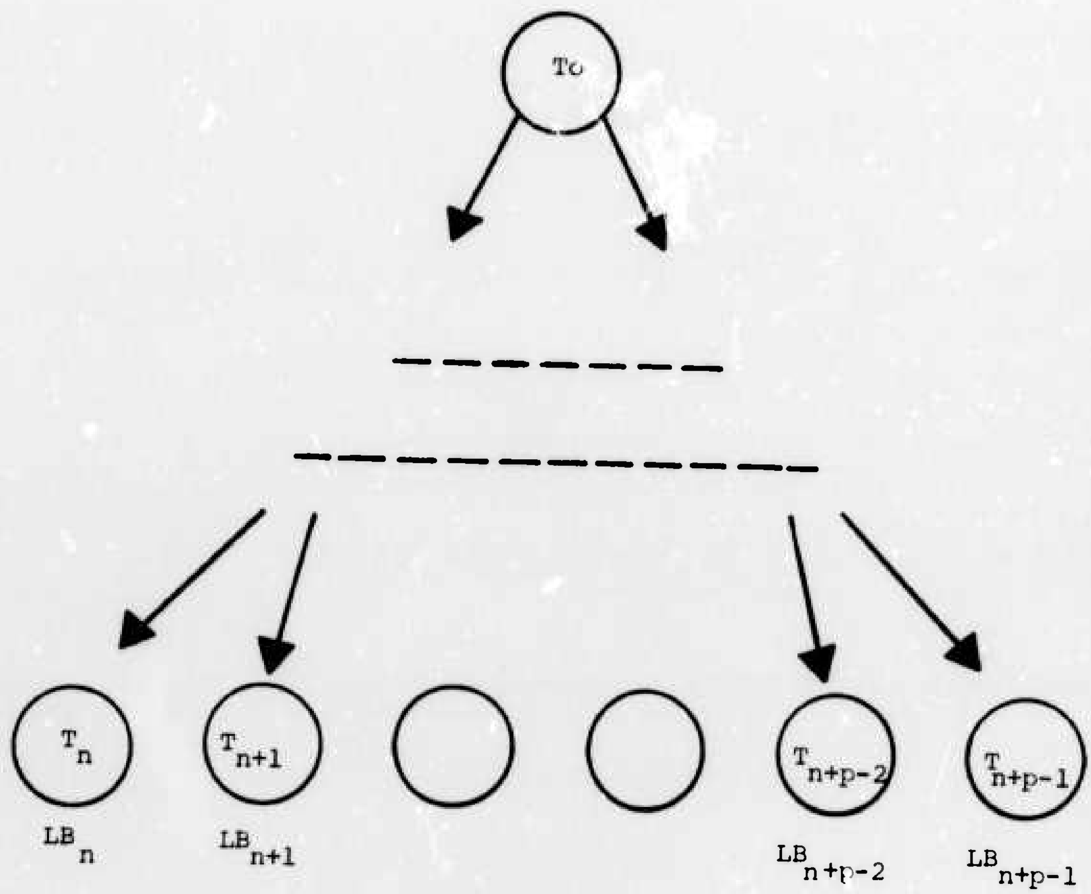


FIGURE 3: BRANCH AND BOUND TREE

IV. COMPUTATIONAL CONSIDERATIONS

The only way to determine the computational efficiency of a search algorithm such as the branch and bound method, is that of performing experiments on a reasonable number of sample problems of various size and complexity. Since experimental results are not available yet, we can only make qualitative statements.

It can be anticipated that the computational effort required to find the optimal solution will be considerably larger (by several orders of magnitude) than the effort required by suboptimal techniques. Therefore, it is very unlikely that branch and bound is a suitable network design tool.

The indication that the method will be very time consuming follows from the conjecture that there is typically a very large number of near-optimal solutions. The method will have to search all such solutions before being able to declare that a particular solution is optimal. Incidentally, it is of interest to note that the presence of a large number of near-optimal solutions is detrimental to the efficiency of exact methods, but is beneficial to the efficiency of heuristics, and of locally optimal techniques in general.

In favor of branch and bound, we have the following facts:

A. It is much less time consuming than enumeration. In fact, although there are a lot of good topologies, there is a much larger number of very bad topologies that will never be explored by branch and bound.

B. There are a variety of features in a branch and bound algorithm that can be properly taylored, in order to obtain maximum efficiency. For example, one must properly design the selection criterion for links to be assigned or excluded, based on cost, utilization, connectivity, etc. Also, the results of the solution of a predecessor subproblem could be used for the solution of successor subproblems, with substantial computational savings. In addition, one might consider other types of search (e.g., the depth first search) instead of the steepest descent search described in the previous section.

C. Although the optimal solution is elusive, it is likely that the bound becomes sufficiently tight much before we find the optimum. In particular, we hope that a reasonable computational effort will reduce the gap between heuristic solution and lower bound from 10-20% (as we have now) to less than 5%.

V. IMPLICATIONS FOR FUTURE RESEARCH

Beside the determination of globally optimal topologies, the branch and bound concept can be used for other applications related to topological design. Two possible applications are here described.

Suppose we want to optimally upgrade an existing network configuration, following an increase of traffic requirements and the installation of new nodes. Let us assume that we are able to idenfity a set of candidate links that most likely include as a subset, the set of links corresponding to the optimum topological reconfiguration. The branch and bound algorithm can be applied setting $S_A \triangleq$ set of original links and $S_U \triangleq$ set of candidate links. If the optimal solution appear to be elusive, we can always use the lower bound information to control a suboptimal technique such as the Cut-Set Method.

Another area of application of the branch and bound concept is within a suboptimal procedure. During the application of the Cut-Saturation Algorithm, for example, [2], we are at each step confronted with the problem of which new links to introduce, or which old links to eliminate from the current topology. Presently, a choice is made, without the ability of evaluating the performance of the remaining alternatives. To avoid this, one might think of solving a partially defined topological problem, with S_U corresponding to the set of links which are candidate for introduction or removal. The nature of the solution (e.g., the relative utilization of links, etc.) might offer valuable insight into the relative cost effectiveness of the links, and provide guidance in the selection of links to introduce or eliminate.

The branch and bound algorithm here described can be further extended to the solution of multiple capacity option problem. However, it is conceivable that the computational complexity will rapidly increase due to the large number of possible combinations.

In summary, we have shown that branch and bound is a valid approach to the exact solution of the topological problem; we have also indicated that it might find applications in the solution of practical problems, when combined with heuristics or used to develop bounds. We have to wait now for the first experimental results in order to better define the area of applicability of this theoretically very attractive technique.

REFERENCES

1. Frank, H., I.T. Frisch, and W. Chou, "Topological Considerations in the Design of the ARPA Computer Network," AFIPS Conference Proceedings, Vol. 36, SJCC, Atlantic City, N. J., 1970, pp. 581-587.

2. Network Analysis Corporation, "The Practical Impact of Computer Advances on the Analysis and Design of Large Scale Networks," Second Semiannual Technical Report, December 1973.

3. Gerla, M., "The Design of Store-and-Forward Networks for Computer Communications," Engineering Report #7319, School of Engineering and Applied Sciences, UCLA, Los Angeles, California, January 1973.

4. Kleinrock, L., Communication Nets: Stochastic Message Flow and Delay, McGraw-Hill, New York, 1964.

5. Fratta, L., M. Gerla, and L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design," Networks, Vol. 3, No. 2, 1973.

A SYSTEM FOR LARGE SCALE NETWORK COMPUTATIONS-PART II

In order to support the many and varied network calculations desired in other chapters of this report, a software system for large scale network computations is being developed. Besides the support function, this system is designed to be portable and general so that it can be used for almost all network based applications in a distributed computational environment such as ARPANET. While the initial applications of the system within NAC have been to communication networks, discussions are in progress with the army to apply parts of the system to applications in risk analysis and circuit analysis. Within NAC the software has been used with:

(1) RELOUT: An interactive program for analyzing the cost throughput, delay, and reliability of packet switched data networks. This program is described in the chapter. Impack Of Interactive Graphics On Network Design.

(2) A simulator of the Packet Radio System: This application is notable because it is a hybrid batch-interactive program running on three computers an IBM 360-91, a PDP-10, and an Imlac PDS-1 Graphic Display Computer. The simulator is described in the chapter: Simulation of Computer Communication Networks.

(3) A heuristic algorithm for Set Covering Problems Which Illustrates another dimension of interactive graphics used for network analysis. This is the use of visual feedback to guide the design of heuristics. The heuristics and their application to radio repeater location problems is described in the chapter: Repeater Location Optimization.

The software in the system consists of three parts:

(a) Input, output, and editing functions,

(b) Data structuring and mapping,

(c) Languages for coding network algorithms.

Virtually all the work has been done in the first area. The first version of the network editor described in the Second Semi-annual Report has been implemented: a simple but effective windowing for the graphics display is 90% implemented: and general purpose parameter selection software has been implemented. It is this software that has been used in the application programs described above.

Exploratory work is being carried out in the second area of data structure and mapping. The immediate goal is to simplify hybrid interactive-batch calculations on the ARPANET. This is being done in conjunction with the dynamic modelling group at project NAC.

Probably the best characterization of the software is to show how it is used. To do this we first reproduce a scenario showing the use of the TTY version (without graphics) of RELROUT. Following this are photographs of some of the graphics displays. Figure 1(a) shows the June 1974 ARPANET. Figures 1(b) and 1(c) show two 2x en-larged details using the windowing capability. Figures 2(a) and 2(b) are graphic output from the reliability and routing programs respectively. In 2(b) throughput-delay curves are superimposed for three different line capacities 19.2, 50, and 230 kilibits per second.

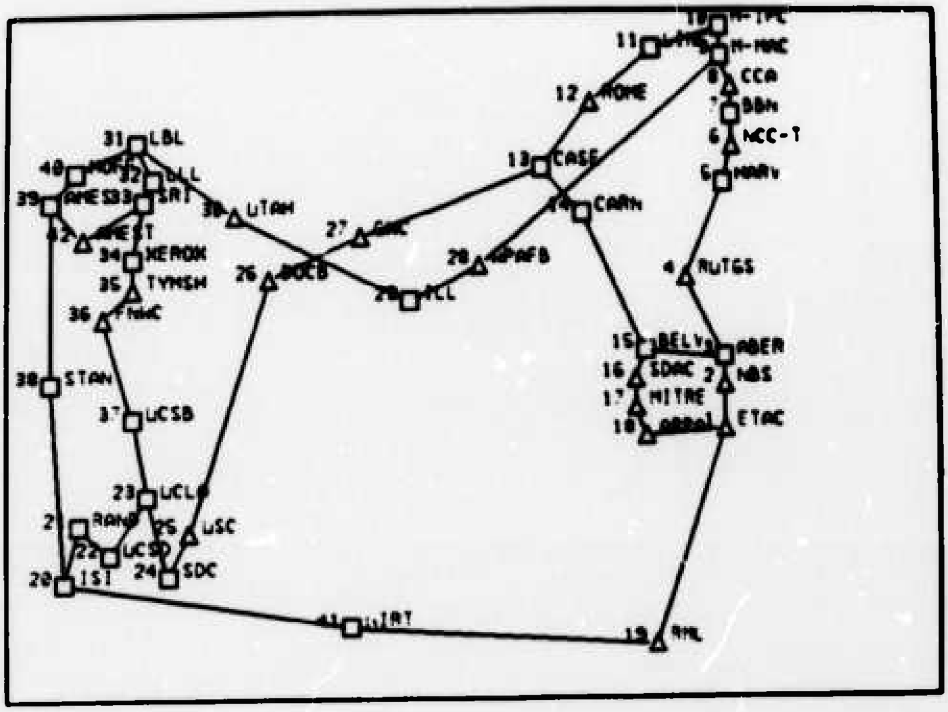

31 LBL
32
SRI
38 UTAH
39
42
34 XEROX
35 TYMSH
36
38 STAN
UCSB
23 UCLA
RAND
25 USC
22 UCSD
24 SDC
20 ISI
CCA
BBN
6 NCC-T
12 ROME
13 CASE
CARN
4 RUTGS
ABER
2 NBS
ETAC
16 SDAC
MITRE
15
11
28
26
27
19

FIGURE 1 (A)

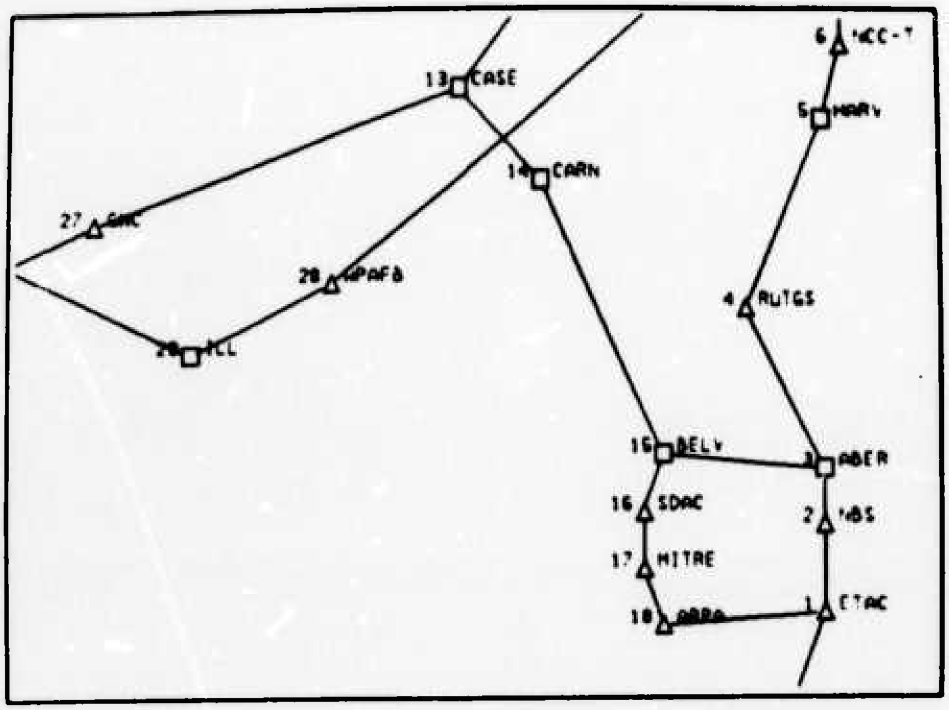


FIGURE 1 (B)

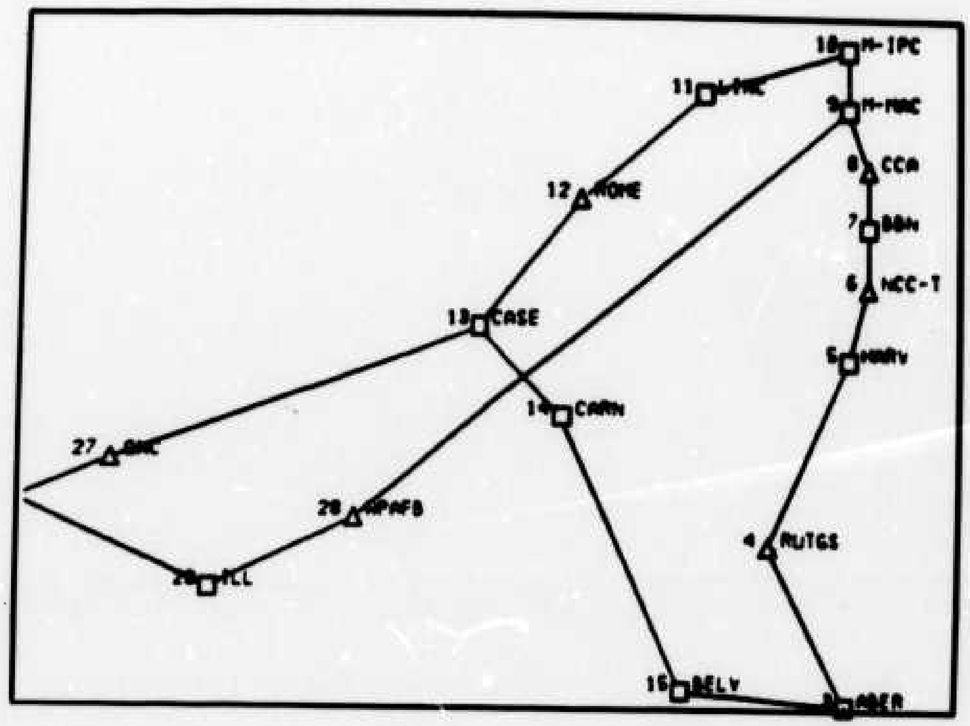

M-IPC
9
M-MAC
8
CCA
7
BBN
6
NCC-T
5
HARV
4
RUTGS
12
13
CASE
14
CARN
15
BELV
ABER
27
28
APAFB
11

FIGURE 1 (C)

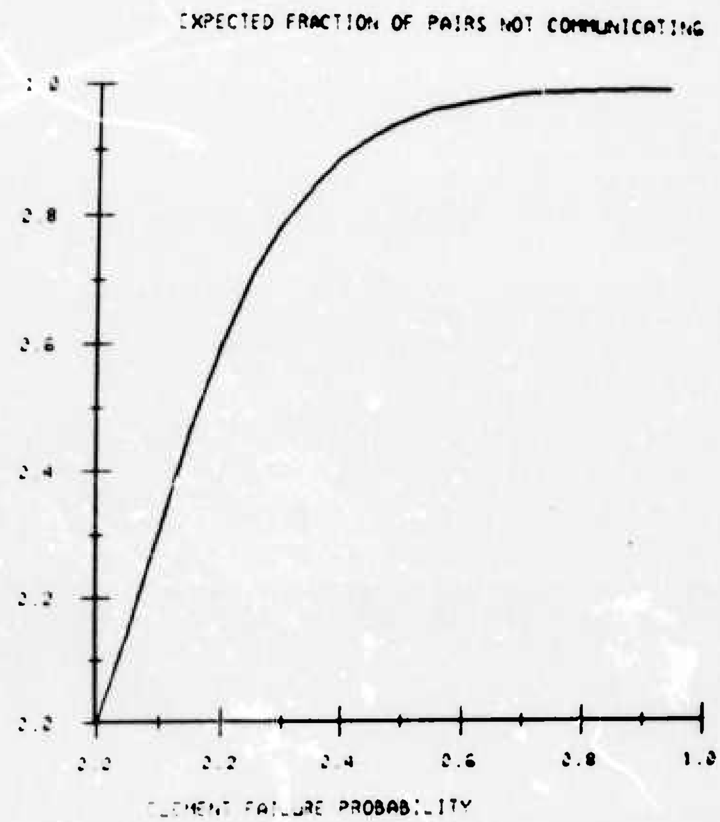
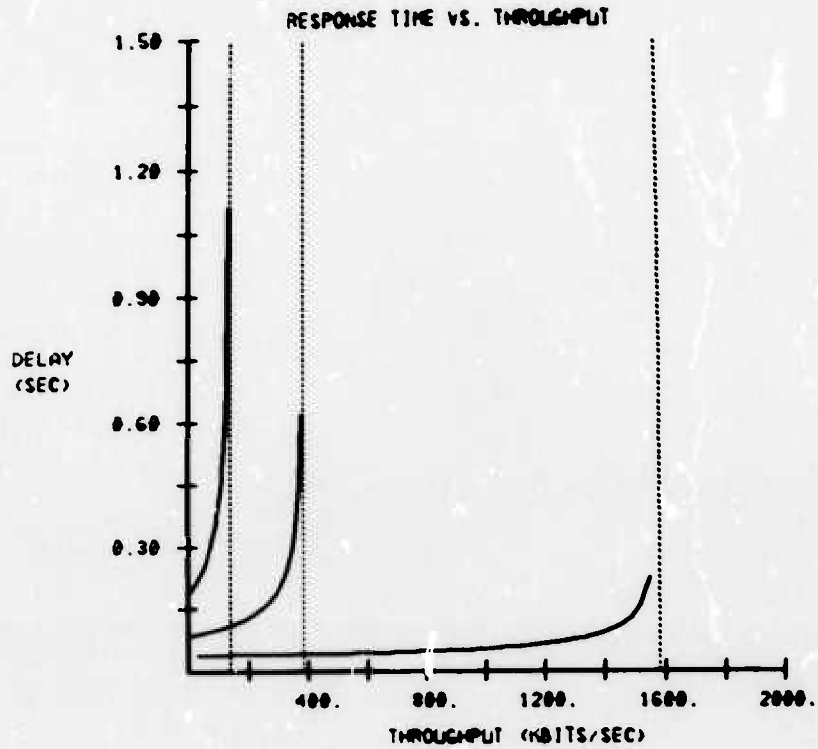


FIGURE 2 (A): RELIABILITY CURVE



NTER OUTPUT COMMAND OR ? <DX_

FIGURE 2 (B): THROUGHPUT - DELAY CURVE

```
TY NSCA ?  ?
STY NSCENARIO.;1

;  <NAC>NSCENARIO.;1   MON 20-MAY-74 9:35AM          PAGE 1

                 ****************************
                 NETWORK ANALYSIS CORPORATION
                 ****************************

                 NETWORK RELIABILITY ANALYZER

                             AND

                    NETWORK ROUTING PROGRAM

                         MAY  15, 1974
                 ****************************

        THIS PROGRAM CREATES A NETWORK DATA BASE WHICH
CAN BE FILED ON DISK FOR FUTURE REFERENCE OR SUBMITTED
TO A NETWORK RELIABILITY ANALYZER OR A NETWORK ROUTING
PROGRAM.


        TERMINATE ALL INPUT WITH A CARRIAGE RETURN.
(IF YOU HAVE A SPECIALLY SUPPLIED IMLAC).
DO YOU WANT IMLAC GRAPHICS? (Y OR N)? <N;IF YOU HAVE IMLAC WITH LVH AND S
**PECIAL IMLAC MONITOR TYPE Y AND GET GRAPHICS
ENTER SYSTEM COMMAND OR ? <?;WE WISH TO SEE OPTIONS NEXT LEVEL DOWN IN C
**OMMAND HIERARCHY

ONE OF THE FOLLOWING:
 E     EDIT TOPOLOGY;    P     TYPE OUT AND SET PROP. VALUES
 I     INPUT NETWORK;    REL   RELIABILITY ANALYSIS
 F     FILE NETWORK ;    ROUT  ROUTING ANALYSIS
 Q     QUIT PROGRAM ;    D     DISPLAY NETWORK
ENTER SYSTEM COMMAND OR ? <I;WE NEED A NETWORK
ENTER INPUT FILE NAME OR ? <?;WE SEE WHAT IS AVAILABLE
 ONE OF THE FOLLOWING:
 ARPA       FOR ARPANET DEMO NET
 <ESC>      FOR DEFAULT NETWORK
 <CR><CR>   TO NOT FILE NETWORK
 NIL:       FOR A NULL NETWORK
 5 CHAR. FILE NAME FOR -----.DAT FILE
ENTER INPUT FILE NAME OR ? <ARPA;WE USE ARPANET DEMO
READ NETWORK FROM ARPA  .DAT ? (Y OR N)? <Y

FILE: ARPA .DAT FOUND...OPENED...READING NETWORK...READY.
NETWORK OF   42 NODES AND   47 LINKS NOW IN MEMORY
ENTER SYSTEM COMMAND OR ? <P;IF WE HAD AN IMLAC WE COULD DRAW NET
ENTER PROPERTY COMMAND OR ? <?
```

```
    ONE OF THE FOLLOWING:
TAN (TAL)   TYPE ALL NODES(LINKS)
TN  (TL )   TYPE SPECIFIC NODES(LINKS)
SAN (SAL)   SET SAME VALUES TO ALL NODES(LINKS)
SN  (SL )   SET VALUES TO SPECIFIC NODES(LINKS)
FSN (FSL)   FORMATTED SET OF PROPERTY VALUES
ENTER COMMAND OR <CR> <TAN>WE LOOK AT NODE PROPERTIES OF ARPANET
```

NODE		FP	LONG	LAT
1	ETAC	1.0000000	77.000	38.830
2	NBS	1.0000000	77.160	39.130
3	ABER	1.0000000	77.000	39.000
4	RUTG	1.0000000	74.450	40.480
5	HARV	1.0000000	71.250	42.500
6	NCC	1.0000000	71.250	42.500
7	BBN	1.0000000	71.250	42.500
8	CCA	1.0000000	71.330	42.500
9	MIT2	1.0000000	71.200	42.500
10	MITI	1.0000000	71.200	42.500
11	LINC	1.0000000	71.330	42.580
12	RADC	1.0000000	75.420	43.250
13	CASE	1.0000000	81.750	41.500
14	CMU	1.0000000	79.930	40.500
15	BELV	1.0000000	77.000	39.080
16	SDAC	1.0000000	77.160	38.920
17	MITRE	1.0000000	77.000	39.000
18	ARPA	1.0000000	77.000	39.000
19	RML	1.0000000	80.570	28.250
20	ISI	1.0000000	118.580	34.000
21	RAND	1.0000000	118.580	33.920
22	UCSD	1.0000000	117.160	32.660
23	UCLA	1.0000000	118.520	34.070
24	SDC	1.0000000	118.550	34.020
25	USC	1.0000000	118.350	34.000
26	DOCB	1.0000000	105.000	39.500
27	GWC	1.0000000	96.000	41.000
28	WPAFB	1.0000000	84.200	39.750
29	ILLI	1.0000000	88.500	40.080
30	UTAH	1.0000000	111.830	40.660
31	LBL	1.0000000	122.280	37.830
32	LLL	1.0000000	121.750	37.630
33	SRI	1.0000000	122.160	37.360
34	XEROX	1.0000000	122.170	37.300
35	TYMSH	1.0000000	121.900	37.330
36	FNWC	1.0000000	121.920	36.500
37	UCSB	1.0000000	119.750	34.500
38	STAN	1.0000000	122.170	37.300
39	AMES	1.0000000	122.030	37.280
40	MOFF	1.0000000	122.030	37.280
41	BBNT	1.0000000	71.250	42.500
42	AMET	1.0000000	122.030	37.280

```
;  <NAC>NSCENARIO.;1   MON 20-MAY-74 9:35AM           PAGE 1:2


ENTER PROPERTY COMMAND OR ? <;WE HIT CARRIAGE RETURN TO MOVE UP THE COMM
                                          **AND HIERARCHY

ENTER SYSTEM COMMAND OR ? <E
ENTER EDITING COMMAND OR ? <AL 15 20;WE ADD A NEW CROSS COUNTRY LINK
LINK ADDED:        15  BELV        20  ISI
ENTER EDITING COMMAND OR ? <
ENTER SYSTEM COMMAND OR ? <P
ENTER PROPERTY COMMAND OR ? <SAN FP 0;WE ARE GOING TO DO A RELIABILITY A
                                          **NALYSISWHERE ONLY LINKS FAIL SO WE
                                          ** SET ALL NODE FAILURE PROBABILITIE
                                          **S TO ZERO
ENTER PROPERTY COMMAND OR ? <TAN ;WE CHECK TO SEE IF IT WORKEED
   NODE          FP          LONG       LAT
    1 ETAC     0.0000000    77.000    39.930
    2 NBS      0.0000000    77.160    39.130
    3 ABER     0.0000000    77.000    39.000
    4 RUTS     0.0000000    74.450    40.490
    5 HARV     0.0000000    71.250    42.500
    6 NCC      0.0000000    71.250    42.500
    7 BBN      0.0000000    71.250    42.500
    8 CCA      0.0000000    71.330    42.500
    9 MIT2     0.0000000    71.200    42.500
   10 MIT1     0.0000000    71.200    42.500
   11 LINC     0.0000000    71.330    42.590
   12 RADC     0.0000000    75.420    43.250
   13 CASE     0.0000000    81.750    41.500
   14 CMU      0.0000000    79.930    40.500
   15 BELV     0.0000000    77.000    39.080
   16 SDAC     0.0000000    77.160    38.920
   17 MITRE    0.0000000    77.000    39.000
   18 ARPA     0.0000000    77.000    39.000
   19 RML      0.0000000    80.570    28.250
   20 ISI      0.0000000   118.580    34.000
   21 RAND     0.0000000   118.580    33.920
   22 UCSD     0.0000000   117.160    32.660
   23 UCLA     0.0000000   118.520    34.070
   24 SDC      0.0000000   118.550    34.020
   25 USC      0.0000000   118.350    34.000
   26 DOCB     0.0000000   105.000    39.500
   27 GWC      0.0000000    96.000    41.000
   28 WPAFB    0.0000000    84.200    39.750
   29 ILLI     0.0000000    88.500    40.080
   30 UTAH     0.0000000   111.930    40.660
   31 LBL      0.0000000   122.280    37.930
   32 LLL      0.0000000   121.750    37.630
   33 SRI      0.0000000   122.160    37.360
   34 XEROX    0.0000000   122.170    37.300
   35 TYMSH    0.0000000   121.900    37.330
   36 FNWC     0.0000000   121.920    36.500
   37 UCSB     0.0000000   119.750    34.500
```

```
   38 STAN    0.0000000  122.170   37.300
   39 AMES    0.0000000  122.030   37.280
   40 MOFF    0.0000000  122.030   37.280
   41 BBNT    0.0000000   71.250   42.500
   42 AMET    0.0000000  122.030   37.280
ENTER PROPERTY COMMAND OR ? <TL 15 20;WE MUST INPUT PROPERTIES OF NEW LI
                                        **NK
     NODE PAIR              FP         CAP      FLOW     COST
   15 BELV    20 ISI    0.0000000      0.00      0.00      0.00
ENTER PROPERTY COMMAND OR ? <
                                   SL 15 20 FP=1. CAP.50;WE TRY ALTERNATE DELI
                                        **MITERS
ENTER PROPERTY COMMAND OR ? <↑C

ENTER SYSTEM COMMAND OR ? <P
ENTER PROPERTY COMMAND OR ? <TL 15 20 00

     NODE PAIR              FP         CAP      FLOW     COST
   15 BELV    20 ISI    1.0000000     50.00      0.00      0.00
ENTER PROPERTY COMMAND OR ? <
ENTER SYSTEM COMMAND OR ? <REL;NOW WE GO TO THE RELIABILITY PROGRAM
ENTER RELIABILITY COMMAND OR ? <?
ONE OF THE FOLLOWING:
  R       RUN RELIABILITY;    LP   LIST PARAMETERS
  CP      CHANGE PARAMETER;   DP   DEFAULT PARAMETERS
  T       TABULAR OUTPUT;     G    GRAPHIC OUTPUT
  RJS     WRITE RJS FILE FOR CCN/91
ENTER COMMAND OR <CR> <
                              LP
RELIABILITY PARAMETERS ARE:
 NSAMP                  100  NUMBER OF SAMPLES
 SEED                     0  RANDOM NUMBER SEED
 MAXP             1.0000000  PROBABILITY FACTOR
ENTER RELIABILITY COMMAND OR ? <CP
ENTER   KEYWORD,VALUE   OR ? <NSAMP 10;WE REDUCE NUMBER OF SAMPLES IN SI
                                        **MULATION TO SAVE COMPUTER TIME
VALUE OF NSAMP CHANGED FROM                100 TO            10
ENTER   KEYWORD,VALUE   OR ? <
ENTER KEYWORD OR <CR>     <
ENTER RELIABILITY COMMAND OR ? <R;WE NOW RUN THE RELIABILITY ANALYSIS

COMPUTATIONS PROCEEDING
ENTER RELIABILITY COMMAND OR ? <T;ASK FOR TABULAR OUTPUT
ENTER OUTPUT COMMAND OR ? <?
ONE OF THE FOLLOWING:
  TTY:     TO TELETYPE;    F        TO DISK FILE
  <CR><CR> TO CANCEL
ENTER OUTPUT COMMAND OR ? <TTY:
↑L
```

```
NAC RELIABILITY RUN:
NUMBER OF NODES=  42    NUMBER OF BRANCHES=  49
SEED=          0        NUMBER OF SAMPLES=         10
MAX.PROB.=   1.00000    PROB.INCR.=   0.05000
HIT <CR> TO CONTINUE; <N> TO STOP <
```

EXPECTED FRACTION OF NODE PAIRS NOT COMMUNICATING

FACTOR	EXPECTATION	VARIANCE	STANDARD DEV
0.00	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.05	0.2984901E-01	0.4846934E+00	0.6961992E+00
0.10	0.1429733E+00	0.2440847E+01	0.1562321E+01
0.15	0.2430894E+00	0.4248271E+01	0.2061133E+01
0.20	0.4104530E+00	0.4663419E+01	0.2159495E+01
0.25	0.6305459E+00	0.1115992E+01	0.1056405E+01
0.30	0.7034843E+00	0.8507560E+00	0.9223644E+00
0.35	0.7773519E+00	0.6032070E+00	0.7766640E+00
0.40	0.8554007E+00	0.2932462E+00	0.5415221E+00
0.45	0.9049942E+00	0.1156519E+00	0.3400763E+00
0.50	0.9272938E+00	0.3703107E-01	0.1924346E+00
0.55	0.9345109E+00	0.3886324E-01	0.1971376E+00
0.60	0.9493612E+00	0.3178136E-01	0.1782733E+00
0.65	0.9610918E+00	0.2711382E-01	0.1646628E+00
0.70	0.9699187E+00	0.1457462E-01	0.1207254E+00
0.75	0.9764229E+00	0.1061702E-01	0.1030389E+00
0.80	0.9830430E+00	0.7367886E-02	0.8583639E-01
0.85	0.9882695E+00	0.3424797E-02	0.5852176E-01
0.90	0.9919861E+00	0.2309814E-02	0.4806053E-01
0.95	0.9962834E+00	0.4602207E-03	0.2145275E-01

```
HIT <CR> TO CONTINUE; <N> TO STOP <
```

PROBABILITY OF NET DISCONNECTED

FACTOR	EXPECTATION	VARIANCE	STANDARD DEV
0.00	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.05	0.2000000E+00	0.1600000E-01	0.1264911E+00
0.10	0.7000000E+00	0.2100000E-01	0.1449138E+00
0.15	0.7000000E+00	0.2100000E-01	0.1449138E+00
0.20	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.25	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.30	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.35	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.40	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.45	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.50	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.55	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.60	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.65	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.70	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.75	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.80	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.85	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.90	0.1000000E+01	0.0000000E+00	0.0000000E+00
0.95	0.1000000E+01	0.0000000E+00	0.0000000E+00

```
;  <NAC>NSCENARIO.;1    MON 20-MAY-74 9:35AM          PAGE 2:1

DONE.
ENTER RELIABILITY COMMAND OR ? <
ENTER SYSTEM COMMAND OR ? <ROUT;NOW WE DO A ROUTING ANALYSIS FOR THE SAM
                                      **E NET

ENTER ROUTING COMMAND OR ? <?
ONE OF THE FOLLOWING:                 DF - DEFAULT ROUTING PARAMETERS
R  - RUN ROUTING ANALYSIS              Q - QUIT ROUTING SECTION
LP - LIST ROUTING PARAMETERS          TR - CHANGE TRAFF. REQUIREMENTS
CP - CHANGE ROUTING PARAMETERS         O - SEE OUTPUT      D - DRAW NETWORK

ENTER ROUTING COMMAND OR ? <LP

ROUTING PARAMETERS ARE:
KEYWORD              CURRENT VALUE      DESCRIPTION

 ITMAX                     5        DESIRED NUMBER OF ITERATIONS
 PKLEN              0.500000        AVG. PACKET LENGTH (KBITS)
 DELAY              0.200000        AVG. PACKET DELAY (SEC)
 PROVR              0.350000        PROTOCOL OVERHEAD (FRACTION)
 RUOVR              0.070000        ROUTING OVERHEAD (FRACTION)
 NDPRS              0.001000        NODAL PROCESSING TIME (SEC)
 UNIRQ              1.000000        UNIFORM TRAFFIC REQUIREMENT (KB/S)
 PRPGA              0.000008        LINE PROPOGATION DELAY (SEC/MILE)
 THACC              0.000100        THROUGHPUT ACCURACY  (%)
 TIACC              0.000100        TIME DELAY ACCURACY  (SEC)
 NCAP                      2        NUMBER OF DIFFERENT CAPACITIY OPTIONS
 CAP1                 50.000        CAPACITY(KB/S) OF CAP OPTION # 1
 FIX1                850.000        SUM OF ALL FIXED COSTS FOR CAP OPT. # 1
 RAT1                  5.000        RATE/MILE FOR CAPACITY OPTION # 1
 CAP2                230.000        CAPACITY(KB/S) OF CAP OPTION # 2
 FIX2               1300.000        SUM OF ALL FIXED COSTS FOR CAP OPT. # 2
 RAT2                 30.000        RATE/MILE FOR CAPACITY OPTION # 2

FOR HELP ON THESE PARAMETERS, ENTER THE ROUTING COMMAND <CP>
ENTER ROUTING COMMAND OR ? <CP
ENTER   KEYWORD,VALUE   OR ? <HELP;WE WANT MORE INFORMATION
ONE OF THE FOLLOWING:
 <KEYWORD>,<VALUE>        TO CHANGE VALUES
 <CR><CR>                 TO TERMINATE
 LIST                     LISTS KEYWORDS AND VALUES
 HELP <KEYWORD>           DESCRIBES <KEYWORD>
 HELP ALL                 DESCRIBES ALL KEYWORDS
ENTER   KEYWORD,VALUE     OR ? <
                                    HELP ITMAX
ITMAX: MAX NUMBER OF ROUTING ITERATIONS DESIRED;
ERATIONS TO SAVE RECOMMENDED MAX IS 10, GOOD APPROXIMATION AFTER 5
ENTER   KEYWORD,VALUE    OR ? <
                                    ITMAX=2;WE REDUCE ITMAX TO SAVE TIME
VALUE OF ITMAX CHANGED FROM                   5 TO                2
ENTER   KEYWORD,VALUE    OR ? <
ENTER KEYWORD OR <CR>
```

```
;  <NAC>NSCENARIO.;1   MON 20-MAY-74 8:35AM           PAGE 2:2


ENTER ROUTING COMMAND OR ? <R
 COMPUTATIONS PROCEEDING
AT ITERATION  1, THRUPUT=   280.993 KB/S     DELAY=  0.19997 SEC.
AT ITERATION  2, THRUPUT=   329.897 KB/S     DELAY=  0.19992 SEC.

  DESIRED NO. OF ITERATIONS  REACHED

THRUP=  329.897 KB/S      WHICH IS   19.158 % OF THE BASE REQUIREMENT
 DELAY=  0.19992 SEC    TOTAL COST=  103318. $/MO
ELAPSED TIME :  0 MIN., 21 SEC.        CPU TIME :  0 MIN., 10 SEC.
CONTINUE ROUTING WITH MORE ITERATIONS ?  ENTER Y OR N
<N

ENTER ROUTING COMMAND OR ? <O;WE LOOK AT OUTPUT
ENTER OUTPUT COMMAND OR ? <?

ONE OF THE FOLLOWING:
    LF   - LOOK AT LINK FLOWS AND COSTS OF MOST RECENT RUN
    T    - LOOK AT THROUGHPUT-DELAY TABLE OF MOST RECENT RUN
    G    - DRAW THROUGHPUT-DELAY CURVES
    Q    - QUIT OUTPUT
ENTER OUTPUT COMMAND OR ? <T
SPECIFY OUTPUT DEVICE OR ?<?
ONE OF THE FOLLOWING :
TTY  - OUTPUT TO TELETYPE        F - OUTPUT TO DISK FILE
?    - PRODUCES THIS LIST        Q - QUIT THIS SPECIFIC OUTPUT
HC - OUTPUT TO HARDCOPY DEVICE (IF SLAVE IS AVAILABLE)
SPECIFY OUTPUT DEVICE OR ?<F

ENTER 5 CHARACTER FILE NAME<ROUTO

FILE  ROUTO.DAT  OPENED ON DISK UNIT # 21
SPECIFY OUTPUT DEVICE OR ?<
SPECIFY OUTPUT DEVICE OR ?    <

ENTER OUTPUT COMMAND OR ? <

ENTER OUTPUT CONMMAND OR ?    <

ENTER ROUTING COMMAND OR ? <
ENTER ROUTING COMMAND OR ?    <
ENTER SYSTEM COMMAND OR ? <
ARE YOU ALL DONE? Y OR N? <Y


CPU TIME: 45.10 ELAPSED TIME: 35:14.15
NO EXECUTION ERRORS DETECTED

EXIT.
```

REFERENCES

1. Balinski, M., "Integer Programming: Methods, Uses, Computations," Proceedings of the Princeton Symposium on Mathematical Programming, Princeton University-Press, Princeton, 1970, pp. 199-266.

2. Coxeter, H., Introduction to Geometry, J. Wiley, New York, 1961.

3. Edmonds, J., "Covers and Packing in a Family of Sets," Bulletin American Math. Soc., 68, pp. 494-499.

4. Ford, L.R. Jr., and D.R. Fulkerson, Flows in Networks, Princeton, Chapter II, 1962.

5. Fulkerson, R., "Blocking and Anti-Blocking Pairs of Polyhedra," Mathematical Programming, 1, pp. 168-194, 1971.

6. Fulkerson, D., G. Nemhauser, and L. Trotter, "Two Computationally Difficult Set Covering Problems That Arise in Computing the 1-Width of Incidence Matrices of Steiner Triple Systems," Tech. Report 903, Cornell University, Department of Operations Research, 1973.

7. Garfinkel, R., and G. Nemhauser, Integer Programming, J. Wiley, New York, 1972.

8. Hu, T.C., Integer Programming and Network Flows, Addison-Wesley, New York, 1969.

9. Martin, G., "Solving Large Scale Mixed-Integer Programs of the Fixed Cost Type," SIGMAP Newsletter, 15, 1973, pp. 27-29.

10. Ryser, H.J., Combinatorial Mathematics, Mathematical Association of America, Chapter 6, 1963.

11. Steiglitz, K.P., Weiner and D. Kleitman, "The Design of Minimum Cost Survivable Networks," IEEE Trans. Circuit Theory, CT-16, pp. 455-460, 1969.

12. Toth, F.L., Lagerungen in der Ebene auf der Kugel und im Raum, Springer-Verlag, Berlin, 1953.

13. Van Slyke, R., and H. Frank, "Network Reliability Analysis-I," Networks, Vol. 1, No. 3, 1972, pp. 279-290.

14. Walkup, D., "On a Branch-and-Bound Method for Separable Concave Programs," Boeing Scientific Research Laboratory, Report D1-82-0670, September 1967.

IMPACT OF INTERACTIVE GRAPHICS ON NETWORK DESIGN

I. INTRODUCTION

Recently NAC has developed an interactive program that analyzes packet switching networks for cost, throughput delay, and reliability performance. Input and output is handled via a graphic terminal, which displays network topology, allows easy topological reconfiguration and shown curves of network performance (delay, reliability, etc.) versus a variety of network parameters (throughput, failure rates, etc.).

The details of the algorithms used in the analysis are presented in [1], [2], and [3]. This chapter is a description of the interactive program and it's use, the program's performance, a short comparison with an equivalent bath program and implications of interactive graphics on network design.

II. FUNCTIONAL SPECIFICATIONS OF THE INTERACTIVE NETWORK ANALYSIS PROGRAM

NAC's interactive network analysis program called RELROUT is located on the (NAC) directories at USC-ISI and BBN for use by any network user. It runs on a PDP-10 using the TENEX operations system. The program can be run from any type of terminal, but will only support graphics on an IMLAC PDS-10 graphics display unit.

The program analyzes a packet switching network for routing and reliability performance.

The user must specify:

- A network topology.
- Properties of nodes and links.
- General parameters such as packet size, etc.

The topology can be entered in two ways:

1. Create a new network configuration through the use of network editing commands; or

2. Read an existing configuration from a previously generated data file.

The second method of inputting a network is very useful when repeated evaluation of the same basic topology are required. On an IMLAC terminal, the user may move the position of the nodes on the CRT in order to obtain a clearer and more intelligible graphic representation of the network.

The user defines the following node and link properties:

A. Node Properties

1. Name (Optional)

2. Location (longitude and latitude, for cost calculation in the routing analysis.

3. Failure probability (for reliability analysis).

4. Symbol type - □,△,◊,○ (for display purposes).

B. Link Properties

1. Capacity or line speed (for routing and cost analysis)

2. Failure probability (for reliability analysis).

3. Link type - solid or dotted (for display purposes).

After the user has defined the network configuration and properties, he can request either a routing or a reliability analysis to be performed. Associated with each analysis are various general parameters.

For a routing analysis these are:

1. Average packet delay.

2. Average packet length.

3. Overhead

4. Tariff Structure, etc.

For the reliability analysis the parameters are:

1. Number of samples.

2. Random number seed.

3. Range of variation for the probabilities.

Initially all of these parameters are defaulted to specified values. However, the user can change any or all of them. The program does error and validity checking on the parameters that are changed. Also, upon user request, a short description of each parameter can be provided.

When the user is satisfied with the values of the various parameters, he can ask for the analysis to begin. In most cases, the processing is done locally in the PDP-10; however, the reliability analysis can be performed remotely, as discussed later in this chapter.

When the analysis is completed, the user can examine various outputs. The routing analysis output consists of a list of link flows, lengths, and costs, and values for global throughput, delay and cost.

There is also a table of network throughput and delay as a function of relative traffic. Reliability output consists of tables of P_{nc} (probability of network disconnected) and F_{nc} (fraction of node pairs not able to communicate) as a function of component failure rates. If the user's terminal can support graphics (IMLAC), the program will plot curves for throughput vs delay and for P_{nc} and F_{nc} of the most recent run along with at most two previous runs. (See Figures 1,2, and 3).

After examining the output of the specific analysis, the user typically will do one of the following:

1. Perform a sensitivity analysis on the present network configuration by varying the input parameters.

2. Switch to the other analysis (routing reliability or vice versa).

3. Edit the topology (add nodes and/or links) and/or change node or link property values and proceed with another routing or reliability analysis.

4. Exit the program.

At any time during the interactive session, the user may save his network topology on a data file for use at a later time. After leaving the program, the user is returned to the TENEX operating system.

NAC's interactive program assists the network analyst by:

1. Offering a schematic representation of the current network configuration on a graphic display device;

2. Allowing interactive error checking of topology and input parameters so that the user can correct the input data immediately;

3. Removing the necessity of explicitly entering all the data for each run and allowing flexibility in the order and format of input;

4. Using a flexible command structure, with numerous prompts to tutor the novice user if he requests help; and with short, quick commands for the experienced user.

5. Providing network performance results with small enough response time, so that an effective man-machine interactive design can be carried out.

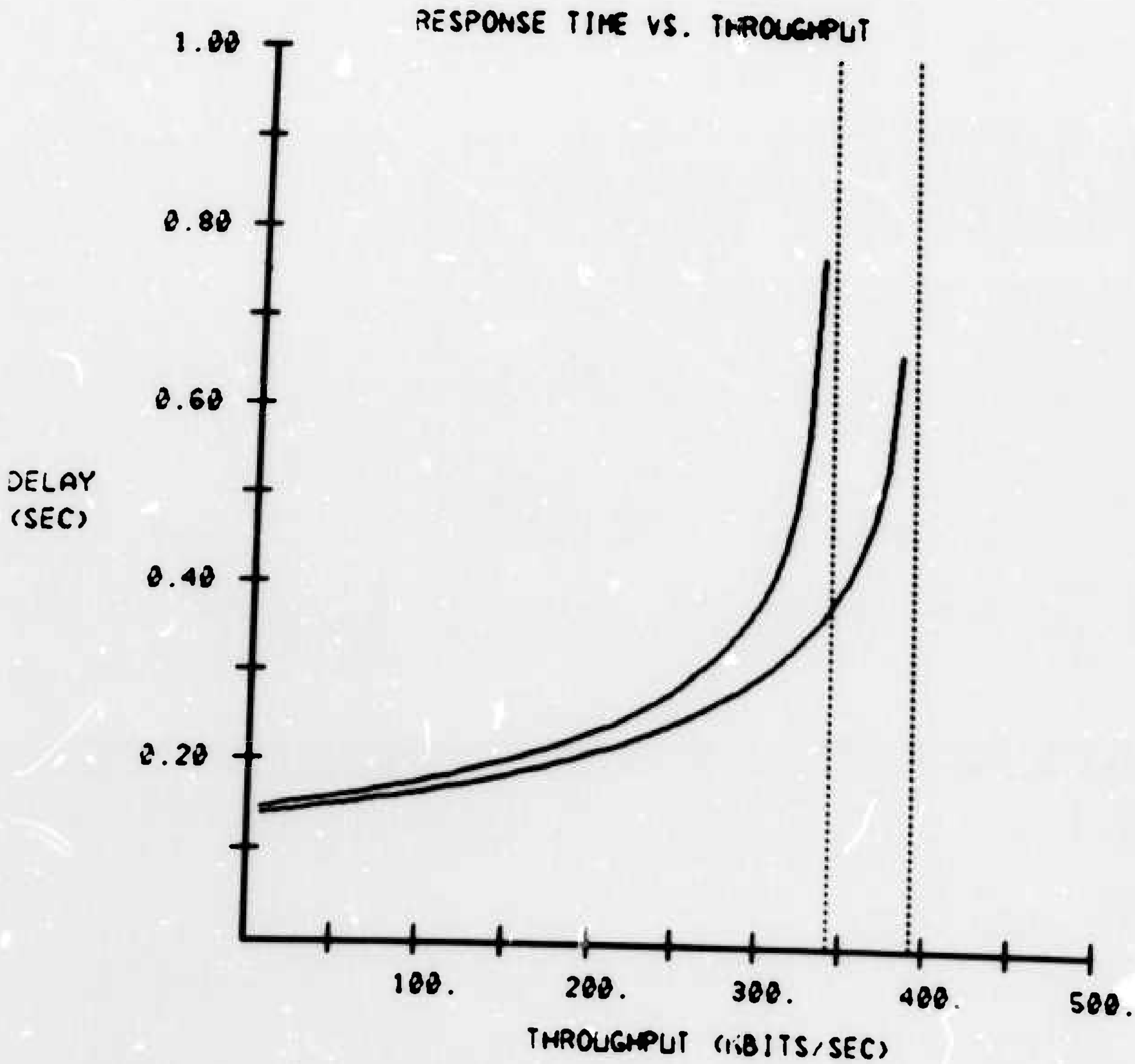
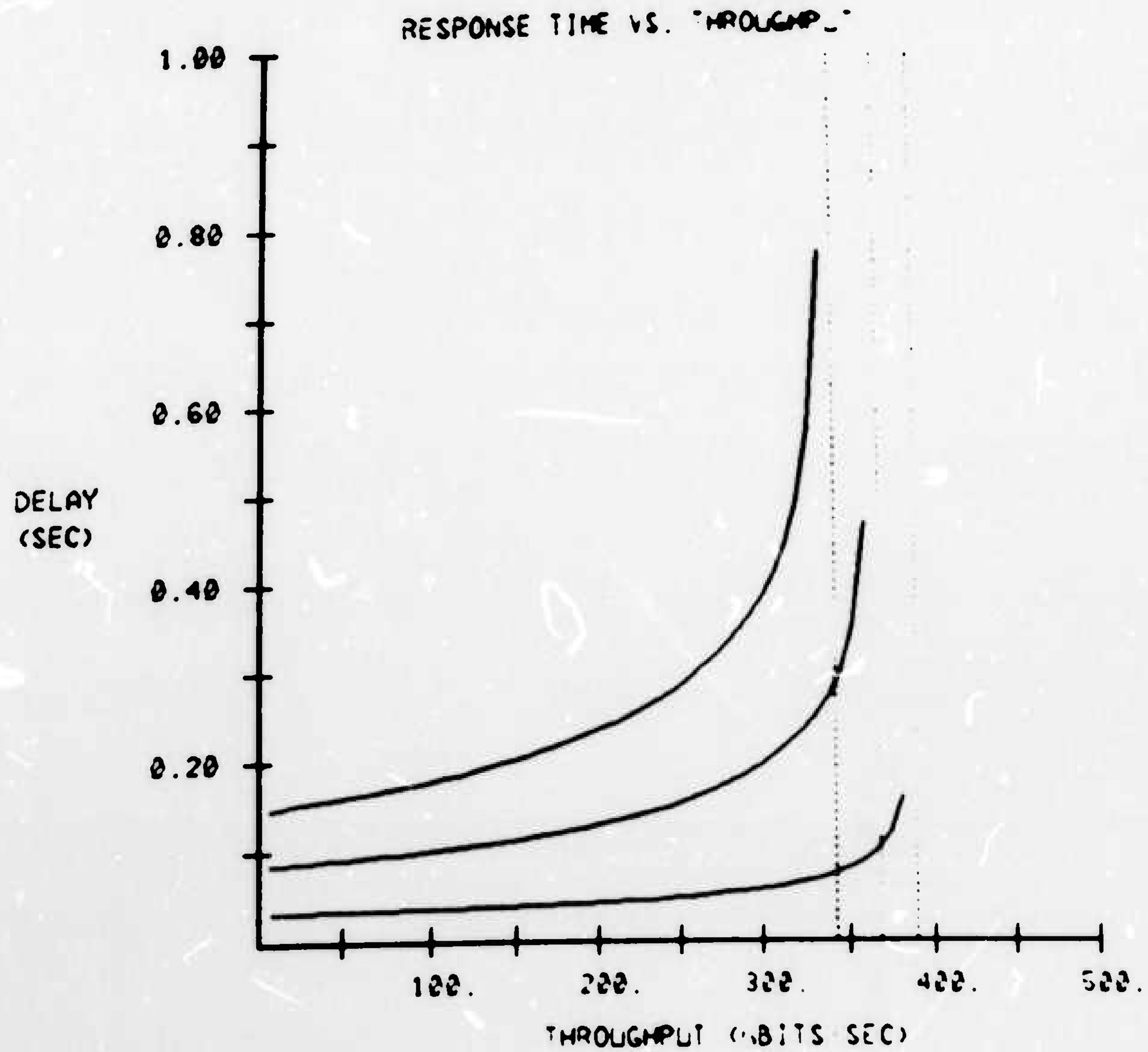


FIGURE 1:

42 NODE ARPANET CONFIGURATION, ALL 50 KB/S LINES VS. INCREASE OF 1 CROSS COUNTRY CHAIN TO 230 KB/S



ENTER OUTPUT COMMAND OR ? <_

FIGURE 2:

42 NODE ARPANET CONFIGURATION 100, 500, 1000 BIT PACKETS

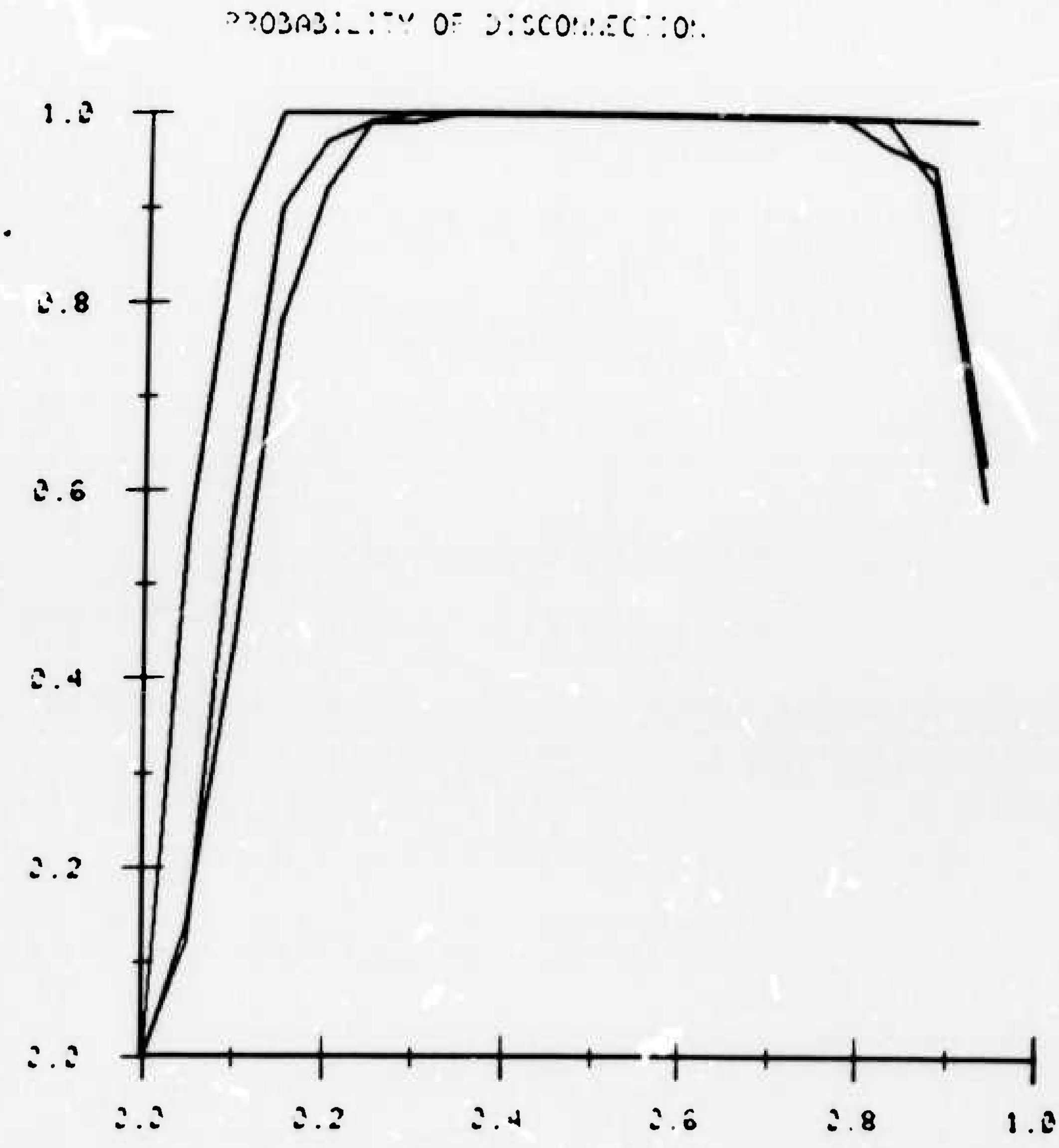


FIGURE 3:

100 SAMPLE RELIABILITY RUNS, ONLY NODES FAIL, ONLY LINKS FAIL, BOTH NODES AND LINKS FAIL

III. TURNAROUND DELAY PERFORMANCE OF INTERACTIVE PROGRAMS

A. General

Turnaround delay is probably the most important performance criterion for the evaluation of the effectiveness of an interactive analysis and design program. In fact, the purpose of an interactive program implementation is to provide the systems analyst with faster answers and, therefore, better man-machine interaction than he could obtain with the batch version of the same program. In this section, the turnaround time performance of our routing and reliability interactive programs is evaluated in a variety of load conditions.

Turnaround time is here defined as the delay between the time the RUN command (which starts execution) is entered and the time the output comes back on the screen. In a time sharing environment, such a delay is approximately proportional to the average number of tasks (system and user) simultaneously requesting the CPU; in a Tenex system, such a number is referred to as the Load Average.

B. Routing Program Turnaround Delay

NAC's routing program is based on an iterative algorithm which attempts to raise network throughput while maintaining the specified delay constraint. Therefore, the CPU time is proportional to the number of iterations performed. In the following experiments, five iterations were allowed for each run, since at that point, a sufficient accuracy was generally obtained.

Several routing runs were performed on three network examples with 10, 26, and 42 nodes, respectively, using NAC's interactive program at USC-ISI. The same network was analyzed at different times during the day, and with different load averages. The curves in Figure 4, 5, and 6 show the elapsed time versus load average for each network application.

For networks on the order of 10 nodes, the elapsed time is quite tolerable for any reasonable value of load average (Note: the typical range of load averages is between 2 and 8). Furthermore, intermediate throughput and delay results are printed at the terminal after each routing iteration, to keep the user's attention while computation is proceeding.

For larger networks (20 nodes or more), load average has a more critical impact on elapsed time. In the case of high load average, the user can avoid long delays by reducing the number of iterations and thus sacrificing accuracy; alternatively, the user can transfer his files to another site with lower load average and run his programs there. This is one of the attractive features offered by a resource sharing network such as ARPANET.

C. Performance of Interactive Reliability Analysis

In general, the reliability of a network with a size such as ARPANET's does not vary drastically with the insertion or deletion of a few links. Furthermore, the reliability criteria, P_{nc} and F_{nc}, will not change at all if only line speeds are varied. The routing performance, on the other hand, can be substantially affected by the above modifications. Therefore, network reliability is usually not evaluated after each topological modification and routing performance evaluation. The designer might, however, want to examine the network reliability after several topological changes. Therefore, an interactive network design "package" should include a reliability program.

The reliability program currently included in NAC's interactive network analysis and design program is essentially a simulation program [2]. The accuracy of the results is, therefore, related to the number of samples generated in the simulation. Table 1 shows the results for several reliability analyses using 1000 samples for networks of 10, 26, and 42 nodes. The runs were made at ISI at approximately 8:30 EDT (5:30 PDT).

TABLE 1
RELIABILITY PROGRAM PERFORMANCE

	10 Nodes	26 Nodes	42 Nodes
Load Average	1.6	1.5	2.5
Elapsed Time	1 min. 49 sec.	3 min. 58 sec.	9 min. 35 sec.
CPU Time	59 sec.	2 min. 37 sec.	4 min. 18 sec.

As can be seen, the elapsed time for the reliability analysis, even at very low load averages, tends to reach levels which are intolerable for interactive analysis.

Because of the extensive amount of computation required for a reliability analysis, it would be desirable to have the analysis be done in batch on a big "number crunching" machine, and yet allow the user the flexibility of interactive editing and validation of data and graphic display of output. This feature can actually be implemented on ARPANET. In fact, with the remote job service (RJS) in association with the IBM 360/91 at UCLA, the user does have the option of local or remote processing of the reliability analysis. He can direct the interactive program to perform the analysis locally by entering the command R(UN). Or, he could request the interactive program to create an RJS data file complete with the job control language (JCL) needed for execution. The user can then leave the interactive program, enter the RJS subsystem, and submit the reliability job file. He can wait for the output or he could return to the interactive program and, perhaps, continue with a routing analysis; periodically checking to see if the IBM 360/91 has completed the execution of the reliability analysis.

Once the output is ready, the data file is read into the interactive program and various reliability curves can be displayed. Efforts are currently being made to make this RJS feature invisible to the user. However, even now this feature of parallel processing, made available by ARPANET resource-sharing capability, is very useful for any interactive computation.

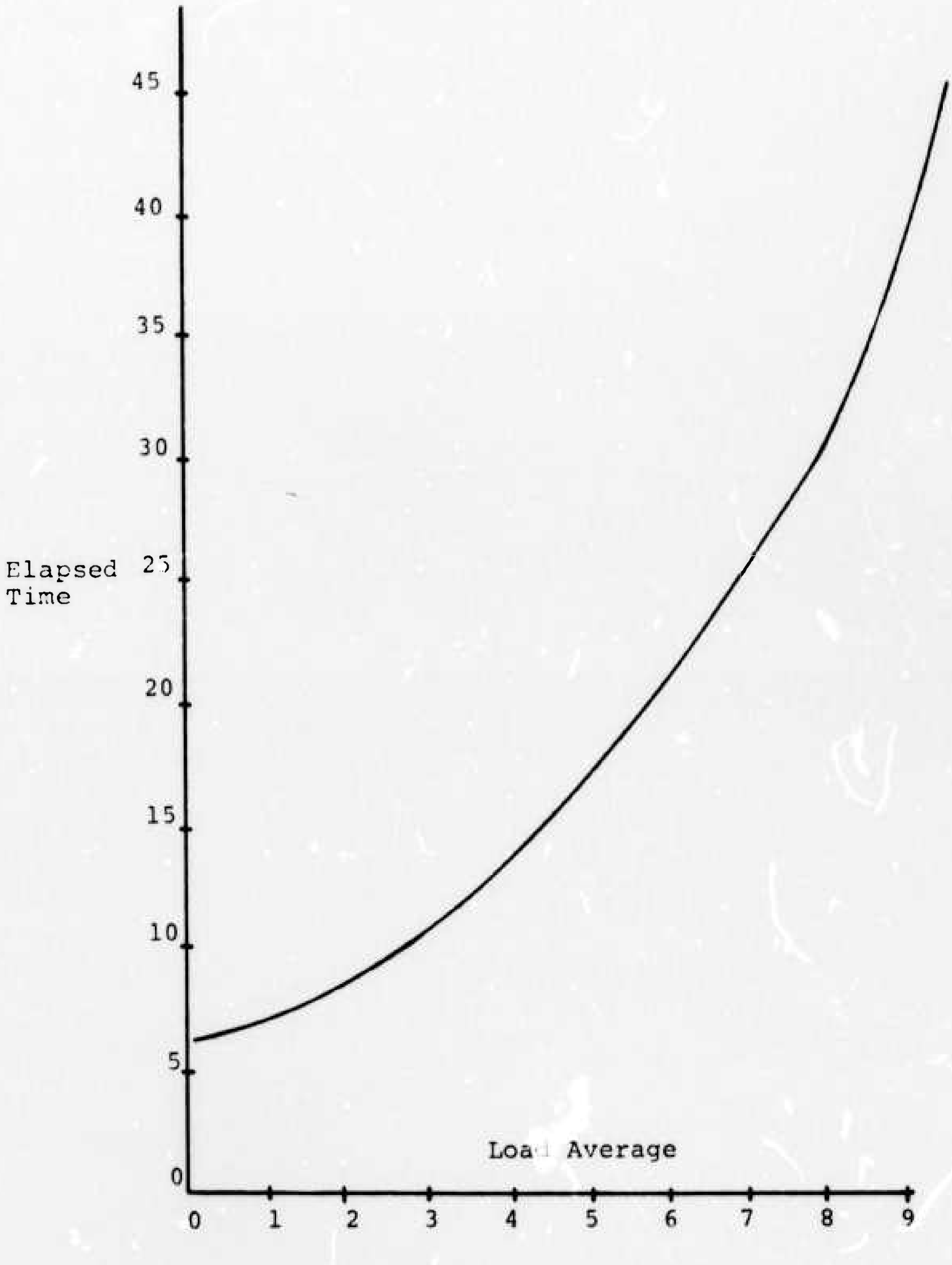


FIGURE 4:

10-NODE NETWORK CPU TIME 3 SEC.

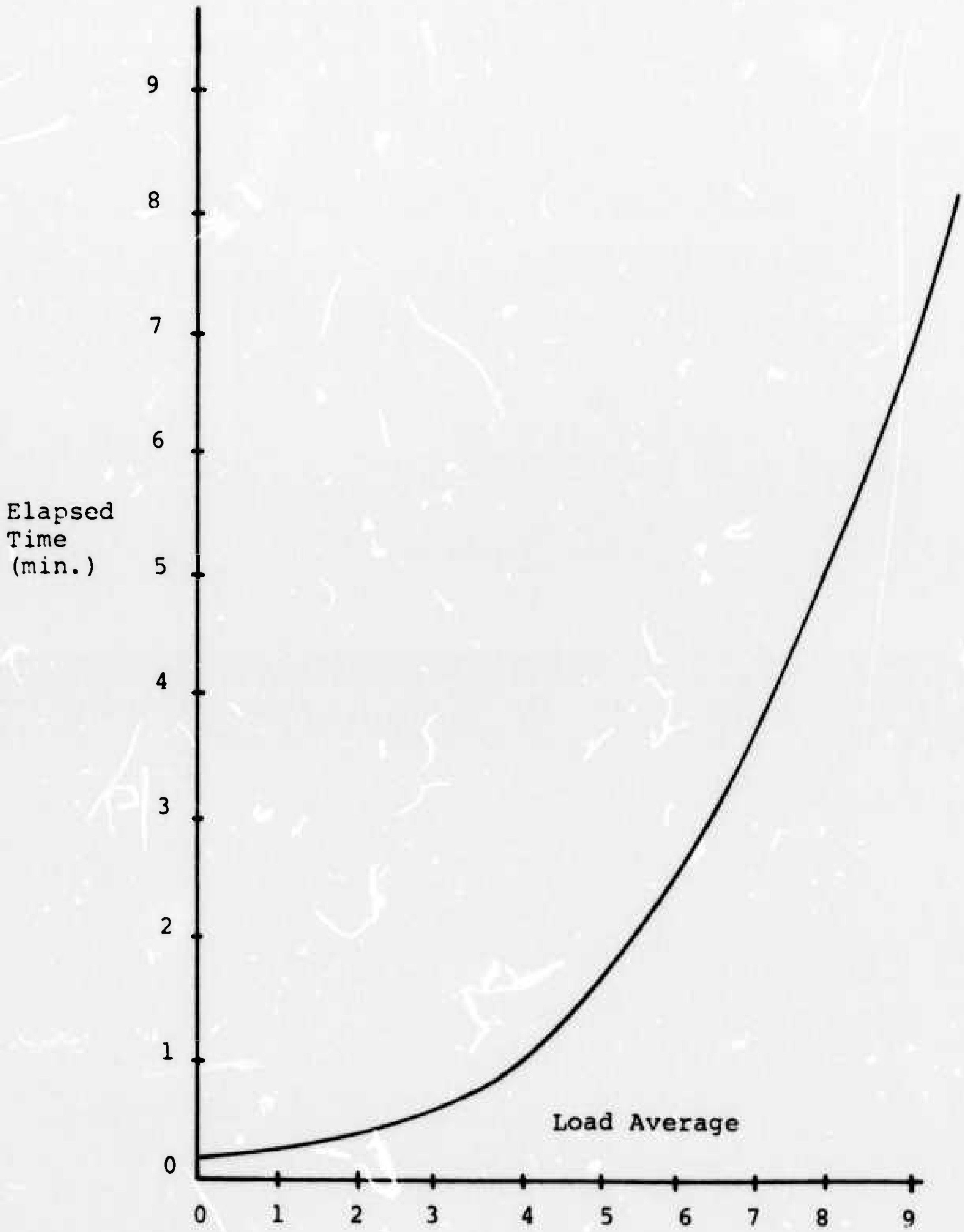


FIGURE 5:

26-NODE NETWORK CPU TIME - 11 SEC.

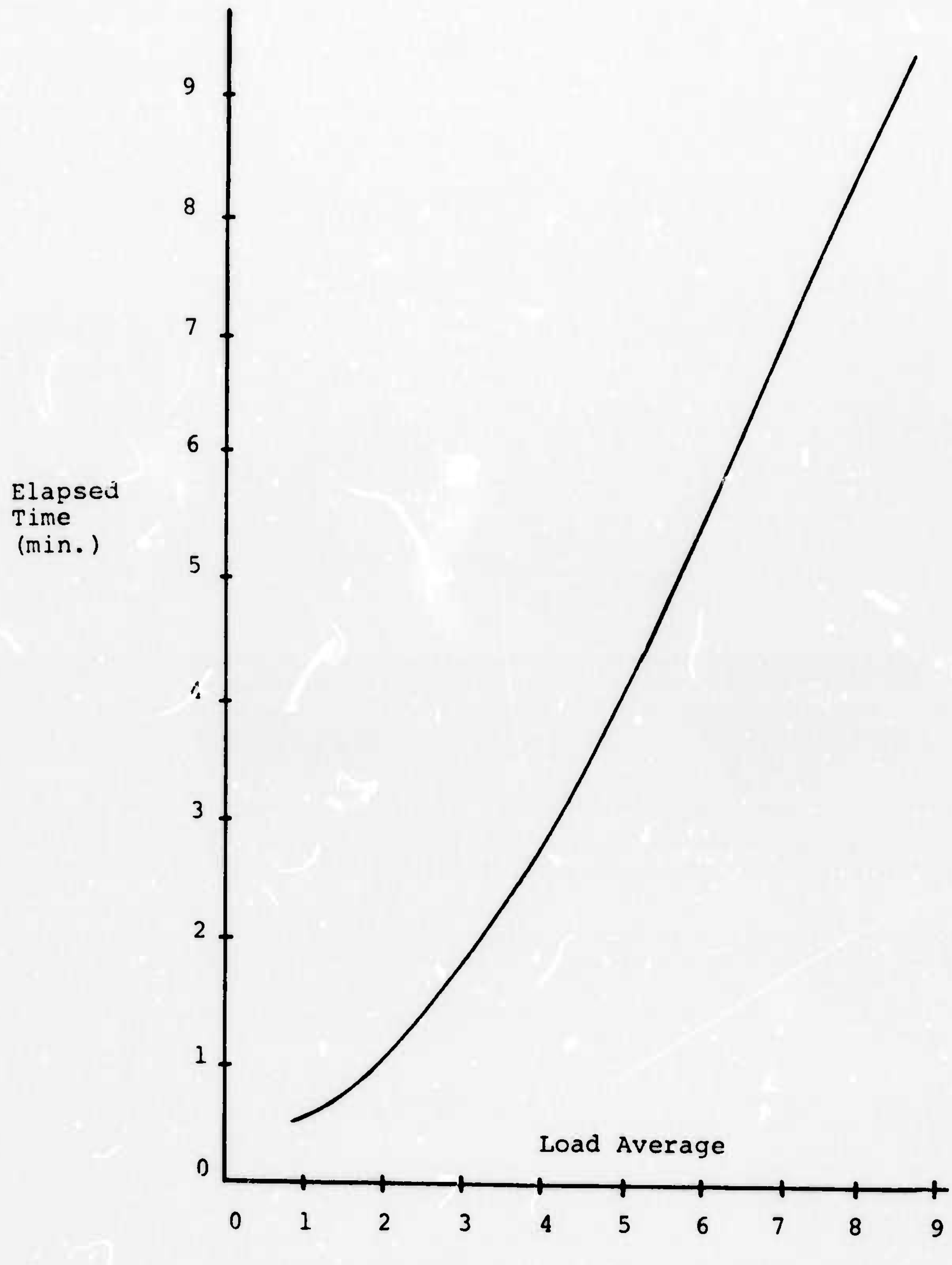


FIGURE 6:

42-NODE NETWORK CPU TIME 24 SEC.

IV. INTERACTIVE VERSUS BATCH NETWORK DESIGN

It is possible that a one-time network evaluation could be performed faster and more economically in a batch mode than in an interactive mode. However, what we are concerned with here is a complete design session in which a network is analyzed and modified repeatedly using man-computer interaction. In this section, we compare the overall length of time involved in a design session using either a batch program or an interactive program. A design session consists of: analysis of a given network configuration; use of results to modify the topology in order to improve network performance (reduce cost, improve throughput and reliability); re-evaluation of the new network configuration, etc.

Table 2 shows the steps a designer would take in a "design session", evaluating routing performance, using an interactive or a batch approach.

In Step I1, the user sets up interactively the network configuration and the data base internal to the program. In the batch case, the designer must first draw the configuration and then keypunch the appropriate cards. The setting up of the routing parameters (I2 and B2) requires basically the same amount of time in both interactive and batch mode; however, the interactive program displays on the screen a description for each input parameter, and performs validity and error-checking on the input data, thereby avoiding error and confusion.

In Step I3, the user enters the R(UN) command and execution of the routing analysis begins. In the batch case, (B3.1 → B3.5), the user initiates the job and receives the output. Even though the actual batch computation is generally faster than the interactive computation, (5 sec. for a 42 node network on a CDC 6600 as opposed to 24 sec. on a PDP 10 Tenex), there is quite a fixed delay before and after job execution which is independent of CPU time requirements (read-in deck, job waiting on input and output queue, wait for printer).

This delay can be on the order of 15 - 20 minutes, sometimes even longer.

In Step I4 or B4, the designer examines the results of the analysis. The time required to examine the data on the screen is comparable to the time required to read the same data on the print out; however, in the interactive mode, the designer can instantly compare the present results with those obtained from previous runs (most notably plots of throughput versus delay). After the topological modifications are made on the configuration displayed on the CRT (Step I5), the designer is ready to perform another analysis and returns to Step I3. In the batch case, on the other hand, the designer must redraw his new configuration by hand and then keypunch the appropriate cards. He then must return to the card-reader to initiate a new analysis.

Thus, for the first iteration of the analysis, the effort on the designer is comparable for both the interactive and batch approaches. However, for the subsequent iterations, the effort is considerably less with the interactive procedure.

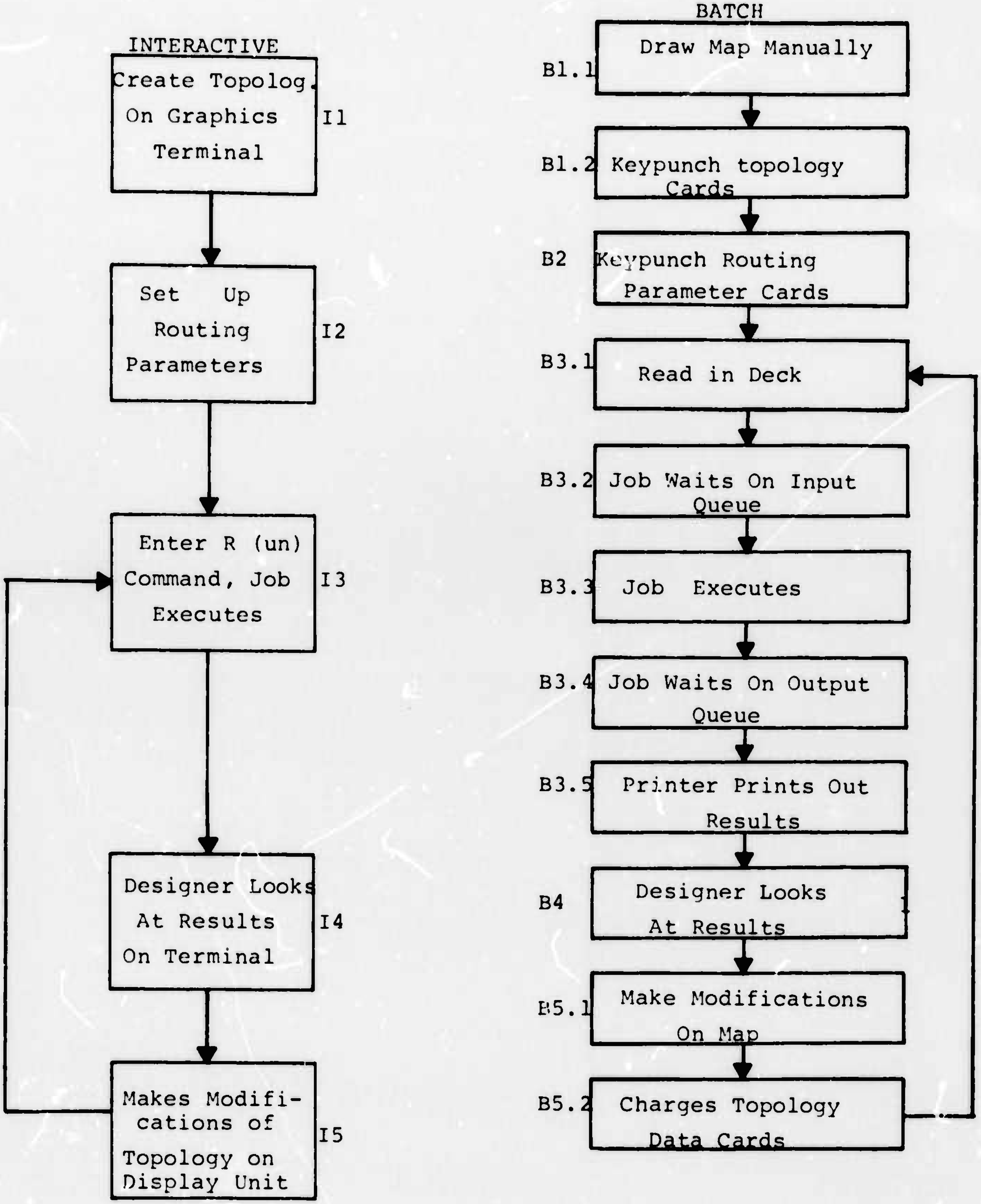

INTERACTIVE
Create Topolog
On Graphics
Terminal
I1
Set Up
Routing
Parameters
I2
Enter R (un)
Command, Job
Executes
I3
Designer Looks
At Results
On Terminal
I4
Makes Modifi-
cations of
Topology on
Display Unit
I5
BATCH
B1.1
Draw Map Manually
B1.2
Keypunch topology
Cards
B2
Keypunch Routing
Parameter Cards
B3.1
Read in Deck
B3.2
Job Waits On Input
Queue
B3.3
Job Executes
B3.4
Job Waits On Output
Queue
B3.5
Printer Prints Out
Results
B4
Designer Looks
At Results
B5.1
Make Modifications
On Map
B5.2
Charges Topology
Data Cards

TABLE 2

V. IMPACT OF INTERACTIVE GRAPHICS ON NETWORK DESIGN

The most conspicuous effect of interactive graphics on network design is that of speeding up the overall design process in a very substantial manner. The amount of time saved and the increase in productivity clearly depend on many factors (efficiency of the interactive graphic system, programmers experience, load on the computer, etc.); our experience indicates that the design process is typically speeded up by 5 to 10 times.

Another important effect of interactive graphics on network design is the interactive use of analysis programs to develop better design algorithms. In fact, most network design algorithms are based on heuristics, and good heuristics are obtained by combining physical intuition, careful observation of several network properties, and evaluation of several examples. Therefore, an interactive program is an extremely valuable tool in the hands of a network designer who is trying to establish experimentally some general relationship between network configuration, topological transformations and network performance.

Besides assisting the network designer in the development of better heuristics, interactive graphics can also provide a way to monitor design algorithms which are completely automatized, and in principle would not require human intervention. Since most network design algorithms are iterative and typically perform a topological transformation at each step, it is possible to implement them in an interactive mode so that they display the current solution at each iteration. The designer, therefore, can evaluate the cost effectiveness of each transformation, and can stop, correct and restart the algorithm, whenever he identifies some inadequacy in the current solution.

The monitoring and verification function of interactive graphics is very valuable for any heuristic design. In fact, it is extremely rare to find heuristics that perform well on all possible problems. In general, there are always cases in which the heuristic solution is very bad. Sometimes the solution is so bad that the designer can detect and correct the inconsistencies by simple visual inspection. Thus, the importance of visually monitoring the solutions.

VI. EXAMPLE

A practical application of the interactive program arose when NAC was asked to evaluate the possibility of significantly reducing the communication costs of ARPANET and the impact of potential cost reductions on the network's performance. Below is the report summarizing the results.

Requirements used in the study are as follows:

1. Average packet delays under 0.2 seconds throughout the net.

2. Capacity for expansion to 64 IMPs without major hardware or software redesign.

3. Average total throughput capability of 200 - 300 kilobits/second for all Hosts.

4. Peak throughput capability of 40 - 80 kilobits/second per pair of IMPs in an otherwise unloaded network.

5. High communication subnet reliability subject to economic constraints.

The time delay and throughput requirements imply that 50 kilobit/second communication lines are needed within the network. Factors impacting network designs have been:

1. Nine month lead times for obtaining lines from AT&T.

2. Rapid expansion of the number of IMPs and TIPs in the net (averaging one new node per month).

3. Rapid increase in traffic in the network.

The traffic growth in the net prompted us to study, within the present contract year, the problem of increasing the network's traffic capacity and the associated costs of such increases. Now that the network traffic has become relatively stable because of the saturation of serving Hosts and the reduction of the rate of addition of new nodes to the network, the object of the present study was to reevaluate network costs as a function of the various parameters in the network.

Several options are available to reduce cost in the network. These are:

1. Rearrangement of lines.

2. Reduction of line capacities in the network either on a limited basis or throughout the network.

3. Introduction of new technology to reduce overall line costs.

A constraint imposed on all alternatives is that communication subnet reliability should not significantly decrease from that of the present network. This constraint dictates that the network remain two connected and imposes certain other technical conditions upon network topology.

Major modifications to the present network cause reduction to network performance with respect to one or more of the network's performance parameters. A simple rearrangement of lines to reduce cost also reduces the network's expansion capability for at least nine months (since this is the time required to obtain new 50 kilobit/second circuits from AT&T). Throughput is also somewhat reduced.

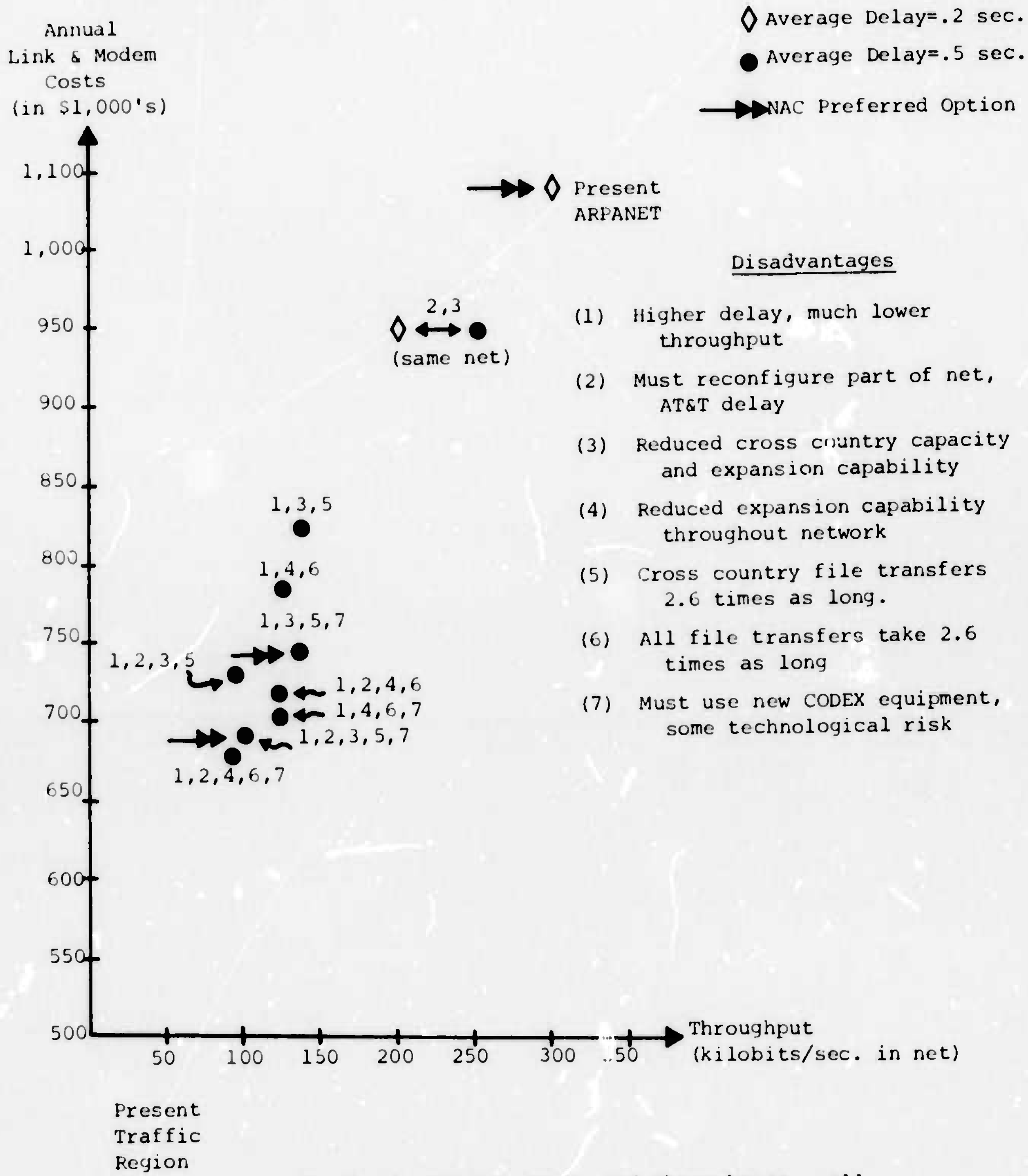
Decrease of line capacities substantially reduces total network throughput, increases time delay, and most significantly, theoretically increases file transfer time by at least a factor of 2.6 (based on available AT&T circuit options). The cases where some or all of the lines are reduced in speed were considered.

If lower speed lines, (in particular, 19.2 kilobit/second lines) are used, the use of a new device called a <u>biplexer</u> to further reduce the cost of some lines is possible. Although many have not yet been installed and thoroughly tested, the technical concept is sound and should a decision to reduce line speeds to 19.2 be made, several biplexers should be obtained and tested, and, if successful, used wherever appropriate. The analyses consider networks both with and without biplexers.

Finally, the case where line speeds are decreased to 9.6 kilobits/second was considered. This results in a network which can only marginally handle existing network traffic, has no expansion capability, very high average time delays (over 1 second) and file transfer times more than five times as great as those achievable at present. While the communication costs of such a network would only be 1/3 of the present networks cost, its performance would be so poor that this option was not extensively examined.

Figure 7 summarizes the available alternatives and communication costs. Options that we prefer given that a decision is made to degrade ARPANET performance are also indicated as are the disadvantages of each alternative. An important point to be understood is that communication costs do not reflect all of the issues that must be made in changing the present ARPANET approach. For example, if all of the network's lines were reduced from 50 kilobits/second to 19.2 kilobits per second (and the network were reconfigured), communication line costs would decrease from about $1.1 million per year to either $736,000 (without biplexers) or $694,000 (with biplexers). However, file transfer time would increase by a factor of 2.6 and the computer connect time costs, to transfer these files, could increase by as much as $72,000 per year. (These calculations are based on available network measurement data).

Using the interactive program, this study took approximately four hours. Using programs that run in batch, this study could have taken several days.



Graph of options, costs and throughputs. All nets have approximately the same reliability. Numbers next to points indicate applicable disadvantages when compared to present ARPANET. NAC preferred options, in addition to present network, incorporate 50 and 19.2 kilobit/second lines and biplexers where appropriate.

FIGURE 7

VII. FUTURE RESEARCH

Future research plans in the area of interactive network design tools include the following items:

1. Implementation of the Cut-Saturation Network Design Algorithm [4] as an interactive program.

2. Development of more efficient and less time consuming reliability analysis algorithms, which would allow more frequent reliability evaluations during the design process.

REFERENCES

1. Fratta, L., M. Gerla, and L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design," Networks, 3:97-133, 1973.

2. Van Slyke, R., and H. Frank, "Network Reliability Analysis I," Networks, Vol. 1, No. 3, 1972

3. Van Slyke, R., and H. Frank, "Reliability of Computer Communication Networks," Proceedings of the 5th Conference on Applications of Simulation, New York, 1971.

4. Network Analysis Corporation, "Issues on Large Network Design," ARPA Report, 1974.

COMPUTATIONAL COMPLEXITY OF NETWORK CONNECTIVITY ALGORITHMS

I. INTRODUCTION

A basic property of a graph is its connectivity structure; that is, how many connected pieces it divides into. The determination of this simple property is fundamental to many more complex calculations. It is equivalent to determining equivalence classes. One important application is in network reliability calculations. Suppose we are given n nodes for a network. There are $n(n-1)/2$ possible distinct undirected links (not counting loops) among them. Suppose we add these links sequentially and in random order. What is the expected number of links we must add to connect all n nodes? This value allows one to estimate a bound on the average running time for a simulation technique for reliability analysis [23]. In this chapter we describe closed form solutions to this and related problems for finite n. The method used to consider the process as Markov processes on the lattice of partitions.

The problem we discuss arises in the evaluation of algorithms for determining spanning trees and/or the connected component structure of graphs and networks. Some of these algorithms are given in [9], [13], [14], and [22]. Mathematically, the problem has its own history and seems to have appeared (in the form studied here) for the first time in [6], although forms of it were studied earlier in [10] and [11].

The problem can be succinctly described as follows: At each point in "time" (sequentially) we select a subset of size two (edge) from a set of n-labelled objects (nodes, vertices) "at random". At random means that each edge not previously selected is equally likely to be picked. A graph generated "at random" is called a "random graph".

What is the probability that the "random graph" is connected after j-edges have been selected? What is the "mean time" to connectedness? More generally, given a connected component structure π, what is the mean time to obtaining a graph with component structure π. We obtain "closed form" solutions and relatively simple computing equations for the problems just posed as well as other combinatorial functions related to the process of generating random graphs. A number of these results have been obtained by other methods by the authors mentioned and others. Our methods are based on the work in [20], [21].

The problems described above have a long and interesting history going back to 1956 and to questions about trees in the late nineteenth century. The problem of determining the probability that a random graph has a given connected component structure was apparently first posed in [6]. The question of determining the number of connected graphs on n-nodes with j edges was posed and solved in terms of generating functions in [19], [11], [1] and [2] and perhaps elsewhere between 1959 and 1971.

In analyzing the computational complexity of algorithms for determining the connected component structure of a graph, it is useful to know the mean-time to obtaining that structure as well as the sojourn time in various structures sequentially obtained. To our knowledge, these problems have not been discussed in the literature. In the special case of a connected graph with (n-1) edges we ask for the number of trees on n-vertices. This question has its own history starting with [4], [5], [3], see especially [15]. Some results of these authors are obtained herein as special cases of our results.

In [21] the suggestion was offered that a number of probability questions are more "naturally" posed and solved by mapping sample spaces into semilattices (semilattice variables) rather than to the real line (random variables). In some examples, the distribution function of the mapping (called a generating function) can be determined and inverted using the Möbius-Rota inversion therorem (see e.g. [20]) to obtain the density function (distribution function in [21]) of the mapping. Indeed this is a useful method for obtaining the probability that a "random graph" with j-edges has a given connected component structure. The approach yields a complicated but closed form formula from which the generating function is easily obtained. Furthermore, we show that the process of adding edges at random is a Markov process and hence, that theory can be used to study the generation of random graphs. It is easy to obtain the probability transition function for the Markov process, which turns out to be non-stationary. None of the formulae obtained are useful for computing so we close the paper with a pair of coupled equations which can be used for computing the combinatorial functions of interest. This is carried out in Sections III-IV. Notations and basic definitions are introduced in Section II.

The question how "connected" is a graph has been subjected to numerous definitions. The commonly accepted definition is the size of a minimal cut-set. There are many reasons why this is not a particularly good definition. Two obvious reasons are:

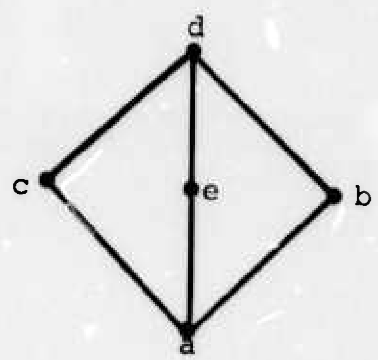
1) It can not be applied to disconnected graphs, although obviously some disconnected graphs are more connected than others.

2) Many connected graphs have the same size minimal cut-set although the complexity of their internal structure is quite different.

A definition of "connectivity" which would differentiate between connected graphs by measuring their internal structure and also differentiates between disconnected graphs would seem to have many obvious merits and applications. We propose and discuss such a measure in Sections VI-XI.

II. THE LATTICE OF PARTITIONS

A lattice is a partially ordered set where each pair of elements has a greatest lower bound and least upper bound (g.l.b. and l.u.b.) We denote the statement that a is less than or equal to b in the partial ordering by $a \le b$. In a lattice as on the real line we have three types of intervals, or segments. $[x,y]$, $[x,y)$, $(x,y]$ which are respectively the sets of a's for which $x \le a \le y$; $x \le a < y$; $x < a \le y$. It is easy to see that $[x,y]$ is a sublattice of the original lattice when endowed with the same ordering. An element b is said to cover an element a when, $a<x<b$ is satisfied by no x, or equivalently when the interval $[a,b]$ contains exactly two elements. A Hasse diagram is a pictorial depiction of a lattice where the elements are represented as points and a line is drawn from b to a when b covers a, e.g.



An element of a lattice is minimal (maximal) if it covers no element (is covered by no elements). A unique minimal (maximal) element is called a zero or least (unit or greatest). When the zero or unit exist they are denoted by 0 an I respectively. An atom is an element which covers a minimal element while a dual atom is an element covered by a maximal element.

Let V_n be a finite set with n-labelled objects. A partition of V_n is a family of disjoint subsets of V_n, say $P_1, P_2, \ldots, P_k$, called parts or blocks of π, whose union is V_n. A natural ordering on the set Π_n of partitions of V_n is given by $\sigma \leq \pi$ if every block or part of σ is a subset of some part or block of π. It is not difficult to verify that the set Π_n of partitions of V_n is partially ordered with the ordering just defined. In fact, Π_n is a lattice with that ordering. Furthermore, the lattice Π_n has a zero 0, or least element given by the partition with n-singleton parts. The lattice Π_n also has a unit I greatest element given by the partition containing one block V_n itself. The following result about covers in Π_n is obvious.

Lemma 2A : An element $\sigma \in \Pi_n$ covers $\pi \in \Pi_n$ if and only if σ is obtained by combining any two parts or blocks of π.

The lattice Π_n is endowed with a very useful rank or numerical ordering as will be evident from an examination of the Hasse diagram for small n, or from Lemma 2A. A partially ordered set is said to satisfy the Jordan-Dedekind chain condition if:

1. It has a zero and unit.
2. All totally ordered subsets having a maximal number of elements have the same number of elements.

A totally ordered subset of a partially ordered set is called a chain, $a > b > \ldots > w$, e.g.. A chain is maximal if it cannot be enlarged. When a lattice satisfies the Jordan-Dedekind chain condition one can introduce a rank function R (p) on the lattice. The functions r(p) is defined as the length of a maximal chain in the segment [o,p] minus one. The rank of zero is zero, the rank of an atom is one, etc. If the rank of I is n-1 then the rank of a dual atom is n-2. It is easy to verify that Π_n satisfies the Jordan-Dedekind chain condition and indeed the rank of a partition $\pi \in \Pi_n$ is n minus the number of blocks or parts of π.

Another fundamental descriptive combinatorial notation for partitions is the <u>type</u> or <u>class</u> of π. A partition π is of type or class $(k_1, k_2, \ldots, k_n)$ when k_i is the number of blocks or parts with i elements. Obviously, if $\pi \varepsilon \Pi_n$ then $\sum_{i=1}^{n} k_i = n - r(\pi)$ where $r(')$ is the rank function, and $\sum_{i=1}^{n} ik_i = n$. The following structure lemma is fundamental for forming recurrences on Π_n.

<u>Lemma 2B</u>: Let Π_n be the lattice of partitions of a set with n-elements. If $\pi \varepsilon \Pi_n$ is of rank k then the segment or interval $[\pi, I]$ is isomorphic to Π_{n-k}. If π is of class of type $(k_1, k_2, \ldots, k_n)$ then the interval $[0, \pi]$ is isomorphic to the direct product of k_i lattices isomorphic to Π_1, k_2 lattices isomorphic to $\Pi_2, \ldots, k_n$ lattices isomorphic to Π_n.

Among other applications of lemma 2B it was shown in [20] that it can be used to compute the Mobius function of Π_n. The Mobius function was shown by Rota to be an important invariant of lattices and hence a distinguished member of the incidence algebra over a partially ordered set.

If P is a partially ordered set, the <u>incidence algebra</u> of P denoted by I(P) is the algebra of real valued functions $f: PxP \to R; \ni x \not\leq y \Longrightarrow f(x,y) = 0$; addition and scalar multiplication are defined as usual and the product (convolution) is given by $e = f*g$, $e, g, f \varepsilon I(p)$

$$e(x,y) = \sum_{x \leq z \leq y} f(x,z) g(z,y).$$

It is easy to verify that I(p) is an algebra. The convolution becomes essentially matrix multiplication when P is finite. Among some distinguished elements of the incidence algebra are the functions:

zeta function : $\zeta(x,y) = 1$ if $x \leq y$ and 0 otherwise
delta function: $\delta(x,y) = 1$ if $x = y$ and 0 otherwise
incidence function: $n(x,y) := \zeta(x,y) - \delta(x,y)$.

Since the delta function is easily verified to be an identify in I(P) certain elements of I(P) will be invertible in I(P) with respect to δ.

The next and last set of definitions can be found in [21] Let S be an arbitrary finite set and W ($\cdot$) be a weight function on S. (S can be thought of as a sample space and W ($\cdot$) a probability measure). Let Π be a lattice. A mapping $X:S\to\Pi$ is called a semilattice variable analogous to probability theory and random variables. Let A (Π) be the algebra of singlevalued functions $f:\Pi\to R$ with addition and scalar multiplication as usual, and $e=f*g$, $f,g,e\varepsilon A(\Pi)$ given by

$$e(x) = \sum_{a\wedge b=x} f(a)g(b),$$

where $a\wedge b$ is the g.l.b. of the pair $\{a,b\}$. With this notation, if the weight function is a probability density on S then there is a function $f\varepsilon\ A(\Pi)$ which represents the density function of a semilattice variable X given by

$$f(\pi) = \sum_{t:\ x(t)=\pi} W(t)$$

The generating function F (probability distribution function) of f is given by

$$F(\sigma) = \sum_{\pi\leq\sigma} f(\pi) = \sum_{\pi\varepsilon[o,\sigma]} f(\pi);$$

and plays the role of a probability distribution function. The density function and distribution functions can be computed from each other by the Möbius Rota inversion formula. This approach is useful in "random graph" theory. For other examples see [21].

III. STRUCTURE OF RANDOM GRAPHS

Let the set $V_n = \{v_1, v_2, \ldots, v_n\}$ denote a set of n-nodes or vertices of a graph. Let $\mathcal{E}_n = \{(e_1, e_2, \ldots, e_{\binom{n}{2}}) \mid e_i$ is a subset of size two from $V_n\}$ be the set of sequences or all permutations of the edges of the complete graph on V_n. We define the lattice stochastic process $X_1, X_2, \ldots, X_{\binom{n}{2}}$; $X_j : \mathcal{E}_n \to \Pi_n$ by $X_j\ (e) = X_j\ (e_1, e_2, \ldots, e_{\binom{n}{2}}) =$ the partition of V_n determined by the connected components of the graph $(V_n, \{e_1, e_2, \ldots, e_j\})$; $j=1, 2, \ldots, \binom{n}{2}$. Thus for example $X_1(e)$ is always a partition of rank 1 and type $(n-2,1,0,0,\ldots,0)$; $X_2(e)$ is always a partition of rank 2 and $(n-2)$ parts but can be of two possible types $(n-3,0,1,0,\ldots,0)$ or $(n-4,2,0,0,\ldots,0)$. In fact for each $e \in \mathcal{E}_n$, $X_1(e) \le X_2(e) \le \ldots \le X_{\binom{n}{2}}(e) = I$ so that the process is monotonic nondecreasing in the natural ordering on Π_n. Furthermore, each pair $(X_j\ (\cdot),\ X_{j+1}\ (\cdot))$ has $X_{j+1}(e) = X_j(e)$ or is a cover of $X_j(e)$ so that no wild jumps take place.

We assume that each $e \in \mathcal{E}_n$ has the same probability $\frac{1}{\binom{n}{2}!}$ so that we can define the density function;

$$f_j(\pi) = P\{X_j = \pi\} = \frac{\text{no. of } e \in \mathcal{E}_n \text{ for which } X_j(e) = \pi}{\binom{n}{2}!}$$

and the one step transition function.

$$P_j(\sigma, \pi) = P\{X_j = \pi \mid X_{j-1} = \sigma\}.$$

Thus if $X_j(e) = \pi$ then $X_{j+1}(e) = \pi$ if the $(j+1)^{th}$ edge of e is a subset of some part of π, otherwise, $X_{j+1}(e)$ is a cover of π obtained by combining the two parts of π which contain the points of e_{j+1}. The next theorem should be fairly obvious.

Theorem 3A: The semilattice process $X_1, X_2, \ldots$ is Markovian. The state I is absorbing and the process is absorbed with probability one.

Proof: The state I is absorbing since eventually the graph becomes connected. To see that the process is Markovian we observe that the probability of moving to any state depends only on the number of edges in the graph and the type of the partition. The number of ways of staying in the same state is computed by selecting edges not already selected from within the blocks of π, this is independent of the history. The number of ways of moving to any given cover depends only on the sizes of the parts to be combined which is also independent of the history of the process.

It is easy to determine the transition functions which are incidently, members of the incidence algebra I (Π_n), and invertible.

Theorem 3B: The transition function is given by,

$$P_j(\sigma,\pi) = \begin{cases} \dfrac{h(\sigma,\pi)}{\binom{n}{2}-(j-1)} & \text{when } \pi \text{ covers } \sigma. \\[2ex] 1 - \sum\limits_{\sigma \lessdot \pi} \dfrac{h(\sigma,n)}{\left(\binom{n}{2} - (j-1)\right)} & \text{when } \sigma=\pi, f_j-1(\pi)\neq 0 \\[2ex] 1 \text{ When } \pi=\sigma, f_{j-1}(\pi)=0 & \\[1ex] 0 \text{ Otherwise; } j=1,2,\ldots,\binom{n}{2}; & \end{cases}$$

where $h(\sigma,\pi)$ is the number of edges connecting the pair of blocks of σ which is one part of π when π covers σ and zero otherwise.

Proof: The formulae for $p_j(\pi,\pi)$ are obvious since at each stage each unselected edge is equally likely to be chosen.

Theoretically theorem 3A and 3B can be used to answer all questions about the process since we know the initial conditions and the transition functions. We will use these observations in the next section to compute various combinatorial functions. We can use the Mobius-Rota Inversion formula to obtain a closed form solution for the probability of being in any given state at any given time since the distribution function is easy to compute. First, however, it is interesting to obtain a new formula in the form of the transition functions for the number of trees on n-vertices.

Theorem 3C: (Chapman-Kolmogoroff Equations). The following formulae hold;

$$f_j(\pi) = \sum_{0 \leq \pi_1 \leq \pi_2 \leq \cdots \leq \pi_{j-1} \leq \pi} p_1(0,\pi_1)p_2(\pi_1,\pi_2)\cdots p_j(\pi_{j-1},\pi)$$

$$\pi \in \Pi_n;\ j=1,2,\ldots,\binom{n}{2}.$$

Corollary: If T_n is the number of trees on n vertices then

$$T_n = \binom{\binom{n}{2}}{n-1} \sum_{0 \leq \pi_1 \leq \pi_2 \leq \cdots \leq \pi_n \leq I} p_1(o,\pi_1)\ p_2\ (\pi_1,\pi_2)\cdots p_{n-1}(\pi_{n-2},I).$$

Let $F_j(\pi)$ be the distribution function of X_j; $j=1,2\ldots\binom{n}{2}$ i.e. $F_j(\pi) = \sum_{\sigma \leq \pi} f_j(\sigma)$ or the probability that the partition of connected components is a refinement of the partition π after j-edges are introduced.

<u>Theorem 3D</u>: If π is of type $(k_1, k_2, \ldots, k_n)$ then

$$F_j(\pi) = \frac{\binom{\sum_{i=1}^{n} k_i \binom{i}{2}}{j}}{\binom{\binom{n}{2}}{j}} \quad ; \quad j=1,2,\ldots,\binom{n}{2}.$$

<u>Proof</u>: The j-edges are to be chosen from within the parts of π.

Since $F_j(\cdot)$ is known $f_j(\cdot)$ can be computed by the Mobius-Rota inversion formula.

<u>Corollary 1</u>: The probability that the graph is connected after j edges have been added is;

$$f_j(I) = \frac{n!}{\binom{\binom{n}{2}}{j}} \sum_{k=1}^{n} (-1)^{k-1} (k-1)! \sum_{\substack{(k_1,\ldots,k_n) \\ \Sigma k_i = k \\ \Sigma i k_i = n}} \frac{\binom{\Sigma k_i \binom{i}{2}}{j}}{1^{k_1} (2!)^{k_2} \ldots (n!)^{k_n} k_1! k_2! \ldots k_n!}$$

Proof: By Theorem 2A

$$f_j(I) = \sum_{k=1}^{n} \sum_{r(\sigma)=n-k} F_j(\sigma)\mu(\sigma,I)$$

$$= \sum_{k=1}^{n} \sum_{\substack{(k_1,\ldots k_n) \\ \Sigma k_i=k \\ \Sigma ik_i=n}} B(n;k_1,\ldots,k_n)F_j(\sigma)\mu(\sigma,I)$$

where σ is of type $(k_1,\ldots,k_n)$ and $B(n;k_1,k_2,\ldots k_n)$ is the number of partitions of type $(k_1,\ldots k_n)$ with k parts. It is easy to see that,

$$B(n;k_1,\ldots,k_n) = \frac{1}{k_1!k_2!\ldots k_n!} \frac{n!}{1^{k_1}(2!)^{k_2}\ldots(n!)^{k_n}}$$

So that

$$f_j(I) = \sum_{k=1}^{n} \sum_{\substack{(k_1,\ldots,k_n) \\ \Sigma k_i=k \\ \Sigma ik_i=n}} \frac{n!\binom{\sum_{i=1}^{n} k_i\binom{i}{2}}{j}\mu(\sigma,I)}{k_1!k_2!\ldots k_n!1^{k_1}(2!)^{k_2}\ldots(n!)^{k_n}\binom{\binom{n}{2}}{j}}$$

Since $\mu(\sigma,I)=(-1)^{k-1}(k-1)!$ when $[\sigma,I]$ is isomorphic to Π_k the result is proved.

IV. DESCRIPTIVE COMBINATORIAL QUANTITIES

We will now derive formulae and generating functions for the interrelated combinatorial quantities of interest in the evaluation of algorithms. We will count sequences rather than deal with additional symbols for the associated probabilities. Obviously, each of the formulae are easily converted to probabilities and moments can be computed. We will treat n, the number of vertices as a parameter since it will be helpful in the next section where we give a relatively simple way of computing mean-time to a given component structure. The basic functions are:

A. $h_n(\pi;j)$ = the number of edges that can be added to a graph with j-edges on n-vertices with component structure π so that the component structure stays at π.

B. $p_j(\sigma,\pi)$ = the number of ways of adding an edge to a graph G with (j-1) edges and component structure σ to change the component structure to π.

C. $f_n(\pi,j)$ = the number of sequences of j-edges on n-vertices which determine the connected component structure π.

D. $C_n(\pi,j)$ = the number of sequences of j-edges on n-vertices which "enter" π for the <u>first</u> time with the addition of the j^{th} edge.

E. $S_n(\pi,j,k)$ = the number of sequences of k-edges which enter π for the first time on the j^{th} edge and leaves π on the k^{th} edge. This quantity is called the conditional sojourn time in π.

If a partition σ covers a partition π then the pair of numbers (i,j), where i and j are the sizes of the parts of π which are combined to form σ, is called the cover type pair of (π,σ).

Lemma 4A: The quantity $p_j(\sigma,\pi)$ is given by

F.
$$p_j(\sigma,\pi) = \begin{cases} 0 \text{ if } \sigma \neq \pi \text{ and } f_n(\sigma,j-1) = 0, \\ 1 \text{ if } \sigma = \pi \text{ and } f_n(\sigma,j-1) = 0, \\ h_n(\pi,j-1) \text{ if } \sigma = \pi \text{ and } f_n(\pi,j-1) > 0, \\ i \cdot k \text{ if } (i,k) \text{ is the pair cover type of } (\sigma,\pi), \\ 0 \text{ otherwise.} \end{cases}$$

Proof: The proof is immediate, actually, this is a repeat of Theorem 3B, restricted for completness.

Lemma 4B: The quantity $h_n(\pi,j)$ is given by

G.
$$\left(\sum_{i=2}^{n} k_i \binom{i}{2} \right) - j$$

when π is of type $(k_1,k_2,\ldots,k_n)$.

<u>Proof</u>: Obvious.

<u>Lemma 4C</u>: The quantity $C_n(\pi,j)$ is given by,

H. $C_n(\pi,j) = \sum_{\sigma \leq \pi} f_n(\sigma,j-1)\, p_j(\sigma,\pi)$.

<u>Proof</u>: Immediate.

<u>Lemma 4D</u>: The quantity $S_n(\pi;j,k)$ is given by.

I. $$S_n(\pi;j,k) = \sum_{\sigma>\pi} C_n(\pi,j) \left[\prod_{\eta=1}^{k-j-1} n_n(\pi,j+\eta) \right] p_k(\pi,\sigma).$$

<u>Corollary</u>: The number of sequences which spend k-units of time in π denoted by $s_n(\pi,k)$ is given by,

J. $$s_n(\pi,k) = \sum_{j=1}^{\binom{n}{2}-k} S_n(\pi;j,j+k) \; ; \; k=0,1,2,\ldots,$$

The proofs of all of the above lemmas follow from the Markovian property of the process $X_1, X_2, \ldots, X_{\binom{n}{2}}$. All of the quantities are computable from the above equations and the "closed form" equation for $f_n(\pi,j)$ as given in the previous section. However, the calculations are cumbersome and can be done more directly, as we will do in the next section. We close this section by obtaining the generating function for $f_n(I,j)$ and hence, for all the quantities given above.

<u>Theorem 4A</u>: If $F_n(y) = \sum_{j=1} \frac{f_n(I,j)}{j!} y^j$, then,

K. $$F_n(y) = -n! \sum_{k=1}^{n} \sum_{\substack{(k_1,\ldots,k_n) \\ \Sigma k_i = k \\ \Sigma i k_i = n}} \frac{(k-1)!}{k_1! k_2! \ldots k_n!} \prod_{i=1}^{n} \left(\frac{-B_i(y)}{i} \right)^{k_i}$$

Where $B_i(y) = (1+y)^{\binom{i}{2}}$, (This formula is related to Faa Di Bruno's formula for derivatives and the classic Bell polynomials.)

<u>Proof</u>: From the result of Section 3,

$$\frac{f_n(I,j)}{j!} = n! \sum_{k=1}^{n} \sum_{\substack{(k_1\ldots,k_n) \\ \Sigma k_i = k \\ \Sigma i k_i = n}} \frac{(-1)^{k-1}(k-1)!}{k_1! k_2! \ldots, k_n!} \frac{\binom{\Sigma k_i \binom{i}{2}}{j}}{1^{k_1} (2!)^{k_2} \ldots (n!)^{k_n}}$$

Thus

$$F_n(y) = n! \sum_{k=1}^{n} (k-1)!\, (-1)!^{k-1} \sum_{\substack{(k_1,\ldots,k_n \\ \Sigma k_i = k \\ \Sigma i k_i = n}} \sum_{j \geq 1} \frac{\binom{\Sigma k i \binom{i}{2}}{j} y^j}{1^{k_1} (2!)^{k_2} \ldots (n!)^{k_n}}$$

$$= n! \sum_{k=1}^{n} \sum_{\substack{(k_1,\ldots k_n) \\ \Sigma k_i = k \\ \Sigma i k_i = n}} \left[\frac{(1+y)^{\Sigma k_i \binom{i}{2}}}{1^{k_1} (2!)^{k_2} \ldots (n!)^{k_n}} \right] (k-1)!\, (-1)^{k-1}$$

and the result follows by expanding

$$(1+y)^{k_1\binom{1}{2} + k_2\binom{2}{2} + \ldots + k_n\binom{n}{2}} = (1+y)^{\binom{1}{2}k_1} (1+y)^{\binom{2}{2}k_2} \ldots (1+y)^{\binom{n}{2}k_n}$$

V. CALCULATIONS:

We can determine pairs of equations from which we can recursively compute $f_n\ (I,j)$ from $c_n\ (I,j)$ and conversely. These equations can be derived from the Markovian property or from the fundamental isomorphic decomposition of Π_n.

Theorem 5A: (Basic Recurrence)

If π is of type $(k_1,k_2,\ldots,k_n)$ then

$$f_n\ (\pi,j) = \sum M(w_{2,1},w_{2,2},\ldots,w_{2,k_2};w_{3,1},w_{3,2},\ldots,w_{3,k_3};\ldots;w_{n,1},w_{n,2},\ldots,w_{n,k_n})$$

$$\prod_{j=1}^{k_2} f_2\ (I,w_{2,j}) \prod_{\tau=1}^{k_3} f_3\ (I;w_{3,\tau}) \ldots \prod_{\nu=1}^{k_n} f_n\ (I,w_{n,\nu}).$$

where $M\ (j,w_{2,1},w_{2,2},\ldots,w_{2,k_2},w_{3,1}w_{3,2},\ldots,3,k_3;\ldots;$

$w_{n,1},w_{n,2},\ldots w_{n,k_n})$ is the multinomial coefficient and the sum is over all partitions of the integer j into $\sum_{\tau=2}^{n} k_\tau$ parts.

Proof: If π is a partition of type $(k_1, k_2, \ldots, k_n)$ then if a sequence of j edges produces π the sequence must contain places for the $k_{\nu,\tau}$ edges which come from the τ^{th} part of those parts which have ν-vertices. The set of subscripts $\{1,2,\ldots,j\}$ can be partitioned in $M(j;\ldots)$ such parts. The edges in the part with $w_{\nu,\tau}$ edges can then be arranged in $f_\nu(I, w_{\nu,\tau})$ ways. The product formulation follows from the basic isomorphism theorem.

Corollary 1: If π is a dual atom of type (r,n-r) (i.e. one part with r-elements the other with (n-r) elements) then

$$f_n(\pi, j) = \sum_{w=1}^{j} \binom{j}{w} f_r(I,w) \cdot f_{n-r}(I, j-w)$$

when $n>2$, $r>1$, $(n-r)>1$;

$f_n(\pi, j) = f_{n-1}(I;j)$ when $r=1$ or $n-1=r$;

$f_2(I,1) = 1$, $f_2(I,j) = 0$ $j>1$

Corollary 2: [5], [3], [16].

If T(n) is the number of trees on n-vertices then,

$$T(n) = \frac{1}{2(n-1)} \sum_{i=1}^{n-1} \binom{n}{i} T(i) \cdot (n-i) \cdot i \cdot (n-i)$$

Proof: If we set j=n-2 in Corollary 1 then,

$$f_n(\pi, n-2) = \sum_{w=1}^{n-2} \binom{n-2}{w} f_r(I,w)\, f_{n-r}(I, n-2-w).$$

This expression has one non-zero term so,

$$f_n\ (\pi,n-2) = \binom{n-2}{r-1} f_r(I,r-1) \cdot f_{n-r}(I,\ r-2-(r-1)).$$

Since obviously $f_r(I,r-1) = T(r) \cdot (r-1)!$ we obtain

$$f_n(\pi,n-2) = (n-2)!T(r)\cdot T(n-r).$$

Since π has two parts we get a tree on n-vertices by a single edge connecting those parts. Summing over all such partitions then yields all trees on n-vertices.

<u>Theorem 5B:</u> The quantity $C_n\ (I,j)$ is given by

$$C_n\ (I,j) = \sum_{\substack{\pi \text{ is} \\ \text{a dual atom}}} f_n\ (\pi,j-1)\ h\ (\pi,I)\quad n>2$$

$C_1\ (I,j) = 0\ \forall j;$

$C_2\ (I,1) = 1\quad C_2\ (I,j) = 0\quad j>1.$

<u>Proof</u>: The proof follows by the Markovian property. If we are to enter I for the first time on the j^{th} then after (j-1) we must be in a dual atom.

<u>Corollary</u>:

$$C_n(I,j) = \frac{1}{2} \sum_{r=2}^{n-2} \binom{n}{r} f_n\ (\pi_{r,n-r};j-1)\cdot r\cdot(n-r)$$

$$+\ n\ (n-1)\cdot f_n\ (\pi_{1,n-1};j-1)$$

(The notation $\pi_{i,j}$ means π is a dual atom of type (i,j).)

Proof: Substitution of the formula of Corollary 1 to Theorem 5A into Theorem 5B.

Theorem 5C: The quantity C_n (I,j) can be obtained from f_{n-1}, $f_{n-2}, \ldots$ using

$$C_n(I,j) = \frac{1}{2}\sum_{r=2}^{n-2} \binom{n}{r} \sum_{w=1}^{j-1} \binom{j-1}{w} \cdot f_r\,(I,w) \cdot f_{n-r}\,(I,j-1-w) \cdot r \cdot (n-r)$$

$$+ n\,(n-1) \cdot f_{n-1}\,(I;j-1).$$

Proof: Combine Theorem 5A and 5B

Theorem 5D: The quantity f_n (I,j) can be computed from C_n (I,j) by the equation

$$f_n\,(I,j) = \sum_{k=n-1}^{j} C_n\,(I,k) \left(\binom{n}{2}-k\right) \left(\binom{n}{2}-(k+1)\right) \ldots \left(\binom{n}{2}-j\right).$$

Proof: This follows from the Markovian property. If we are to be in state I after j steps then we arrived for the first time on the k^{th} step and stayed in I.

Theorems 5C and 5D provide a coupled pair of equations to compute C_n from f_{n-1} and then f_n from $C_n, \ldots$ and so forth. Some sample calculations are, f_2 (I,1) = 1, and f_2 (I,j) = 0 $j>1$

We compute C_3 from f_2;

C_3 (I,1) = 0

C_3 (I,2) = $\frac{1}{2}$ (empty sum) + $3 \cdot 2 \cdot f_2$ (I;1) = 6.

C_3 (I,j) = 0 $j>2$.

Now we compute f_3 from C_3

$$f_3(I,2) = \sum_{k=2}^{2} C_3(I,k) = C_3(I,2) = 6$$

$$f_3(I,3) = \sum_{k=2}^{3} C_3(I,k)\cdot(3-k) = C_3(I,2)\cdot 1 = 6$$

$f_3(I,j) = 0,\ j>3.$

We can compute C_4 from f_3

$$C_4(I,3) = \frac{1}{2}\sum_{r=2}^{2}\binom{4}{2}\sum_{w=1}^{2}\binom{2}{w} f_r(I;w)\cdot f_{n-r}(I,j-1-w)\cdot r\,(4-r)$$

$$+\ 4\cdot 3\cdot f_3(I;2)$$

$$= \frac{6}{2}\sum_{w=1}^{2}\binom{2}{w} f_2(I;w)\cdot f_2(I;2-w)\cdot 2\cdot 2 + 4\cdot 3\cdot 6$$

$$= 12\ [2\cdot f_2(I;1)\cdot f_2(I,1) + 1\cdot f_2(I;2)\cdot f_2(I,0)] + 72$$

$$= 12\ [2\cdot 1+0] + 72 = 96$$

$$C_4(I,4) = \frac{1}{2}\sum_{r=2}^{2}\binom{4}{2}\sum_{w=1}^{3}\binom{3}{w} f_r(I,w)\cdot f_{n-r}(I;3-w)\cdot r\cdot(4-r)$$

$$+4\cdot 3\cdot f_3(I,3)$$

$$= 12\ [3\cdot 0 + 3\cdot 0] + 72 = 72$$

$C_4(I,j) = 0;\ j\geq 5.$

Similarly, we can compute f_4 from C_4 and obtain $f_4(I,3) = 96$, $f_4(I,4) = 360$, $f_4(I,5) = 720$ and $f_4(I;j) = 0,\ j>5$. The coupled equations themselve can be used to obtain the exponential generating functions for the sequences f_n, and C_n. These generating functions turn out to be exponential convolutions and have been recently studied [17].

It is possible to compute f_j (I) from the formula of corollary 1 but the calculation is lengthy. An example follows:

Example (n=4). For n=4 $T_n=16$, hence the probability of a tree should be $\frac{16}{20} = \frac{4}{5}$ when j=3.

Parts (k)	Class or Type	$\sum_{i=2}^{4} k_i \binom{i}{2}$	$\binom{\Sigma k_i \binom{i}{2}}{j}$
1	(0,0,0,1)	6	20
2	(1,0,1,0)	3	1
2	(0,2,0,0)	2	0
3	(2,1,0,0)	1	0
4	(4,0,0,0)	0	0

$$f_3(I) = \frac{24}{20}\left[\frac{20}{4!}\right] = \frac{24}{20}\left[\frac{96}{(24)\ (6)}\right] = \frac{16}{20} = \frac{4}{5}$$

VI. A NEW MEASURE OF CONNECTIVITY

Let G be a connected graph with n nodes and m edges. Let S(G) be the set of the m! permutations of edges, viewed as a set of m!-tuples. For each sequence s S(G) define $C_c(s)$ as the index of the first edge in s for which the graph with the first $C_c(s)$ edges is connected. The number $m-C_c(s)$ then measure "how long" the sequence s has been a connected graph. Intuitively, if many of the sequences in S(G) have large $m-C_c(s)$ then the graph G is more connected. In particular we take the average or first moment of numbers $\{m-C_c(s)\}$ as the definition of the connectivity of G. We can also take higher moments which would lead to more precise measures of connectivity.

Definition: The connectivity or "mean connectivity" of a connected graph G, denoted by C(G), is given by:

$$C(G) = \frac{1}{m!} \sum_{s \varepsilon S(G)} (m-C_c(s)) = m - \frac{1}{m!} \sum_{s \varepsilon S(G)} C_c(s)$$

Alternatively, if $C_G(k)$ is the number of sequences in S(G) for which $C_c(s) = k$ then,

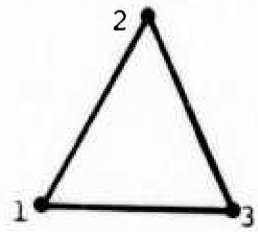
$$C(G) = m - \frac{1}{m!} \sum_{k=n-1}^{m} k\, C_G(k)$$

Example: If G is a tree on n nodes then m=n-1 and $C_c(s) \equiv n-1$ for all $s \varepsilon S(G)$, so that

$$C(G) = (n-1) - \frac{1}{(n-1)!}(n-1)\ (n-1)! = (n-1) - (n-1) = 0.$$

Thus, all these have connectivity zero.

Example: Consider a triangle:



There are six edge sequences

$s_1 = \{1,2\}, \{1,3\}, \{2,3\}$, $C(s_1) = 2$

$s_2 = \{1,2\}, \{2,3\}, \{1,3\}$, $C(s_2) = 2$

$s_3 = \{1,3\}, \{2,3\}, \{1,2\}$, $C(s_3) = 2$

$s_4 = \{1,3\}, \{1,2\}, \{2,3\}$, $C(s_4) = 2$

$s_5 = \{2,3\}, \{1,3\}, \{1,2\}$, $C(s_5) = 2$

$s_6 = \{2,3\}, \{1,2\}, \{1,3\}$, $C(s_6) = 2$

$$C(G) = 3 - \frac{1}{6} \sum_{s \in S(G)} 2 = 3 - \frac{1}{6} [12] = 3 - 2 = 1$$

So that the connectivity of the complete graph K_3 is $C(K_3) = 1$. The connectivity of K_2 is zero since K_2 is a tree.

Problem 1: Determine the sequence $C(K_2)$, $C(K_3)$,...; or its generating function; i.e.,

$$C_k(z) = \sum_{i=2}^{\infty} \frac{C(K_i)}{i!} z^i$$

We now turn our attention to disconnected graphs. Let G be a disconnected graph and let S(G) be the set of all permutations of edges <u>not</u> in G. For each $s\varepsilon S(G)$ let $C_d(s)$ be the index of the first edge in the sequence s for which G would become connected if all the preceeding edges were added to G.

<u>Definition</u>: If G is a disconnected graph with m-edges we define its connectivity,

$$C(G) = -\sum_{s\varepsilon S(G)} \frac{C_d(s)}{((\binom{n}{2})-m))!} = -\frac{1}{((\binom{n}{2})-m!} \sum_{s\varepsilon S(G)} C_d(s)$$

<u>Example</u>: The empty graph Ω_2 on two nodes, has m=0 so C(s)=1. Therefore,

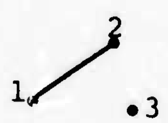
$$C(\Omega_2) = -\frac{1}{(\binom{2}{2})-0)!} = -1$$

<u>Example</u>: We compute the connectivity of Ω_3. There are six sequence in S(G) and $C(s)\equiv 2$.
Therefore,

$$C(\Omega_3) = -\frac{1}{3} \sum_{s\varepsilon S(G)} 2 = -\frac{12}{6} = -2$$

The computation of the sequence $\{C(\Omega_n)\}_{n=2}^{\infty}$ or its generating function is carried out in the next sections.

<u>Example</u>: We compute the connectivity of the graph G depicted below:



The graph G is disconnected. The edges not in G are {1,3}, {2,3}, there are two permutations, and C(s) = 1 for both so that,

$$C(G) = -\sum_{s\epsilon S(G)} \frac{1}{2!} = -\frac{1}{2}(1+1) = -1$$

Many questions, problems and conjectures are already apparent, some of these are:

Problem 2: There seems to be a duality in terms of connectivity between a graph and its complement. Can the connectivity of a graph be obtained in terms of its complement? The answer is yes and the appropriate result is given in Section VII.

Problem 3: Is the connectivity of a graph related to the reliability of its network? The answer is yes and this is discussed at an elementary level in Section VIII.

Problem 4: If we are given a graph G, then all sequences $s\epsilon S(G)$ have "connectivity weights" C(s). What are the minimal and maximal values of C(s) over all $s\epsilon S(G)$. In statistical terms we are asking for the range of C(s) over S(G); i.e. $\max_{s\epsilon S(G)} C(s) - \min_{s\epsilon S(G)} C(s)$,

Along these same lines given a class G of graphs what are the minimal and maximal values of the connectivities of the graphs in G? Characterize the classes of graphs which achieve these values. This problem is discussed in Section IX.

Problem 5: Can the connectivity of a graph be determined by the connectivity of various bipartite graphs. What is the connectivity of a bipartite graph consisting of two connected components? This question is discussed in Section X.

Problem 6: There are many descriptive combinatorial functions associated with a graph, minimal cutset size, girth, diameter, number of spanning trees, cliques, etc. How are these related to connectivity? The definitions of these quantities are given in Section XI.

Problem 7: There are many combinatorial questions associated with the sequence of graphs determined by a sequence of edges in S(G). Many of these questions concern the number and type of the connected component structure associated with the sequence of graphs associated with each s∈S(G). Some of the questions are formulated in Section XI.

Problem 8: There should obviously be relationships between connectivity of a graph and its chromatic polynomial or other such coloring properties. In Section XI we give Whitney's derivation of the chromatic polynomial of a graph. It seems that the Whitney approach gives a tie in with connectivity since it deals with colorings of subgraphs and that seems to be a way of approaching connectivity.

Problem 9: When is the connectivity of a graph greater than the connectivity of every subgraph?

Problem 10: If two graphs have the same number of edges but different size cutsets, does it follow that their connectivities have the same properties?

Problem 11: How many "moments" are required to uniquely determine a graph or isomorphic classes of graphs?

Problem 12: How do we construct "highly" connected graphs, or build on existing graphs to increase connectivity at low cost?

VII. A GRAPH AND ITS COMPLEMENT

We investigate Problem 2 and derive a relationship between a graph and its complement.

Definition: It G is a graph then its complement G' is the graph on the nodes of G whose edge set is the set of all edges not in G.

Theorem 7A: If G is a disconnected graph with m edges on n nodes then

$$C(G') - C(G) = \binom{n}{2} - m + \frac{1}{\binom{n}{2}-m} \sum_{s\in S(G)} (C_d(s)-C_c(s))$$

Proof: If G is disconnected then G' is connected and G' has $(\binom{n}{2}-m)$ edges. Since S(G) and S(G') are identical as sets of $(\binom{n}{2}-m)$-tuples of edges we have by direct calculation;

$$C(G') - C(G) = (\binom{n}{2}-m) - \frac{1}{(\binom{n}{2}-m)!} \sum_{s\in S(G')} C_c(s) + \sum_{s\in S(G)} \frac{C_d(s)}{(\binom{n}{2}-m)!}$$

$$= (\binom{n}{2}-m) + \frac{1}{(\binom{n}{2}-m)!} [\sum_{s\in S(G)} C_d(s) - \sum_{s\in S(G')} C_c(s).]$$

$$= (\binom{n}{2}-m) + \frac{1}{(\binom{n}{2}-m)!} \sum_{s\in S(G)} [C_d(s) - C_c(s)].$$

Corollary 1: For the complete and empty graphs;

$$C(K_n) + C(\Omega_n) = \binom{n}{2}.$$

Proof: Set $G = \Omega_n$ in Theorem 2.1 and note that $C_d(s) = C_c(s) \ \forall \ s \epsilon S(\Omega_n)$.

Corollary 2: For the generating functions $C_k(z)$ and $C_\Omega(z)$ of $\{C(K_m)\}$ and $\{C(\Omega_n)\}$, respectively we have

$$C_k(z) + C_\Omega(z) = \frac{z^2 e^z}{2}$$

Proof: By direct calculation!

VIII. APPLICATION TO RELIABILITY AND OTHER AREAS

The analysis of sequence sets such as S(G) arises in many areas other than in a "pure" study of graph connectivity. In set-merging algorithms we have a partition of a set of n points which we may think of as nodes of a graph. Unordered pairs of points are presented sequentially which we may think of as edges of the graph. If the two points of an edges are in different parts of the partition then the two parts are merged (their set theoretic union is formed). Otherwise, the same analysis is carried out with the next edge. The obvious questions concern the statistical moments of the number of edges need to connect the graph (merge all parts into one part) or to reach some other "state". Many of these questions can be restated in terms of the statistics of S(G).

Another area in which the set S(G) arises is in the use of "random edges" to generate a spanning tree for a graph or network with n-nodes. Edges are randomly generated and added to a graph (beginning perhaps with the empty graph). The question arises in the analysis of such a process as to how long it will take before the resulting "random graph" is connected. The answer to this question if we start with an empty graph on n nodes is $C(\Omega_n)$. It should also be clear that this question is the same as the set merging question.

A large area of potential application of the notion of connectivity is the study of the reliability of networks. Let G be a graph and $R_p(G)$ be the probability that the graph G is connected if p is the probability that is branch (independently) is "on" or "working" or "up". Clearly if G has m edges.

$$R_p(G) = \sum_{k=n-1}^{m} A_k p^k (1-p)^{m-k}$$

where A_k is the number of connected subgraphs of G with n nodes and k edges. Alternatively

$$R_p(G) = 1 - \sum_{i=w}^{m} B_i q^i p^{m-i},$$

where b_i is the number of disconnected subgraph of G with n nodes and (m-i) edges. W is the size of a minimal cutset of G.

A simple revision for $R_p(G)$ relates the reliability of graph or networks to subgraphs. This might be helpful in relating reliability to connectivity and to coloring. Let e be a given edge in G. so that:

$$\begin{aligned} R_p(G) &= P\{G \text{ is connected} \mid e \text{ is open}\}\, P\{e \text{ is open}\} \\ &\quad + P\{G \text{ is connected} \mid e \text{ is closed}\}\, P\{e\text{is closed}\} \\ &= P\{G \text{ is connected} \mid e \text{ is open}\} + q\, R_p(g') \end{aligned}$$

where G' is the subgraph of G obtained by removal of e. Let G'' be the graph obtained by identifying the nodes incident to e. Thus

$$R_p(G) = pR_p(G'') + aR_p(G')$$

where

$$R_p(G'') > R_p(G) > R_p(G')$$

The same type of formula can be obtained by taking a subset of edges of G which is not a cutset. This would help tie in with coloring problems.

The Moore-Shannon definition of reliability is similar although it asks for the probability p_{ij} that a given pair of nodes {i,j} is connected by some path in G. Clearly:

$$P_{ij} = \sum_{k=1}^{m} A_k p^k (1-p)^{m-k}$$

Where A_k is the number of distinct k edge subgraphs of G which contain a path between nodes i and j, and m is the number of edges. Alternatively, if B_k is the number of k edge subset of edges which when removed leaves nodes i and j disconnected then,

$$1-P_{ij} = \sum_{k=1}^{m} B_k (1-p)^k p^{m-k}$$

It is easy to relate $R_p(G)$ to C(G) and it should be easy to relate P_{ij} to C(G). Let G be a graph on n nodes with m edges. Let $f_G(k)$ be the number of sequences in S(G) for which the first k edges on the n nodes form a connected graph.

<u>Theorem 8A</u>: Let $E_w(p)$ be the expected value of X! when X is a binomial random variable b(w;p). Let $C_k(p) = \frac{C_G(k)p^k}{k!}$, then

$$R_p(G) = \sum_{k=0}^{m} E_{m-k}(p)\ C_k(p)$$

<u>Proof</u>: By definition:

$$R_p(G) = \sum_{k=n-1}^{m} A_k p^k (1-p)^{m-k}$$

Now $A_k = \frac{f_G(k)}{k!} = \frac{1}{k} \sum_{w=n-1}^{k} C_G(w)\ (m-w)_{(k-w)}$

where $(t)_{(n)} = t\ (t-1)\ldots(t-k+1)$ is the falling factorial.

It follows by substitution that

$$R_p(G) = \sum_{k=n-1}^{m} \frac{1}{k!} \sum_{w=n-1}^{k} C_G(w)\,(m-w)_{(k-w)}\, p^k (1-p)^{m-k}$$

Rearranging terms and collecting by $C_G(w)$ yields.

$$R_p(G) = \sum_{k=n-1}^{m} \frac{C_G(k)p^k}{k!} \sum_{j=0}^{m-k} (m-k)_{(j)}\ p^j (1-p)^{m-j-k}$$

which proves the theorem!

Problem 13: Devise an algorithm for computing $C_G(k)$. Perhaps a Markovian type algorithm as used in [22] for the complete graph might work.

Problem 14: Study the behavior of the sequence $C_k(p)$ in terms of the connectivity of connected graphs. In fact if $C'_k(p)$ is the derivative of $C_k(p)$ with respect to p, then obviously the connectivity of G is,

$$C(G) = [m-\frac{1}{m} \sum_{k=n-1}^{m} k! \, C'_k(p)]_{p=1}.$$

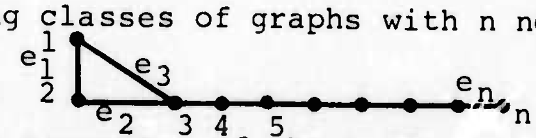
Problem 15: Suppose two graphs or networks with the same number of nodes and edges have different reliability functions. To what extent can the differences be explained by the "moments" of connectivity?

Problem 16: Relate statistical connectivity to the Moore-Shannon definition of reliability.

IX. GRAPHS WITH MINIMAL AND MAXIMAL CONNECTIVITY

In Problem 4, we asked about graphs with minimal and maximal connectivity, we shall examine the case of connected graphs with n nodes and n edges. The connected graphs with (n-1) edges all have connectivity zero. What is the minimal connectivity of a graph with n edges? What is the maximal value? Obviously if G has n nodes and n edges and is connected $0 < C(G) \leq 1$. More generally if G has m edges $0 < C(G) \leq m-(n-1)$. There are no connected graphs with connectivity zero which have more than (n-1) edges.

The following classes of graphs with n nodes and n edges have connectivity $\frac{3}{n+1}$.



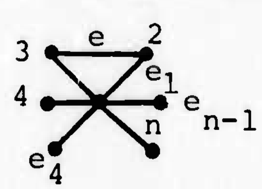
There are 3(n-1)! sequences of the n edges whose associated graph becomes connected for the first time on the $(n-1)^{st}$ edge, and (n-3)(n-1)! sequences which become connected graphs for the first time with the n^{th} edge. Therefore

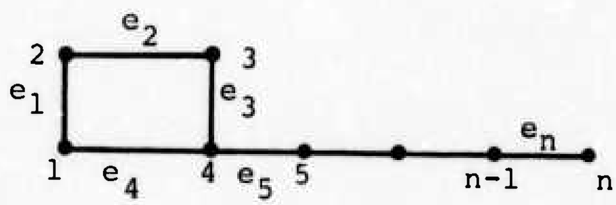
$$C(G) = m - \frac{1}{m} \sum_{k=n-1}^{n} kC_G(k) = n - \frac{1}{n!} \sum_{k=n-1}^{n} k\, C_G(k)$$

$$= n - \frac{1}{n!}\,[(n-1)(n-1)!\cdot 3 + n(n-3)(n-1)!]$$

$$= n - \frac{3(n-1)}{n} - (n-3) = 3 - \frac{3(n-1)}{n} = \frac{3}{n}$$

Another graph with the same connectivity is,





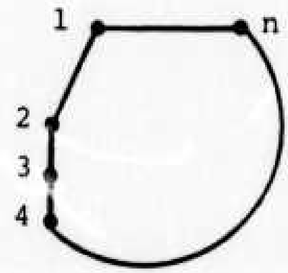
It should be easy to prove the following conjectures.

<u>Conjecture 1</u>: Among the class of connected graphs with n nodes and n edges the minimum value of the connectivity is 3/n and is achieved by those graphs which contain a triangle.

More generally.

<u>Conjecture 2</u>: If a connected graph on n nodes with n edges contains a k-gon then its connectivity is k/n.

Conjecture two applies to the maximum value of the connectivity. Indeed the cycle graph C_n.



has connectivity one since the ommission of any edges does not destroy connectivity. Many other questions are now apparent!

<u>Problem 17</u>: Generalize conjectures 1 and 2 for graphs with more than n edges.

<u>Problem 18</u>: Are there parallel results for the connectivity of disconnected graphs.

<u>Problem 19</u>: In what sense are the reliabilities of the above networks minimal and maximal.

X. CONNECTIVITY OF BIPARTITE GRAPHS

Let G be a graph on n nodes with two connected components C_1 and C_2 with m_1, m_2 edges and V_1, V_2 nodes, respectively.

There are $\binom{n}{2}$ possible edges of which $(m_1 + m_2)$ are already in the graph G so that we form sequences from the $\binom{n}{2} - (m_1 + m_2)$ edges not in the graph G. Let X be the random variable "time to connectivity" on S(G). There are $(V_1 \cdot V_2)$ good edges and $\binom{n}{2} - (m_1 + m_2) - (V_1 \cdot V_2)$ "bad" edges from which to select at random (form sequences) and add to G one at a time. Let $P_k = P\{X=k\}$, so that

$$P_1 = P\{X=1\} = \frac{V_1 \cdot V_2}{\binom{n}{2} - (m_1+m_2)}$$

$$P_2 = P\{X=2\} = (1-P_1)\left(\frac{V_1 \cdot V_2}{\binom{n}{2} - (m_1+m_2)-1}\right)$$

$$\vdots$$

$$P_n = \left[\prod_{i=1}^{n-1} (1-P_i)\right] \left(\frac{V_1 \cdot V_2}{\binom{n}{2} - (m_1+m_2)-(n-1)}\right)$$

Let $G(z) = \sum_{k=1}^{\infty} P_k z^k$ be the generating function for the sequence $\{P_n\}$.

Theorem 10A: The function G(z) satisfies the differential equations;

$$(1-z)\, G'(z) + [(1-z)\,(W+1) = (wz)]\, G(z) + \frac{w}{W}(z-w-1) = 0,$$

where $w = V_1 V_2$ and $W = \binom{n}{2} - (m_1+m_2)$.

Proof: It is easy to see that P_n satisfies the recurrence relation

$$P_n = P_{n-1} \left(\frac{W-w-(n-2)}{W-(n-1)}\right) \text{ for } n \geq 2 \text{ with}$$

$$P_1 = \frac{w}{W}.$$

The theorem follows by direct application of the generating function and a little algebra.

The connectivity of the bipartite graph is given by G'(1). Perhaps a more detailed analysis of the differential equation will yield some more information.

It is possible to obtain an upper bound for the connectivity of a bipartite graph which we conjecture to be exact in the limit.

Let Y be the time to connectivity of the bipartite graph when the edges are chosen with replacement. Obviously $E[Y] \geq E[X]$. The time to connectivity for the replacement process is easy to compute since the distribution of y is geometric. In fact;

$$E[y] = \sum_{k=1}^{\infty} k \left(1-\frac{w}{W}\right)^{k-1} \left(\frac{w}{W}\right) = \frac{W}{w}.$$

so that we have proved.

Theorem 10B: Let G be a bipartite graph on n nodes with connected components C_1 and C_2 with m_1, m_2 edges and V_1, V_2 nodes respectively, then,

$$C(G) \geq - \frac{\binom{n}{2} - (m_1+m_2)}{V_1 \cdot V_2}.$$

Problem 20: Obtain an asymptotic estimate in Theorem 10B by suitably restricting the ranges of m_1, m_2, V_1, V_2.

Problem 21: If we interpret connectivity of a graph in reliability terms as edges going out one at a time, selected at random, what is a reasonable definition for the reliability of a bipartite graph? If terms of Moore-Shannon it might be the average time for two nodes (one in each component) to communicate. Interpret and calculate "reliability of a bipartite graph" and relate it to connectivity.

XI. DESCRIPTIVE COMBINATORIAL QUANTITIES IN A GRAPH

In the literature of graph theory, there are many combinatorial type quantities which measure internal structure of a graph. Each of the quantities should be related to connectivity. We will not spell out the specific questions since they should be obvious!

Definition: If a graph G has p connected components, n nodes and m edges then the rank of G denoted by $\rho(G) = n-p$ and its cyclomatic number denoted by $\nu(G) = m-\rho(G)$.

Theorem 11A: (Known). Let G be a graph and G' be a graph formed from G by adding a new edge between node i and j (arbitrary) then,

$\rho(G') = \rho(G)$ and $\nu(G') = \nu(G)$ if i and j are in the same component, while $\rho(G') = \rho(G)+1$, and $\nu(G') = \nu(G)$ if i and j are in different components.

Corollary: $\rho(G) \geq 0$ and $\nu(G) \geq 0$.

Definition: The chromatic number of a graph is the smallest number of colors needed to color the graph so that no adjacent vertices have the same color.

Theorem 11B: (König) A graph is bi-chromatic if and only if it contains no cycles of odd length.

It is not difficult to verify that the zeta function is invertible. The inverse of the zeta function is called the Mobius function and is denoted by $\mu(.,.)$. These observations of the Mobius Inversion Theorem.

Theorem 2A: (Mobius-Rota Inversion Theorem)

If $f: P \to R$ is a real valued function on a finite partially ordered set P and p so that $f(x) = 0$ when $x \not< p$ and,

$$g(x) = \sum_{y \leq x} f(y) \qquad \text{then we have,}$$

$$f(x) = \sum_{y \leq x} g(y)\mu(y,x),$$

where $\mu(.,.)$ is the Mobius function of the partially ordered set P.

Some useful results for computing the Mobius function are given in [20]: In Particular the next formula for the Mobius function of Π_n.

Theorem 2B: The segment [x,y] is said to be of class $(C_1, C_2, \ldots, C_n)$ when the lattice [x,y] is isomorphic to the direct product of C_1 lattices isomorphic to Π_1, C_2 lattices isomorphic to $\Pi_2, \ldots, C_n$ lattices isomorphic to Π_n. If [x,y] is of class $(c_1, c_2, \ldots, c_n)$ then

$$\mu(x,y) = (-1)^{c_1+c_2+\ldots c_n - n}(2!)^{c_3}(3!)^{c_4}\ldots((n-1)!)^{c_n}$$

Corollary: $\mu(o,1) = (-1)^{n-1}((n-1)!$

REFERENCES

1. Austin, T. L., R. E. Fagen, W. F. Penney, and J. Riordan, "The Number of Components in Random Linear Graphs," Annual Math. Stat. 30, 1959, pp. 747-754.

2. Bender, E. A., and J. Goldman, "On The Enumerative Uses of Generating Functions," Indiana J. Math., Vol. 20, No. 8, 1971, pp. 753-765.

3. Bol, G., "Uber eine Kombinatorische Frage," Abh. Math. Sem., Hansische University, 12, 1938, pp. 242-245.

4. Cayley, A., "A Theorem On Trees," Quart J. Math., 23, 1889, pp. 376-378, Collected Papers Cambridge, 13, 1897, pp. 26-28.

5. Dziobek, O., "Eine formel der Substitutions Theorie," Sitzungsberichte der Berlin Math. Gesellschoft, 17, 1917, pp. 64-67.

6. Erdös, P. and A Renyi, "On Random Graphs I," Publ. Math Debereen, Vol. 6, 1959, pp. 290-297.

7. Erdös, P. and A. Renyi, "On the Evolution of Random Graphs," Publ. of The Math. Inst. of the Hungarian Acad. of Sci. 5, 1960, pp. 17-61.

8. Erdös, P., and J. Spencer, Probabilistic Methods in Combinatorics, Academic Press, 1973.

REFERENCES (continued)

9. Fisher, M. J., "Efficiency of Equivalence Algorithms," Complexity of Computer Computations, Eds. Miller, R. E. and J. W. Thatcher, Plenum Press, N. Y. 1972.

10. Ford, G. W., and G. E. Uhlenbeck, "Combinatorial Problems in The Theory of Graphs I," Proc. Nat'l. Acad. U. S. A., Vol. 42, 1956, pp. 122-128.

11. Gilbert, E. N., "Enumeration of Labelled Graphs," Can. J. Math., Vol. 8, 1957, pp. 405-411.

12. Gilbert, E. N., "Random Graphs," Ann. Math. Stat., 30, 1959, pp. 1141-1144.

13. Hopcroft, J. E. and J. D. Ullman, "Set Merging Algorithms," SIAM J. On Comp. , 2, No. 4, 1973.

14. Kershenbaum, A. and R. Van Slyke, "Computing Minimum Spanning Trees Efficiently," Proc. of the ACM 1972 Conf., Boston, August, 1972.

15. Moon, J. W., Counting Labelled Trees, Can. Math. Conf., Math. Mono., No. 1, 1970.

16. Mullin, R. C., and R. G. Stanton, "A Combinatorial Property of Spanning Forests in Connected Graphs," J. Comb. Th., 3, 1967, pp. 236-243.

REFERENCES (continued)

17. Mullin, R. C., and G. C. Rota, "On the Foundation of Combinatorial Theory III; Theory of Binomial Enumeration," Graph Theory and Its Application, Ed. B. Harris, Academic Press, 1970.

18. Renyi, A., "On Connected Graphs I," Publ. Math. Inst. Hungarian Acad. Sci., 4, 1959, pp. 385-388.

19. Riddell, R. J., Jr., and G. E. Uhlenbeck, "On the Theory of The Virial Development of the Equation of State of Monoatomic Gases," J. Chem. Phys. 21, 1953, pp. 2056-2064.

20. Rota, G. C., "On the Foundations of Combinatorial Theory I," Z. Wahrecheinlick-keitstheorie, Band 2, Hef-4, 1964, pp. 340-368.

21. Tainiter, M., "Generating Functions on Idemputent Semigroups II; Semilattice Variables and Independence," J. Comb. Th., Vol. 13, No. 3, 1972.

22. Tarjan, R. E., "On the Efficiency of a Good But Not Linear Set Union Algorithm," Preprint, Comp. Sci. Dept., Cornell University, 1973.

23. Van Slyke, R. and H. Frank, "Network Reliability Analysis-I," Networks, 1971.

SIMULATION OF PACKET COMMUNICATION NETWORKS

I. INTRODUCTION

In developing a large scale communication network, one encounters many problems which cannot be formulated or solved analytically. Consequently, one resorts to simulation. In this chapter, we outline the structure of a simulation program for packet communication networks; specify the problems which can be resolved by a simulator; give the description of the simulator developed for the packet radio network; present results obtained by the simulator; and discuss the future development of the simulator for the packet radio network.

In general, there are various degrees of simulation depending on the amount of knowledge (or assumed knowledge) about the system operation, and the objectives of the simulation, i.e., the problems to be resolved. We particularly distinguish in this chapter between a simulation for design and a simulation for development.

A simulator for design (e.g., in a design loop) is developed when the operation of the system is completely specified, and the objective of the simulator is to simulate specific parts of the system which cannot be analytically modeled or whose solution is computationally infeasible. The efficiency of such a simulator is of major importance; consequently, one attempts to avoid simulation [8] wherever possible, in the simulation program.

The objectives of a simulation for development are much broader. In this case, one has a set of theoretical (untested) hypotheses which state one or several possible ways for system operation.

These include routing algorithms, protocols, etc. The objectives are to test (verify) the hypotheses, to complete the specification of portions of the system, to compare alternative modes of system operation, to identify system bottlenecks, and finally, to deduce measures of system efficiency by obtaining estimates of the major parameters.

II. OBJECTIVES OF SIMULATOR

We outline specific problems for which a simulation approach solution is most appropriate.

A. Routing Algorithms

The object is to compare the efficiency of existing and newly developed routing algorithms in terms of throughput and delay on the one hand, and storage and processing requirements of the algorithms at the switching nodes on the other hand. The objective of this comparison is to suggest a small set (two or three) algorithms for implementation and further testing in an experimental system. We note that it is not mandatory that a single routing algorithm be used in a network. For example, in the broadcast network that we discuss later, a simple routing algorithm is used to load the switching nodes (repeaters) with a more sophisticated algorithm. Furtheremore, one may examine the possibility of using an alternative routing algorithm under overload conditions or in different hierarchies of the network, if a hierarchical network is investigated.

B. Protocols

There may be many protocols in a communication network considered, depending on the type of communicating devices and on the type of application. The following protocols are examples: terminal-terminal, terminal-switching node, terminal-host computer, switching node-switching node, switching node-host computer, host computer-host computer. Some of the above may contain more than one protocol depending on the application, and others may be needed in a hierarchical network. The objective of the simulation is to test the efficiency of these protocols in a dynamic environment, to change them when necessary, and to complete details which may be missing.

C. Identify System Bottlenecks

One of the advantages of the simulator is that it enables one to observe and trace the detailed flow of packets in the network. For example, a specific packet may be traced, such as an information packet, an acknowledgement packet, or various priority packets, from the origination node to the destination node. This allows investigation of questions such as, whether the bottleneck is at the switching nodes or due to the limited capacity of the channel, given a proposed configuration, and communication protocols. It may also suggest improvements in communication protocols.

D. Flow Control

The simulator is the only tool which allows testing and improvement of theoretically developed flow control algorithms.

E. Software Transfer

A simulator can be coded so that subroutines or sections of code are identified with specific software programs of the switching nodes. This will reduce the effort of software development by transfering or coding according to the programs in the simulator. Furthermore, it will allow testing of sections of the software of communication devices by comparing these with the corresponding sections in a simulated device.

F. Trade-Offs

Among the trade-offs which can be investigated are: the trade-off between the storage requirements at switching nodes and the channel capacity of links, given the topology and the offered traffic rates; trade-offs between average delay, maximum delay, and delay as a function of priority and link and switching node capacities.

G. Stand-by Network

The simulator can be useful beyond the stage of development. It can be used to study particular problems which may be encountered in the network, once it becomes operational, and to test suggestions for improvement.

III. GENERAL STRUCTURE OF SIMULATOR

It is important to separate between data structure and management functions on the one hand, and the communication and device functions on the other hand. There are several advantages in doing so; firstly, one communication device does not have access to information available in other devices; and secondly, it is easier to identify and distinguish the communications part of the program for the purpose of modification or software transfer. We have developed efficient data structures which can be used for simulation programs of communication systems. The proposed data structures are described in Section V, and includes the fourteen (14) subroutines of Section VIII, subsection A.

It is useful to have one main subroutine for each type of communication device. For example, one subroutine for all switching nodes of the same type. The differences between the devices (e.g. switching node) can be recorded in a state vector associated with each device. The state vector will include information such as, the switching nodes to which this device has channels and the data rates of these channels, the state of occupancy of the storage buffers of the node, the routing algorithm that this node is currently using, and others. In addition there will be buffers associated with each device in which the content of specific packets (e.g. packet type, priority) will be stored.

There are distinguisable functions which may be used by more than one type of device (e.g. a modem), these functions can be coded in separate subroutines.

A. Performance Measures

There should be an extensive measurement program to enable the resolution of the problems outlined in the objectives. The measurements are divided into network performance measurements, performance of communication devices, and trace information.

B. Network Measurements

- Throughputs

We distinguish between throughput of packets and throughput of information. Packets which successfully travel from origin to destination contribute to the packet throughput measure. Packets which are also acknowledged contribute to the information throughput measure. The distinction comes from the fact that when protocols are not efficient or when delays are very large, the origination node may reissue another copy of a previously transmitted packet. When the network delivers both packets, then the two packets contribute to the packet throughput measure but in terms of information transfered only one packet was delivered. Network throughputs are measured as a function of time. This enables one to determine whether the network can maintain a steady state throughput when offered a given traffic rate, to estimate the time needed to obtain steady state, and to observe the behavior of the network under either minor or major perturbations.

- The Average Number of Links That A Packet Traverses

This measure when compared with the average number of links of the input (offered) rate for a given topology, reflects on the amount of alternate routing. It may also enable one to detect looping in the network.

- <u>Delays</u>

Many delays should be measured in the network, we outline some of these:

1. Delay to negotiate protocol between a terminal and a switching node.
2. Delay to negotiate protocol and reserve storage at destination switching node.
3. Delay of an information packet of a given priority from origination to destination node.
4. Delay for receiving an end-to-end acknowledgment back at the origination node.
5. Delay for the delivery of a maximum size message and receiving an acknowledgment.
6. Delays of special priority packets.

C. <u>Performance of Communication Devices</u>

- <u>Utilization of Devices</u>

- Fraction of time that the device is transmitting or receiving.

- Number of Packets stored in device as a function of time.

- Number of Packets successfully switched and number of packets discarded due to buffer overflow or other reasons.

D. <u>Trace Information</u>

This information includes the listing of significant communication events. The listing contains a unique identifier of the packet, its type and priority, its origin and destination, and the time of the event. These are needed to allow a detailed observation of flow, to trace of specific packets, and to follow specific interactions between source and destination.

Remark: Other measurements may be needed to evaluate the efficiency of flow control algorithms.

IV. PACKET RADIO NETWORK SIMULATOR

The main features which distinguish the broadcast network from a point-to-point (PTP) packet switching network such as the ARPANET are: (i) devices in the network transmit packets by using a random access scheme, and (ii) devices broadcast so that signals can be received by several devices simultaneously.

When using a random access scheme, there is a possibility that several packets are simultaneously received by a receiver due to independent transmissions of several devices; in the event, none of these packets are correctly received, and the corresponding devices must retransmit their packets. This implies that, unlike PTP network, the probability of error due to overlapping packets is much higher than the probability of error due to other causes such as Gausian or impulse noise. Furthermore, the probability of error varies widely depending on the amount of traffic on the channel. The broadcast nature of the network implies that there is correlation between the probabilities of successful transmission in different parts of the network.

The communication system simulated contains three types of devices, terminals, repeaters, and stations. The station is considered as an interface communication device of the broadcast network to a higher level network or to a computer installation. Terminals are considered as traffic sources and traffic sinks, they transmit packets to the station and receive response packets from the station; they may be mobile. The basic function of the repeater is to extend the effective range of the terminals and the station. The network simulated contains repeaters and stations which are placed at fixed locations chosen at random. Terminals originate at random times and are placed in random locations on the plane. A terminal is considered to depart from the system once it completes its communication. A more detailed description of the system simulated can be found in [4], and [5].

The program was coded in FORTRAN. It includes a total of 33 subroutines, approximately 3,000 statements. The compilation time on a CDC 6600 takes approximately 20 CP sec. , and the running time for meaningful results takes 100-300 CP sec. The storage requirement of the program is $245{,}000_8$ 60 bit words.

The simulator has already been used for improving communication protocols, and to answer questions related to the trade-offs between device range and device interference, and the trade-offs between a single and a dual data rate system.

V. DATA STRUCTURES OF THE SIMULATOR

The global information for the Packet Radio Simulation Program is contained in five (5) data structures:

1) Event Structure;
2) Active Message Structure;
3) Active Packet Structure;
4) Repeater-Station Structure;
5) Data Collection Tables.

The meaning, configuration, and elements of these structures will be explained in the next five sub-sections. In the last sub-section, the use of these structures in the context of the entire program will be indicated.

A. Event Structure

The simulation program is event - driven. That is, periodically the Event Structure is consulted to determine the time of occurrence of the next event. The Event Structure also contains information telling the program what the event is. Examples of events are arrivals of messages to terminals, transmission of packets, arrival of packets at receivers, and arrival of messages to the stations. As events are executed, they are deleted from the structure; and, periodically, newly-generated events are added to the structure. Since there are a large number of events; many more than the number of exogenous message arrivals, for example, the process of efficiently determining the next event is of vital importance. To this end the event times are maintained in a "heap." Corresponding to each event "i" is its time "t_i," the device subroutine "d_i" to which it refers (e.g. station, terminal, repeater), the index "w_i" of the device in question (i.e. which repeater, which station, etc.), a number representing the point at which the routine is entered, and, finally, the packet number of the packet in question. In addition, there is the heap index vector itself. "h_j" points to the event which occupies the j^{th} position in the heap. A heap is a structure in which $t_{h_j} \leq \text{Max}\ \{t_{h_{2j}}, t_{h_{2j+1}}\}$. At all times t_{h_1} is the smallest t_i over all the events. This structure allows for quite rapid selection of the minimum t_i while using a minimal amount of storage, [2], [1], [3]. In order to efficiently eliminate old events and to reuse the space created by the elimination, a garbage stack of unused locations is maintained.

B. Active Message Structure

The external or exogoneous traffic which flows into the Packet Radio Network consists of messages which arrive at terminals or stations and represent information that users wish to send to a location using the net. Messages are to be distinguished from packets which carry the message information internally in the net. In general, there may be several packets in the network carrying copies of the same message or of parts of that message. Messages are added to the Active Message Structure when the event corresponding to their generation occurs. It stays on the list until the last packet containing the message is dealt with. Associated with each message "i" is its arrival time "t_i;" the current number of packets representing the message "n_i" (When "n_i" is reduced to zero (0), the message is removed from the structure.); its length "l_i;" "x_i" and "y_i", the co-ordinates of the terminal or station originating the message; and other pieces of data, such as the repeater with which the terminal communicates, the total number of packets, and arrival time of message, which may be needed for output statistics. In order to eliminate old messages and add new ones efficiently, the messages are kept in a doubly-linked list structure. There is a list kept for messages and one for unused message spaces. This information is kept in two vectors, "f" and "b". Thus, "f_i" represents whatever follows the space in question ("i"), be it message or empty space; while "b_i" represents whatever precedes the space. The unit being followed or preceded may be an empty space or may contain a message.

C. Active Packet Structure

The active packets are kept in a list. Associated with the i^{th} packet is a pointer to its corresponding message and several words which state the variable part of the packet label for use by the routing algorithms. A garbage stack similar to the one used for events keeps track of vacant spaces in the Packet Structure.

D. Repeater-Station Structure

The Repeater-Station Structure is basically a list of the repeaters and stations, their locations, their possible neighbors (repeaters and stations within range of their transmitters and receivers), and their state. Associated with the i^{th} station or repeater are "x_i" and "y_i", the co-ordinates of the device; and a state vector which, among others, specifies items such as whether the repeater or station is busy, free, turned off, or failed; and whether the device is a station or a repeater. In addition, there is a list of repeaters and stations adjacent to the i^{th} repeater or station.

E. Data Collection Tables

Sufficient statistics for the evaluation of the system performance measures (see Section VI subsection D) are kept in the data collection tables.

F. An Outline of the Use of Data Structures

The global simulation structure is based on events of packets arriving at devices: stations, terminals, and repeaters. Figure 1 schematically illustrates the program flow. It is quite simplified, one simplification being especially important and needing emphasis. The events of retransmissions and acknowledgements resulting from a packet's arriving at a repeater, station, or terminal are not necessarily generated immediately, but may depend on the arrival of packets at the device subsequent to the packet which gives rise to the new events.

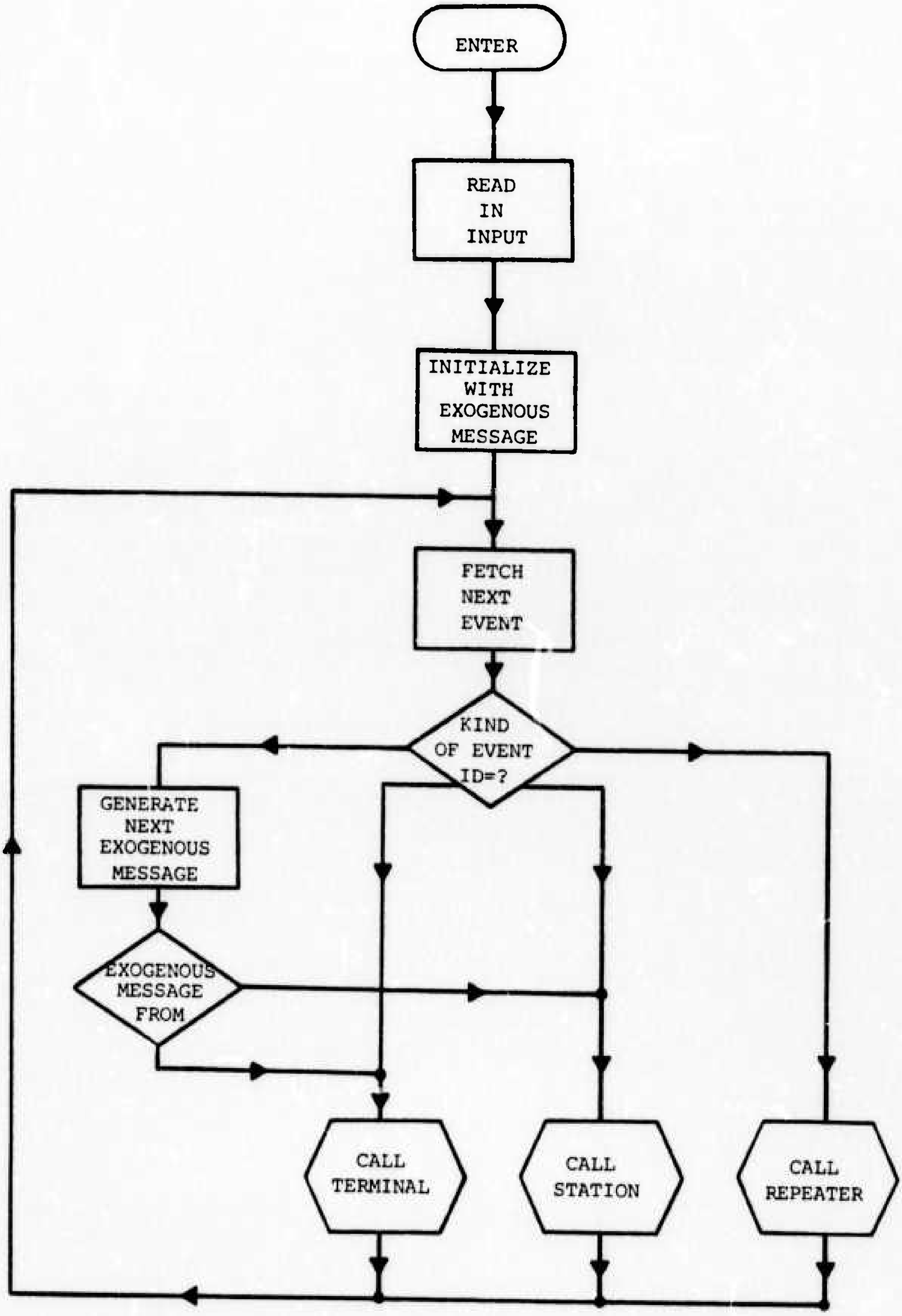


FIGURE 1

GENERAL PROGRAM FLOW

VI. THE COMMUNICATION SYSTEM SIMULATED

A. General System Description

• Channel and Access Mode

The communication channel is shared for transmission in both directions, to the station or from the station to terminals. The channel access mode is the Non-Persistent Carrier Sense Multiple Access (CSMA) [7]. That is, when a packet is ready for transmission, the device senses the channel and transmits the entire packet if the channel is idle. If the channel is busy, the device reschedules the packet for some future random time at which it senses the channel again, and the procedure repeated.

• Capture

A zero capture system is simulated. That is, whenever the reception of more than one packet overlap in time, none of the packets is correctly received.

• Packet Types Simulated

IP - Information Packet

ETE - End-to-End Acknowledgement; Short Packet (Assumed to be 10% of the length of an information packet).

SP - Search Packet, transmitted by Terminal or Repeater to all devices and aimed to identify a specific receiver (short packet).

RSP - Response to a Search Packet, transmitted by Repeater or Station which received a SP and is available for handling packets. This packet contains the label of the transmitting device and is addressed to all devices.

B. Routing Algorithms

Three routing algorithms which are implemented in the simulator are briefly described in the following paragraphs. A detailed description of the routing schemes is given in [5].

- Hierarchical Labeling:

The hierarchical labeling routing scheme enables point-to-point routing between devices along an "efficient path". It is obtained by assigning to every repeater a label, which forms, functionally, a hierarchical structure. The label assigned contains the following information:

1) A specific address of the repeater for routing purposes.

2) The minimum number of hops to the nearest station.

3) The specific address of all repeaters on the shortest path to the station, and the address of the repeater to which a packet has to be transmitted when destined to the station.

In the hierarchical labeling algorithm an information packet (IP) is addressed to one device. If it is received by a device to which it is not addressed, then the receiving device is closer to the destination than the device to which the packet was addressed. If the preassigned path is temporarily blocked, the packet may depart from it. It then uses the most efficient path from its new location. The departure from the preassigned path is obtained by a search procedure.

The Response to Search Packet (RSP) is transmitted by repeaters after a random waiting time. This time randomization is essential in a no-capture system. Otherwise, if more than one receiver wish to respond to the SP, the transmissions will overlap at the searching device. The station, on the other hand, transmits the RSP immediately after receiving the SP (Note that if we assume zero processing time then the channel is idle at this time since otherwise the SP would not have been correctly received). The above enables searching devices within range of an idle station to communicate directly with the station.

A repeater makes one attempt to transmit a RSP and if the channel is busy it discards it, rather than store the RSP for future transmissions. This allows control of the level of terminal blocking by specifying the number of transmissions of the SP by a terminal. Thus, when the system is conjested in the geographical neighborhood of the terminal it will not be able to "enter" the system. This feature also makes repeaters more available for handling information packets.

- Directed Routing (One Level Labels)

Directed routing is a simplified version of the hierarchical labeling algorithm in which the only information preserved is the direction TO or FROM station. Repeaters are assigned labels that indicate the hierarchy level, or the number of hops to the nearest station. When a device transmits a packet, the packet is addressed to all repeaters (stations) that are closer to the destination than the transmitting device. Many devices can receive the same packet and it may arrive at more than one station. The acknowledgement schemes, the station-station protocol, and the station-terminal protocol must then resolve this problem.

- <u>Flooding Algorithm (Plus Repeater Memory)</u>

In this algorithm, there is no directionality of transmission. A packet is addressed to all devices that can hear it. To control the problems of cycling and looping, repeaters are assigned storage for unique identifiers of packets that they recently repeated. When a packet is received by a repeater, it compares its identifier with those stored and discards the packet if a match occurs. A maximum handover number (MHN) in the packet will prevent it from being propagated for very long distances. This feature is also used in the other routing algorithms.

C. Acknowledgement Schemes

The acknowledgement scheme has particular significance in this system because of the broadcast feature and the limited capability of the repeater for processing and storage. The following acknowledgements are used:

- End-to-end acknowledgement (ETE Ack) between station and terminal to ensure message integrity. The frequency and precise meaning of the Ack depends on the particular protocol used, and is part of the protocol.

- A hop-by-hop passive echo acknowledgement (HBH Echo Ack) along the path. When device i transmits a packet, it waits a sufficient time to allow devices that receive the packet to repeat it. When any of these repeats the packet and the packet is received by device i, it considers it as an Ack.

In a point-to-point network, such as the ARPANET, the channels are fixed so that when node j receives a packet on channel k it knows the device, say i, which has transmitted the packet to it, and thus can transmit a specific HBH Ack to device i. In the Packet Radio System, however, this information is not available. Therefore, the HBH Ack must be <u>independent of the path that the packet travels on</u>. The HBH Echo Ack test used included the following:

1) identification of the packet

2) tests that the MHN of the Echo received is <u>smaller</u> than the MHN of the packet stored. This insures that the packet has advanced along the path, rather than being a retransmission from devices that had the packet previously.

The HBH Echo Ack has several advantages over a specific HBH Ack:

1) it simplifies the repeater (hardware and software) so that it need not constract and manage acknowledgement packets.

2) it reduces the traffic overhead of transmitting specific acknowledgements. This is most significant in a broadcast network.

3) it enables acknowledgement of several devices at a time; in particular, all devices which store the packet with a MHN larger than that received are acknowledged.

4) It enables shortening the transmission path, as described below.

Since the RSP's by repeaters are randomized in time, a terminal frequently does not identify the repeater nearest to the station within range of the terminal. In fact, if two repeaters are labelled on the same path to the station and both are within range of the terminal, there is a higher probability that the terminal will identify the repeater farther away from the station since, on the average, the latter handles less traffic. Suppose a single data rate channel is used throughout the transmission path between terminal and station, the terminal identifies a packet transmitted to it by its specific terminal ID, and the station can recognize any packet destined for it. Then a communication path as shown in Figure 2 may be established. In Figure 2, the terminal is within an effective range to R4, R5, and R6; and the station within an effective range to R1, and R2. The terminal shown identified R6 as the repeater to which it transmits. The path from termianl to station will usually be : T→R6→R5→R4→R3→R2→S; and from the station to the terminal S→R2→R3→R4→T. The end devices, terminal and station, transmit the Echo Ack immediately after receiving the packet, and <u>transmit it with MHN=0</u>; thus they acknowledge all devices which still store the packet. In particular, when R4 transmits a packet towards the terminal it is addressed to R5, however, the terminal may receive this packet and acknowledge both R4 and R5 simultaneously. Similarly, when R2 transmits towards the station it addresses the packets to R1, when the packet is received by the station it acknowledges both R1 and R2 simultaneously.

D. Performance Measures

• Throughput

Considering the set of stations and the set of terminals as the end devices, the system throughput (in packets) is defined as the rate of information packets (IP's) that originated at stations and arrived at terminals, plus the rate of IP's that originated at terminals and arrived at stations.

• Delays

The following delays are measured:

1) Terminal delay to identify specific repeater.

2) Terminal delay to establish communication with station and to negotiate protocols.

3) End-to-end delay for an IP.

4) Terminal interaction delay as a function of the number of IP's transmitted and received. The interaction delay is defined as the time elapsed from terminal origination to departure.

• Blocking and Loss

When a terminal does not successfully identify a repeater (or station) after transmitting an SP for the maximum number of times specified, it is considered blocked. In addition, under certain conditions, terminals will depart from the system without completing communication. This will contribute to additional system loss. The blocking and loss are measured separately since the former indicates the difficulty in entering the system, whereas the latter reflects on the inefficiency of the routing.

- <u>Device Performance</u>

1) Probability that the station is busy. The station is sampled during the simulation and is assumed busy if it is actively receiving or transmitting; otherwise, it is assumed idle. This measure is an indication of the channel traffic at the station.

2) Successful completions by repeaters. The number of packets that each repeater has successfully switched are counted. This indicates the distribution of load in the network and reflects on the power duty cycle of repeaters.

- <u>Other Measures</u>

1) The number of terminals in the system, the total number of packets stored in the system, the number of events to be processed; all as a function of time.

2) The complete output includes a detailed description of the flow of significant communication events.

TYPICAL COMMUNICATIONS PATH

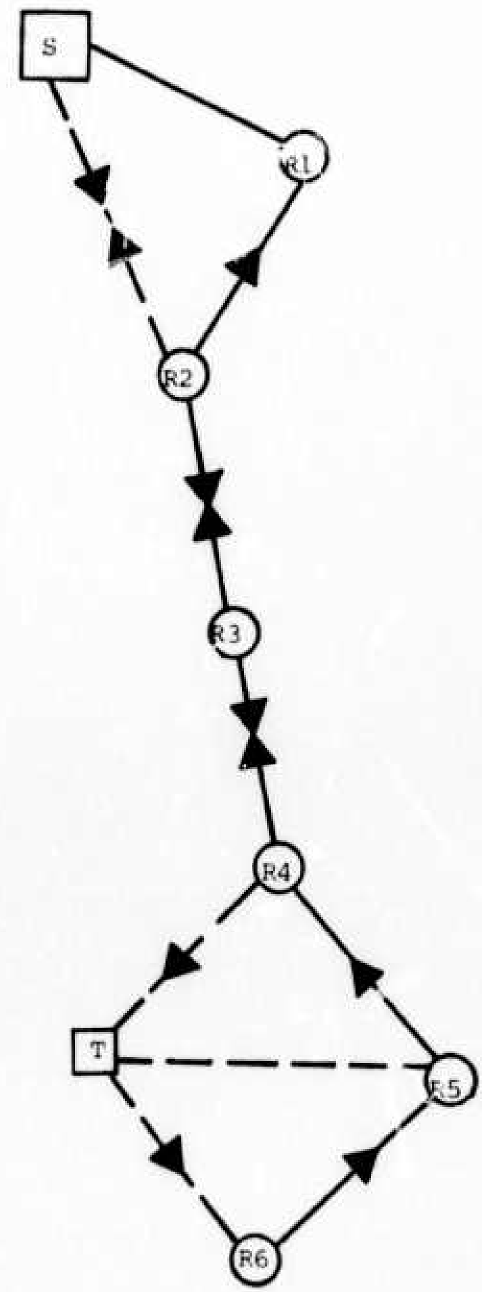


FIGURE 2: The solid lines indicate the labelled path between the repeaters and station. The dashed lines indicate the effective connectivity of terminal and station to repeaters. The arrows indicate the practical transmission path for the particular terminal, to and from the station.

VII. LOGICAL OPERATION OF DEVICES

A. States of Devices

Each device is characterized by a state vector. Some of the state variables will be needed in the physical devices, for example, the label, a parameter indicating the maximum number of transmissions, the maximum handover number to be assigned by repeater and station, the state of occupancy of its storage, etc. Other variables are particular to the simulation. The following are examples:

Operational State of Device

PR - Passive Receive State: The device is in receive state and does not sense any carrier.

AR - Acrive Receive State.

AT - Active Transmit State.

ART- Active Receive and Transmit.

When a device is in state AR or ART, it can be receiving several overlapping packets simultaneously. In the program, we use a common channel configuration (half duplex). Thus, since carrier sense is used for channel access, the device can change to AT only from PR and to ART only from AT.

- Number of Overlapping Packets

This number is incremented by one whenever the device is in state AR of ART and a new packet begins to arrive; and decremented when a packet transmission ends. The number of overlapping packets indicates the number of times an end of packet transmission has to occur before the device changes its state to PR.

- End of Busy Period

This time is recorded for the purpose of saving CPU time. The transmission time of a packet is set to the End of Busy Period plus a random time; otherwise the devide may be called to transmit a packet several times during its busy period.

B. Terminal

When a terminal originates a message, it begins to transmit and retransmit a SP to identify a specific receiver. If it does not identify a specific receiver after a specified number of transmissions, it departs from the system. We say that such a terminal is blocked. When a terminal identifies a specific receiver, it substitutes the label and MHN sent by the receiver itno its IP and begins to transmit its IP. The IP is retransmitted after short waiting periods of time until a HBH Echo Ack is received. At that time, the terminal stores the IP for a longer period of waiting after which the IP is reactivated if an ETE Ack is not received.* The terminal is expecting several IP's

---

* We use the term retransmission when a device waits a relatively short period of time (less then 2 IP slots) and is awaiting a HBH acknowledgement. We say that a packet is reactivated when an end device stores the packet, awaiting an ETE Ack. When a packet is reactivated, it goes through the whole process of retransmissions.

from the station, which are ETE acknowledged by the terminal. When all IP's from the station are received and ETE acknowledged, the terminal departs from the system.

C. Repeater

A repeater does not distinguish between IP's or ETE Acks, except for their transmisison time. When an IP (or ETE Ack) is recrived by a repeater (addressed to it) and the repeater has available storage, it stores the packet, decrements the MHN, modifies the packet label according to the routing, and begins to transmit and retransmit the packet, awaiting the Echo Ack. When an Echo Ack is received, the repeater discards the packet. When a repeater is not successful in transmitting along the "shortest" path, it begins to search for an alternate receiver by transmitting SP's. When one is found, it transmits the entire packet to it; otherwise, it discards the packet. When a repeater receives an SP, it checks whether it has available storage, if it does, it makes one attempt to transmit a RSP and then discards it. When a repeater receives a RSP, it tests whether it needs one, if it does, it used the label, otherwise it discards it. The repeater currently simulated has buffer storage for two packets; one exclusively used for packets directed towards the station, and the second for packets toward the terminal. In addition, the repeater can inspect all packets that is receives, which are stored in common arrays in the simulation program. Thus, from a practical viewpoint, buffer storage for three packets per repeater are provided in the simulation program.

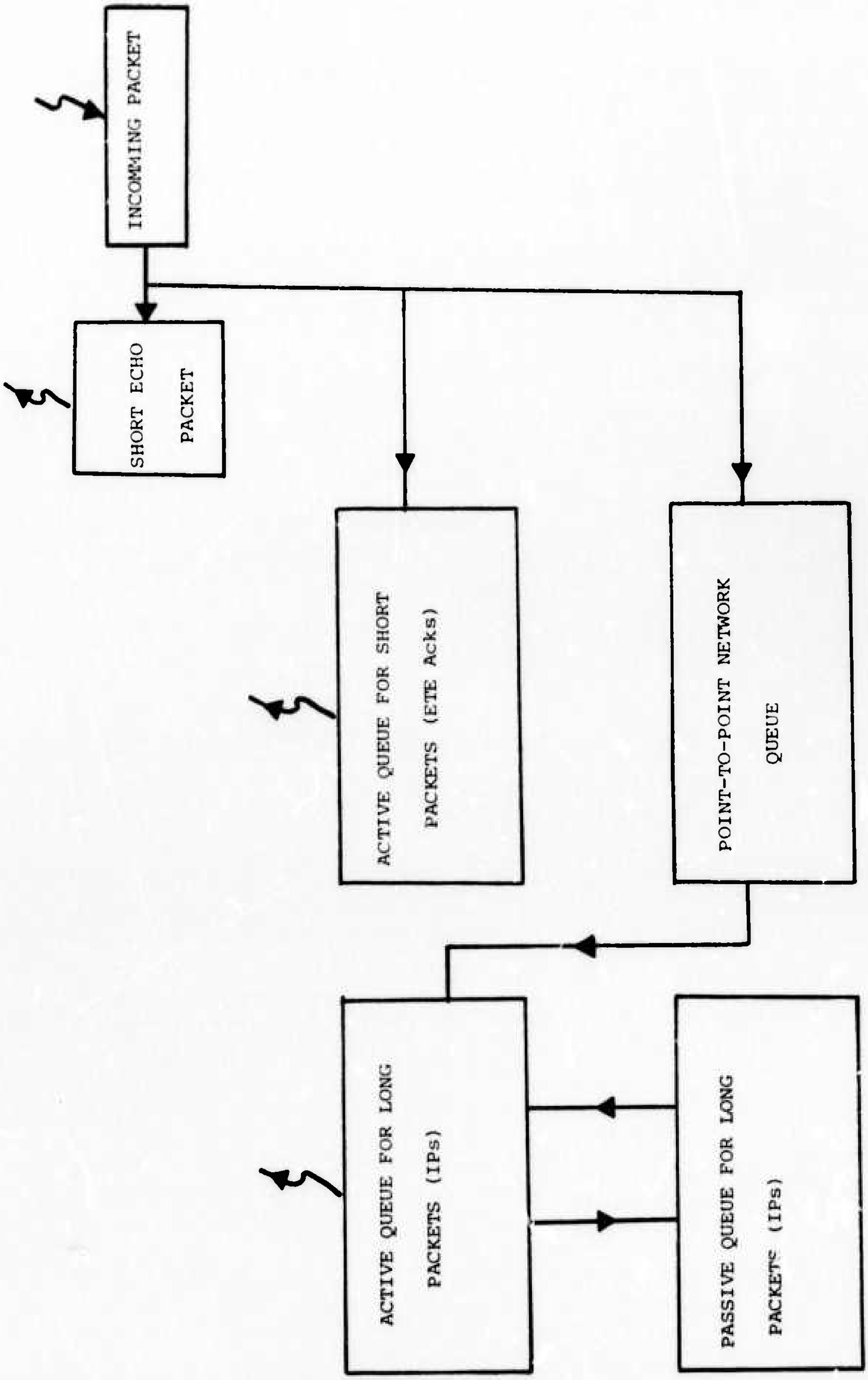


FIGURE 3: STORAGE IN STATION OF IP's AND ETE ACK's.

D. Station

The storage organization in the simulated station is shown in Figure 3.* There are two queues for active packets. Packets in these queues are active in the sense that they are retransmitted after short random periods of waiting until an Echo Ack is received. The active queue for long packets contains IP's from the station to terminals. Once an IP is Echo acknowledged, it is stored in the passive queue for a longer period, after which it is reactivated if an ETE Ack from the terminal is not received. The active queue for short packets contains ETE Acknowldegement packets to terminals, and these have priority over the long active apckets. The ETE Ack packets are, obviously, discarded once an Echo Ack is received. The point-to-point (PTP) network queue simulates the interaction of the packet radio network with a PTP network. When a new IP is received from a terminal, it is stored in the PTP network queue fir a random time, after which a response message containing several response IP's to that terminal are generated and placed into the active queue for long packets. The station responds immediately to SP's, and ignores RSP's.

E. Flow Diagrams of Devices

Figures 4, 5, and 6 show the flow diagrams of the devices used in the simulator. These diagrams are simplified to the extent that they show "what to do" but not sufficiently detailed to show "How to do it." The latter depends on the particular system simulated, i.e., the routing, the channel configuration, etc.

A device is called from the subroutine EVENT; the calling sequence includes, among others, an interrupt number which indicates to the device, the task that is has to perform. The only event which

---

* Figure 3 shows only the storage of IP's and ETE Acks for transmission to terminals in the packet radio network.

is external to a device is that with an interrupt = 1. All other entries to a device are due to events generated by the device itself. Thus, the number of exogeneous events is very small compared to the number of events generated by devices; in particular, when the offered data rate to the system is high.

It is clear that when the offered traffic rate to the system is high, there will be many collisions of packets. Thus, to save computing time, each device was coded so that it does not examine the content of the packet or whether the packet is addressed to it, at the beginning of packet reception (see diagrams). This is done at the end of packet reception, providing there was no interference.

There are parts of the subroutines of Repeater, Station, and Terminal which are identical and thus, coded into subroutines. These relate to the identification of the header content, and the channel access mode. Some of these can be associated with the modem of the physical devices. The parts which differ involve mainly the processing needed once the packet has been identified, e.g., queueing at the station, ETE Ack by end devices, etc.

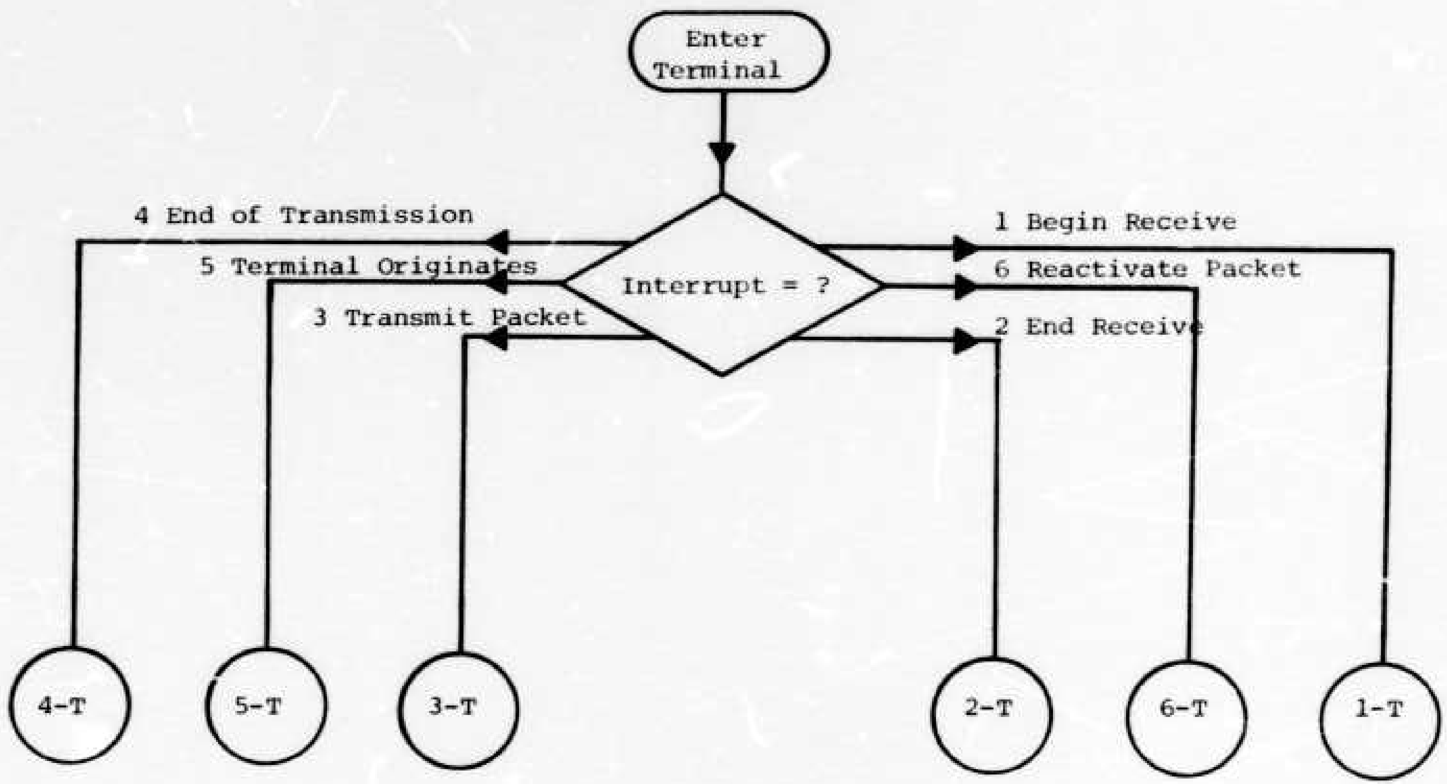


FIGURE 4A

TERMINAL DEVICE FLOW DIAGRAM

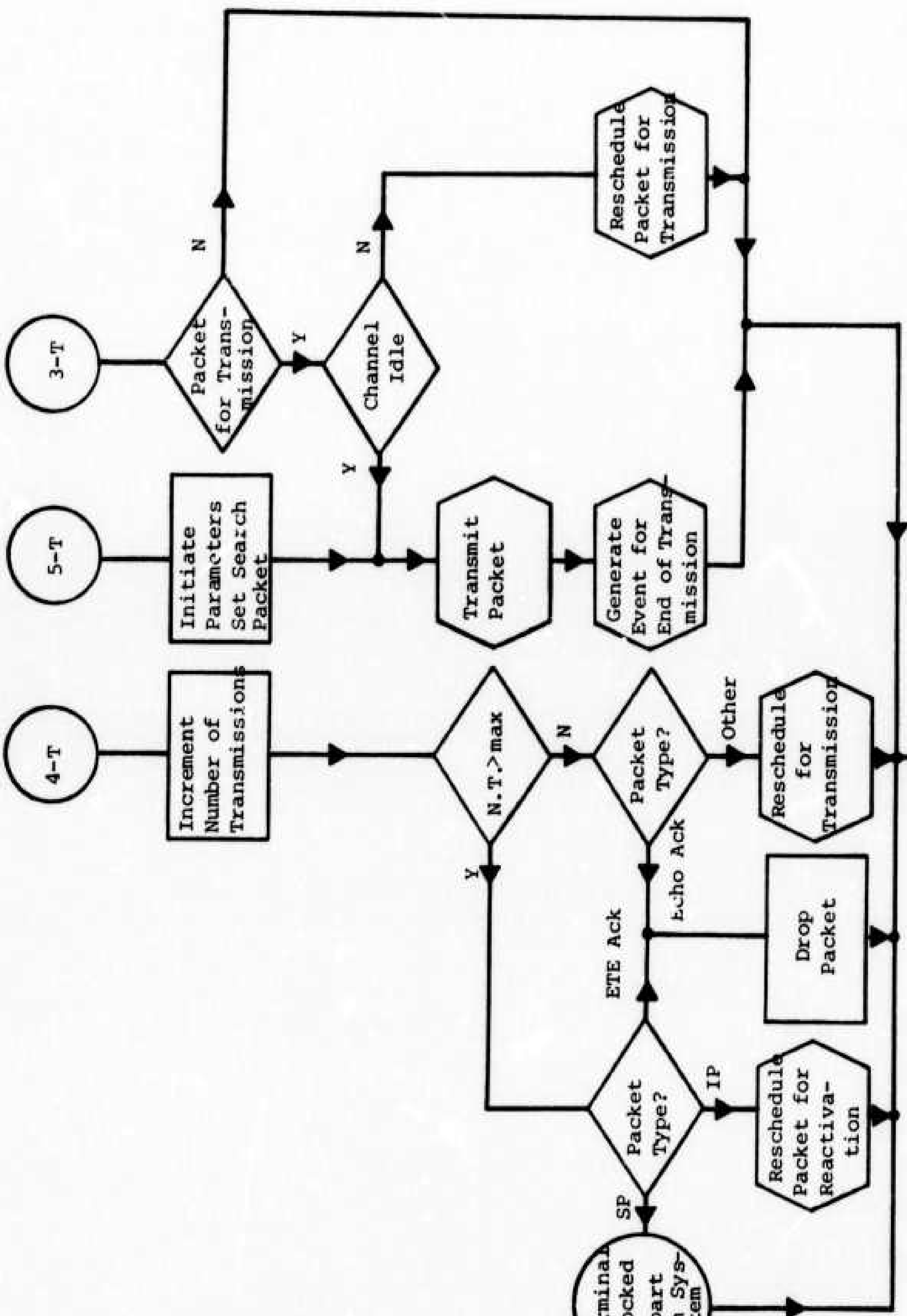


FIGURE 4B

TERMINAL DEVICE FLOW DIAGRAM

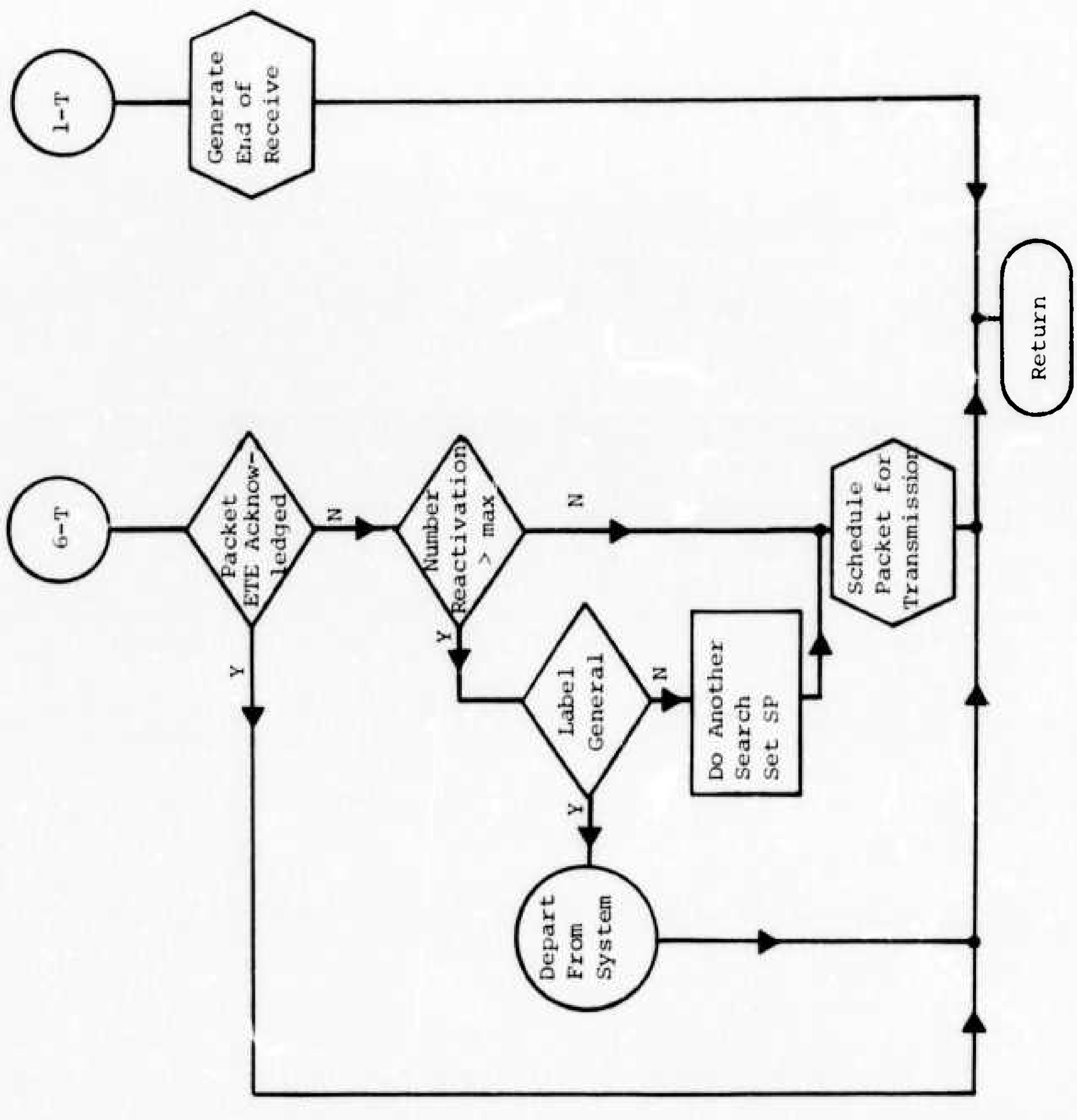


FIGURE 4C

TERMINAL DEVICE FLOW DIAGRAM

TERMINAL DEVICE FLOW DIAGRAM

Network Analysis Corporation

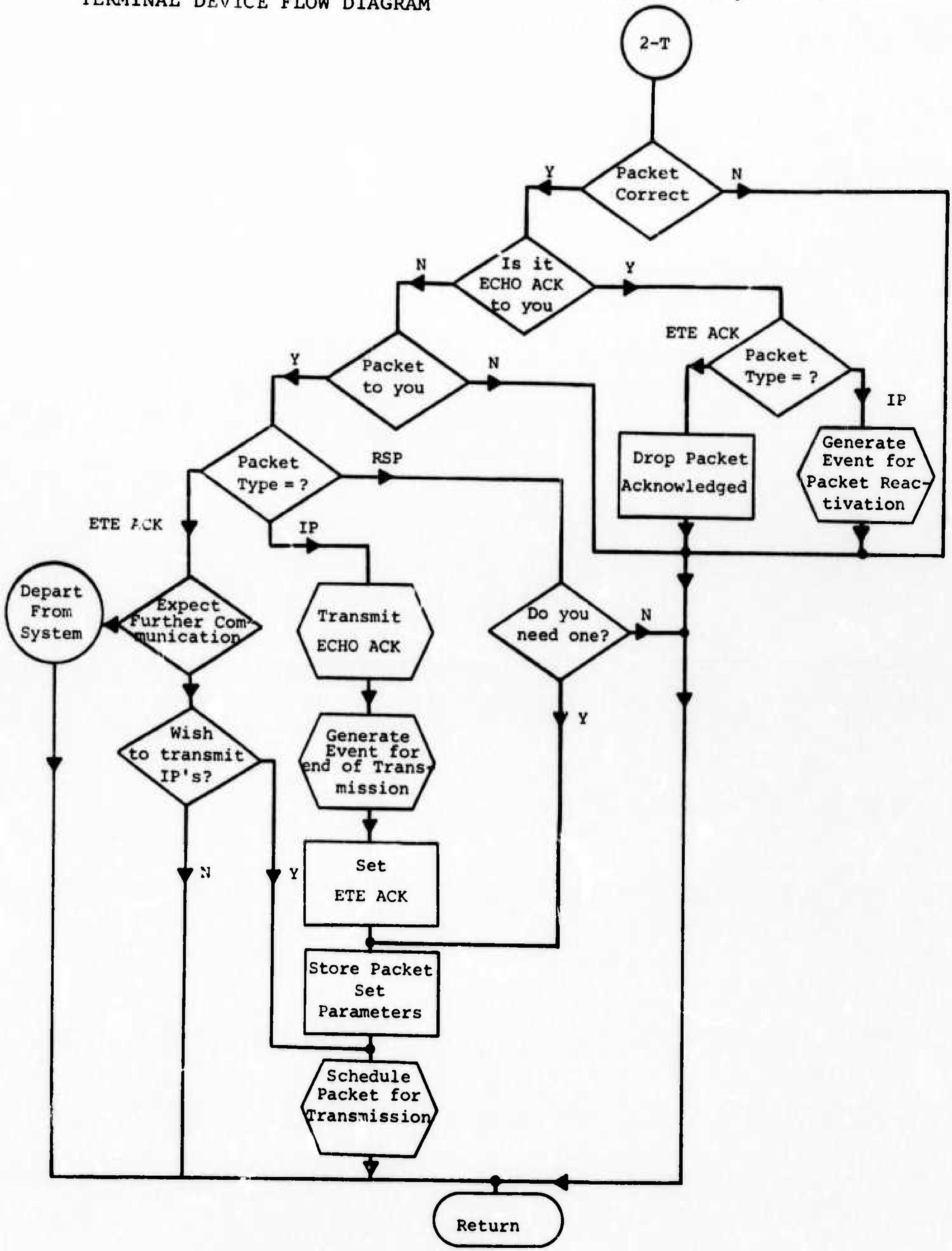


FIGURE 4D

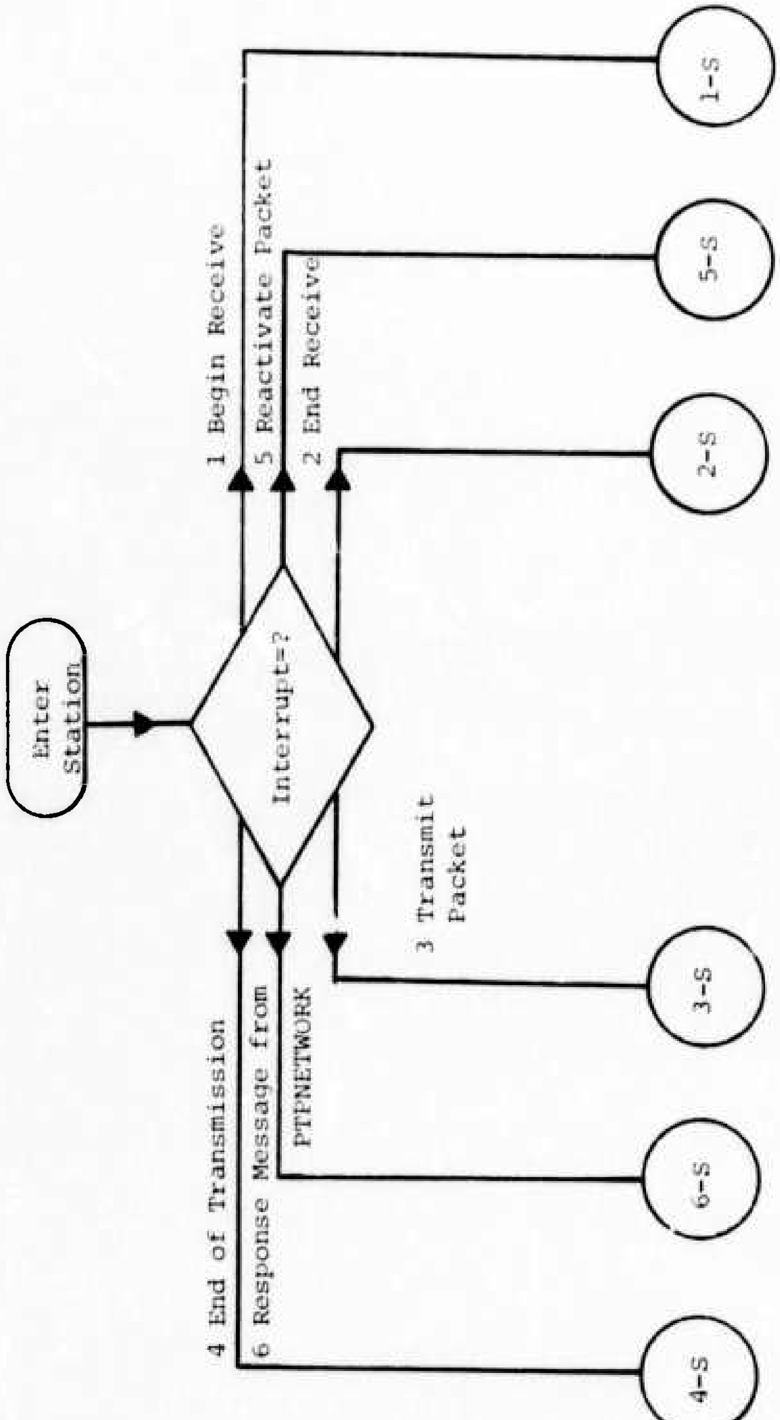


FIGURE 5A

STATION DEVICE FLOW DIAGRAM

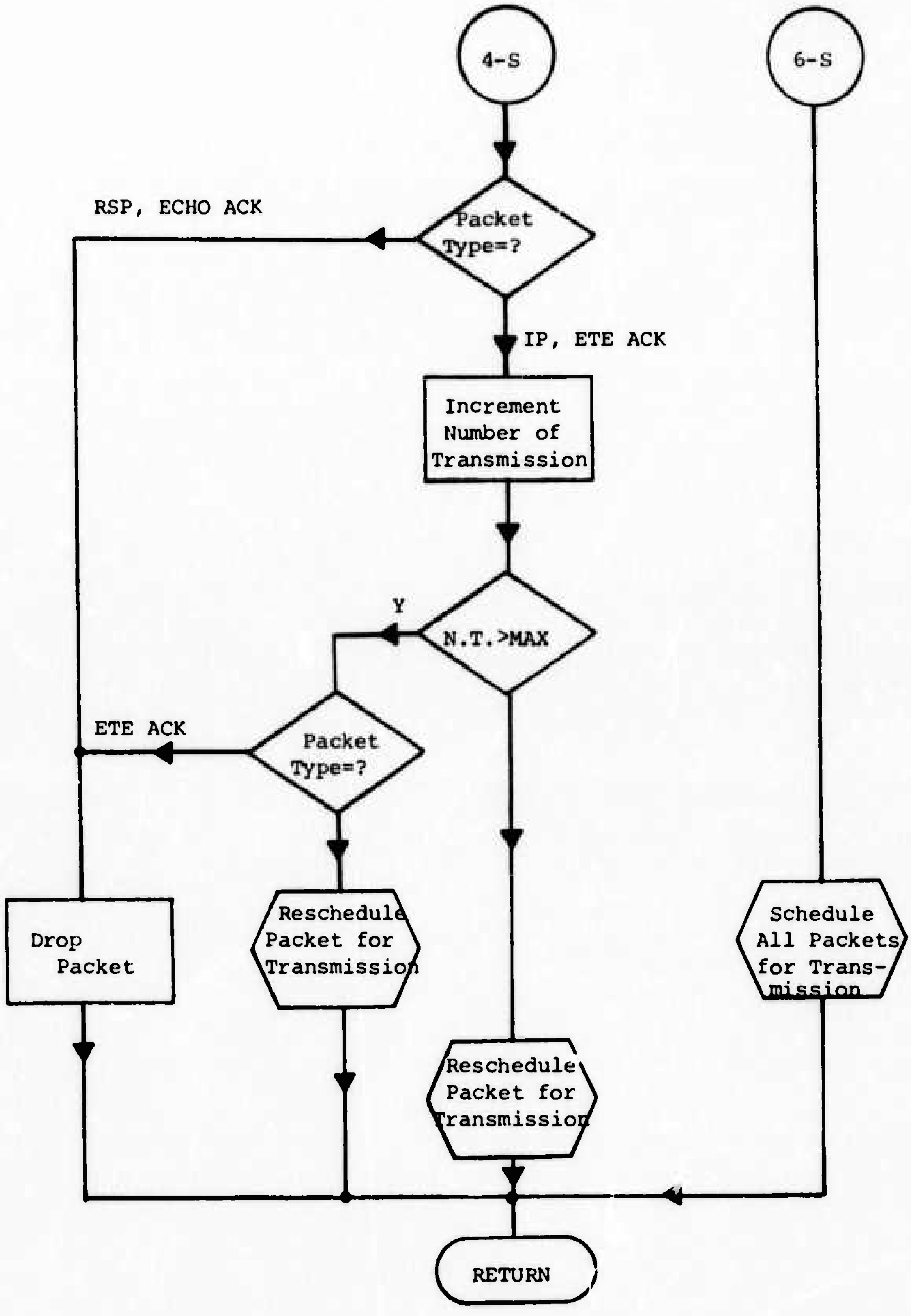


FIGURE 5B

STATION DEVICE FLOW DIAGRAM

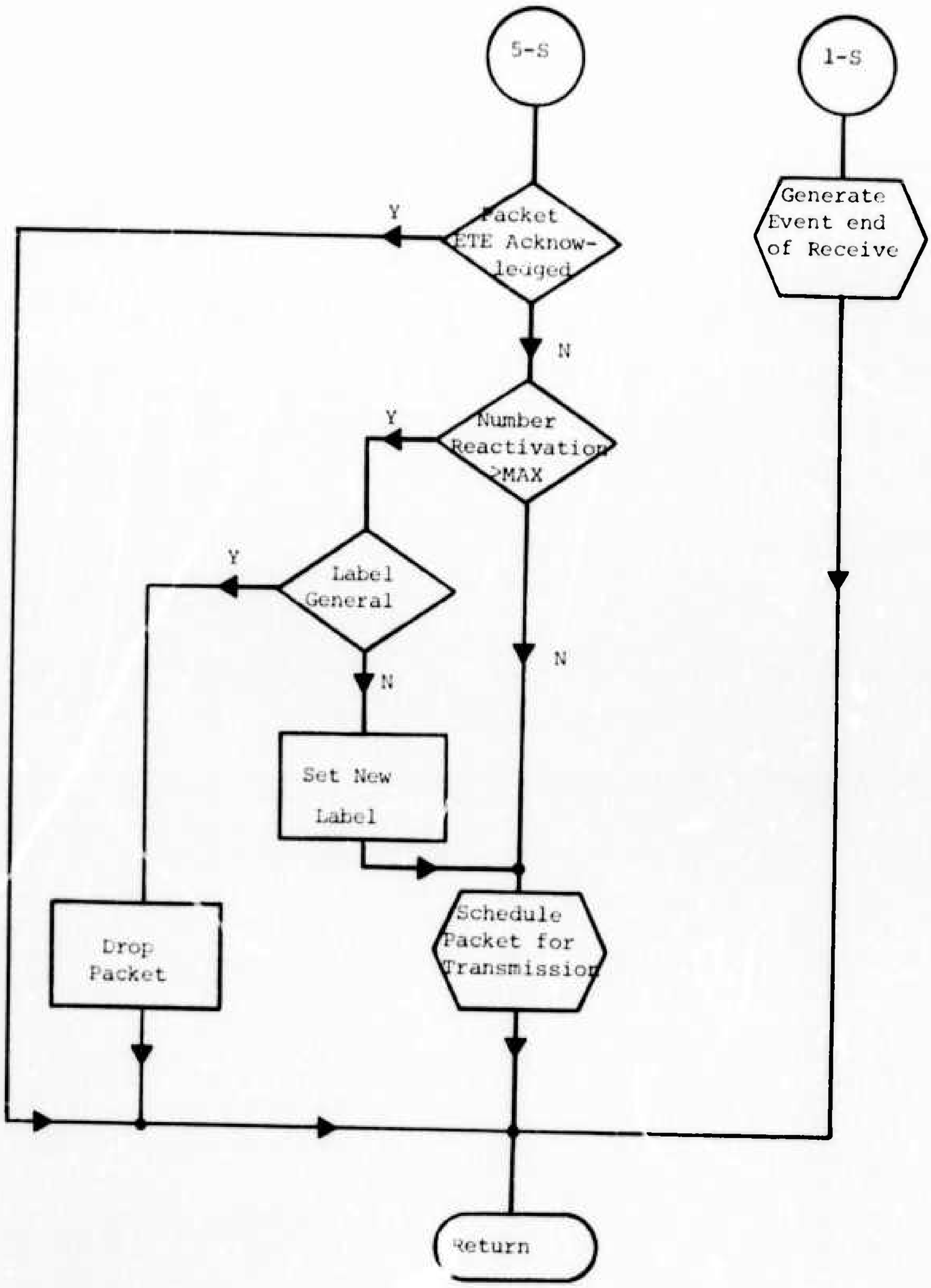


FIGURE 5C

STATION DEVICE FLOW DIAGRAM

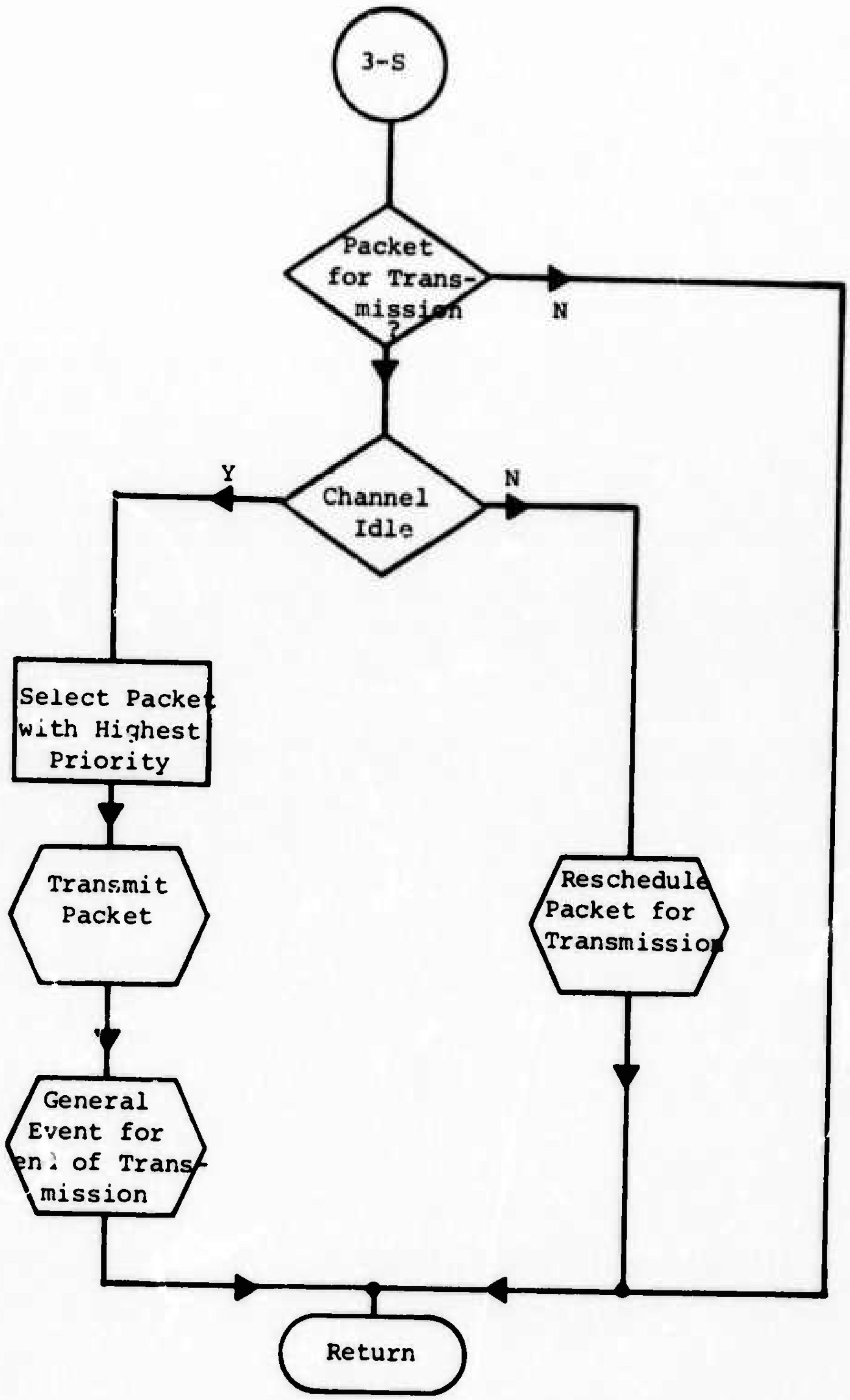


FIGURE 5D

STATION DEVICE FLOW DIAGRAM

STATION DEVICE FLOW DIAGRAM

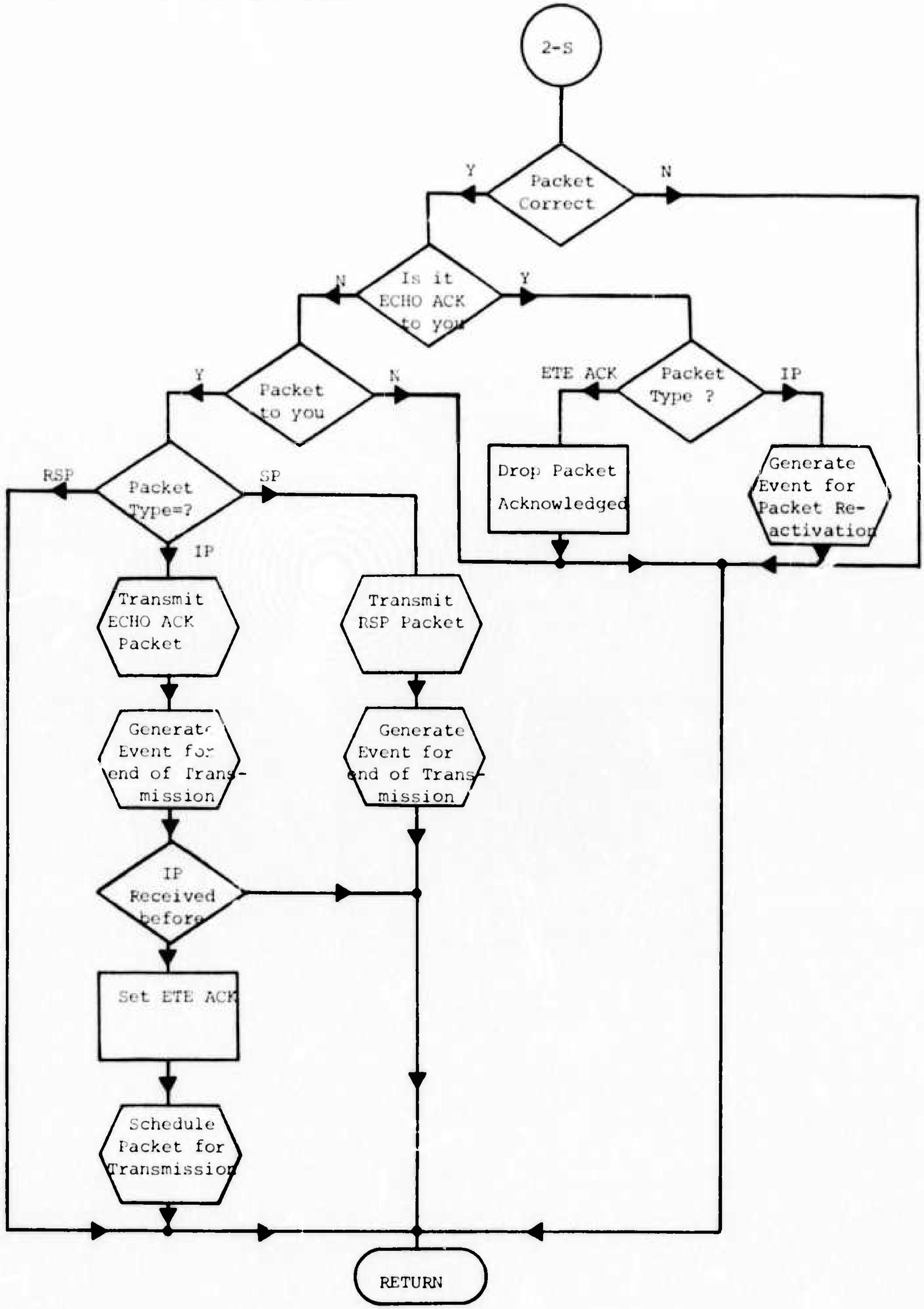


FIGURE 5E

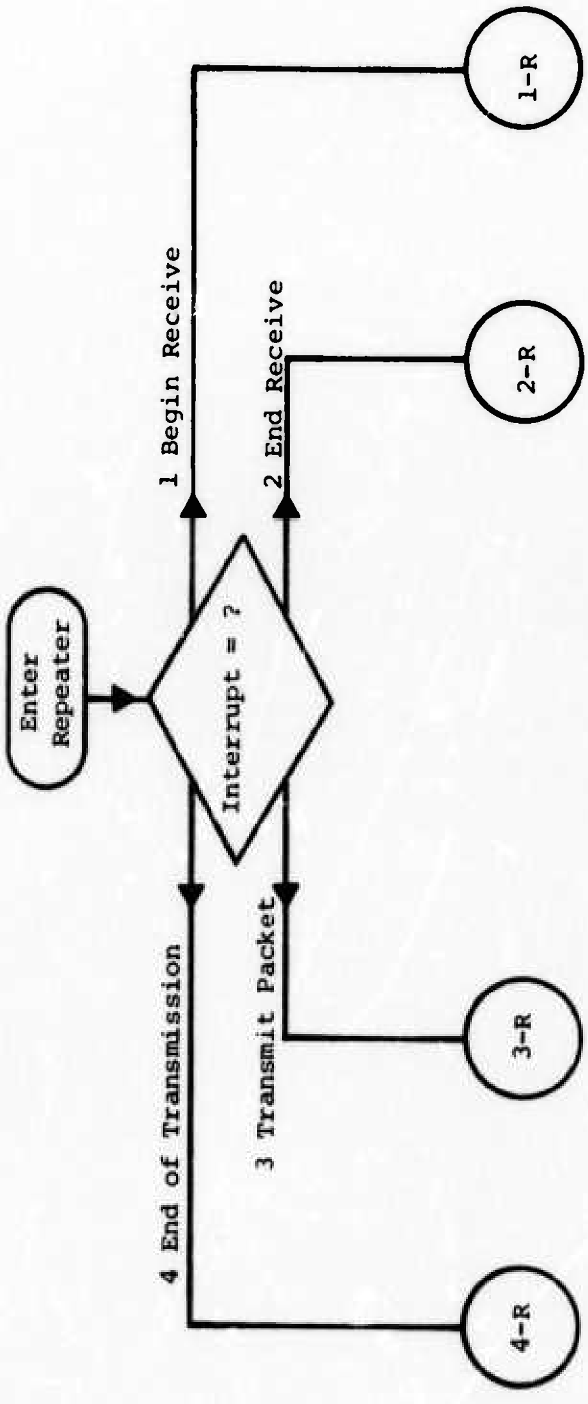


FIGURE 6A

REPEATER DEVICE FLOW DIAGRAM

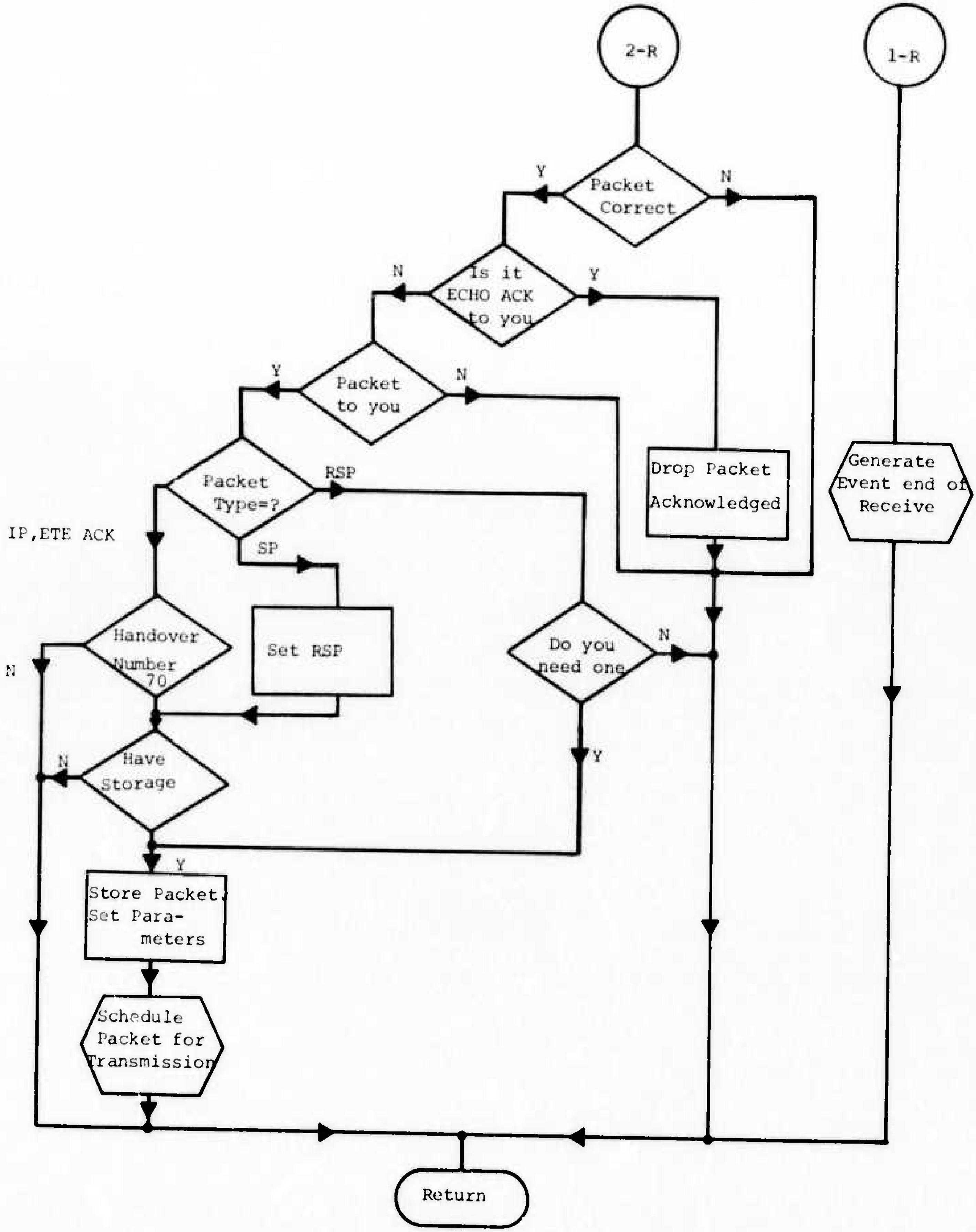

2-R
1-R
Packet Correct
Y
N
Is it ECHO ACK to you
N
Y
Packet to you
Y
N
Packet Type=?
RSP
SP
IP,ETE ACK
Drop Packet Acknowledged
Generate Event end of Receive
Handover Number 70
N
Set RSP
Do you need one
N
Y
Have Storage
N
Y
Store Packet, Set Parameters
Schedule Packet for Transmission
Return

FIGURE 6B

REPEATER DEVICE FLOW DIAGRAM

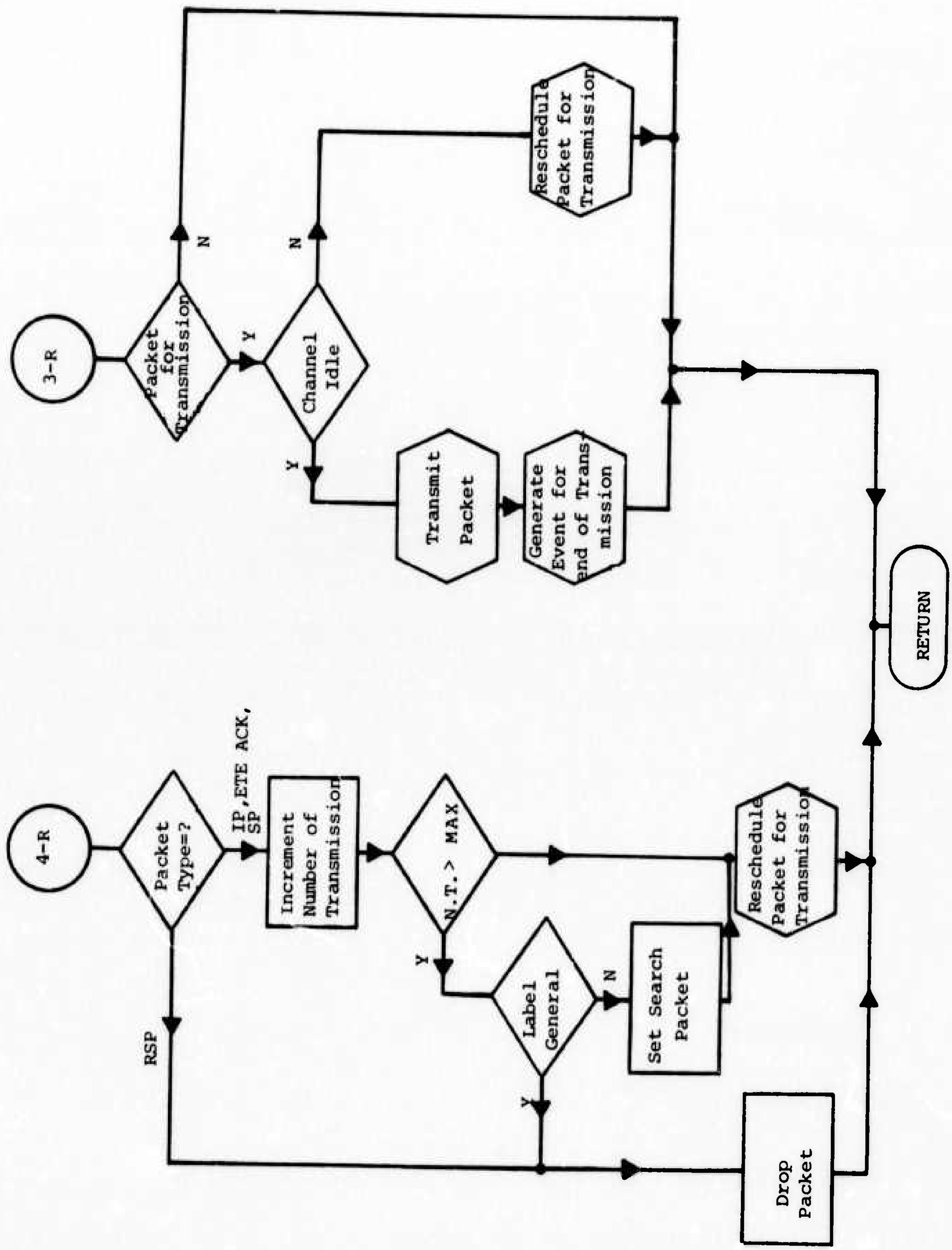


FIGURE 6C

VIII. SUBROUTINES OF THE SIMULATOR

A. Data Structure and Management Subroutines

EVENT Takes the next event out of the Event Data Structure for execution.

INHEAP Adds an event to the heap in the Event Data Structure.

INMESS Allows the introduction of special messages such as acknowledgements, control messages and the like into the Message Data Structure.

INPUT Reads the input parameters and determines the placing of repeaters and stations.

MESSREL Is called by device routines to release a message as soon as all packets representing the message are deleted.

NEWMESS Generates next exogenous message and adds to the Message Data Structure.

NEWPACK Adds a new packet to the Packet Data Structure.

NXTEVNT Adds a new event to the Event Data Structure.

OUT Prints out intermediate data for debugging.

OUTHEAP Takes the index of the next event time from the heap.

PRSIM The driver routine. (main program)

MEASURE Collects data on system performance.

MCOUNT Counts the number of packets associated with each terminal which are stored in the system.

B. Communication and Device Subroutines

REPEAT Main subroutine of repeater.

STATION Main subroutine of station.

TERMINAL Main subroutine of terminal.

DEVINIT Reads parameters which devine the particular communication system, labels, and flow control parameters. Initializes states of devices.

BGNRCV Maintains states of devices related to the RF channel (e.g., number of overlapping packets).

ENDRCV Same as above at the end of packet reception.

ECHO Records that a device is receiving an echo acknowledgement.

SRER Called when a non-overlapping packet is received for testing packet type and label.

ALTROUT Called after repeater receives an RTS, checks whether repeater needs one, and has not used one before.

SRTTRT Transmits packet and generates an event for the end of packet transmission.

REPNEXT Determines which packet of a repeater is to be transmitted next.

TERSTOR Stores correct IP's received by terminal and generates event for transmission of an ETE Ack.

SNREEKO Called by station after receiving an Echo Ack. Identifies and maintains the queue in which the acknowledged packet is stored. If it is in IP, then it transfers the packet to another queue where it waits for an ETE Ack or for reactivation.

SHIFTQ Shifts packets in the various queues of the station.

SNREPAK Called by station after correctly receiving a packet. If the packet is an ETE Ack, then subroutine drops the packets acknowledged, maintains proper queues and the message counts. If it is an IP, subroutine verifies that same packet has not been received before, and if so, it generates packet and an event for transmission of an ETE Ack, and also generates a random time and an event for the arrival of the response message from the PTP network.

RESPONS Called by station, sets all response packets to a terminal into the active queue and generates events for transmitting them.

SEKOTRT Used by station to transmit an Echo Ack for the last hop.

CONNECT Determines the most efficient repeater to which station should address packet when transmitting to a terminal.

TRANSMT Called by a device which transmits a packet. Puts packet in list structure; determines all the devices that should receive the packet, the exact time for beginning to receive it; and generates the events to devices.

C. Summary of Acronyms

Ack - Acknowledgement

AR - Active Receive

ART - Active Receive and Transmit

AT - Active Transmit

ETE - End-to-end

HBH - Hop-by-hop

IP - Information packet

Label- An address assigned to a device for routine purposes

MHN - Maximum handover number

MNT - Maximum number of transmissions

RP - Passive Receive

PTP - Point-to-point

RSP - Response to search packet

SP - Search packet

IX. OBSERVATION OF TRAFFIC FLOW IN THE PACKET RADIO NETWORK

The first system simulated was a Common Channel Single Data Rate system, in which the station is routing traffic as a repeater (Waive Station). We denote the system as CCSDR (NS). The system defined has a single data signalling rate for communication between terminal and repeater (or station) and in the repeater-station network; the channel is used in a half duplex mode. When the station is routing traffic as a repeater, it cannot receive packets not specifically addressed to it.

In all experiments reported here, the labels of repeaters and staiton were preassigned. The hierarchical (directed) labelling scheme of the system in this experiment are shown in Figure 7. Figure 8 shows the connectivity of the repeaters and station. That is, when a device transmits, all the devices connected to it by line are within an effective range and "hear" the transmission.

The objective of the first series of experiments was to observe the detailed operation of devices and the efficiency of the system. The following observations were made:

1. The "critical hop" in the system is that between the first level repeaters and the station. This was concluded by observing the frequency at which repeaters begin to search and at which they discarded packets, and from the observation that there is no significant difference in the delay when the number of hops from the station that a packet travels is increased.

2. There is a higher probability of end-to-end successful completion when routing from the station to a terminal than when routing from a terminal to the station. Practically, there is almost "no" difference in time delay between the delay of an information packet from the terminal arriving at the station and the time that the terminal receives an ETE Ack from the station.

3. Many packets associated with terminals that have departed from the system are routed in the network.

The effect of improving the routing capacibilities of the station can be readily observed. In particular, one can see in Figures 7 and 8 that while the connectivity station is 7, there are only 4 repeaters labelled from the station. Consequently, the station is busy of the time with non-useful traffic. This situation can be improved by changing the routing of the station so that: (1) it receives any packet that it can hear and which is (eventually) addressed to it; and (2) it transmits response packets to the repeater nearest to the terminal along the routing path that it can reach. This change was implemented for all system studies subsequent to the initial experiments. Apart from the change implemented, the observation suggests that particular attention should be given to the design of the repeater network in the neighborhood of the stations. It is also noted that these repeaters have a higher power duty cycle since they handle all packets collected from other parts of the network. The routing change made at the station enables the allocation of many more repeaters in the neighborhoos of the station, than are functionally needed, without resulting an increase in the artivicial traffic generated. The exact labelling of these repeaters is also not critical.

One of the reasons leading to observation 2 is that the station has a higher probability then the first level repeaters of successful transmission over the critical hop, because it is the largest user and does not interfere with its own transmissions. Theoretically, one may expect a similar conclusion when considering transmission in a section of the network in which two repeaters, one of which "homes" on the other, compete. This, however may not be realized in the system simulated because of the limited storage available in repeaters.

Observations 2 and 3 suggest a change in the Terminal-Station protocol. The basic question is whether a terminal should release itself from the system or whether it should be released by the station. The former was initially simulated. It was observed that in many cases, a terminal departed from the system after receiving an Echo to the ETE Ack for the last IP without this ETE Ack arriving at the station. This resulted in the reactivation if IP's by the station for this terminal, the routing of these packets in the net, and then the maximum number of transmissions and search by the repeater nearest to the terminal. The protocol simulated in the systems discussed later is such that the last packet must always be from the station to the terminal. This transmission may be considered as a terminal release packet. Another change in protocol implemented is that whenever possible, the terminal acknowledges a sequence of packets rather than individual ones, to reduce the overhead in the direction towards the station.

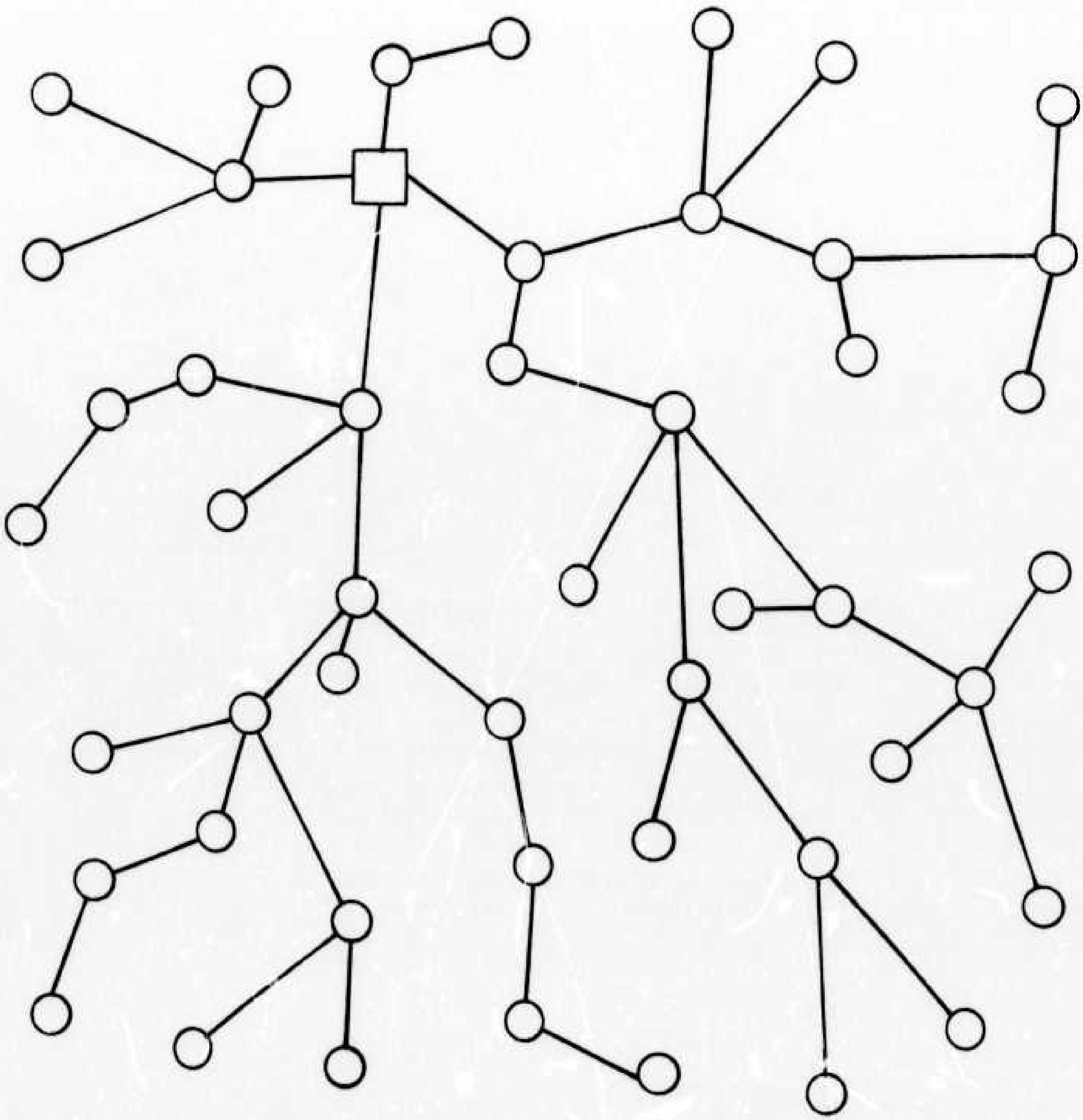


FIGURE 7

HIERARCHIAL LABELLING SCHEME

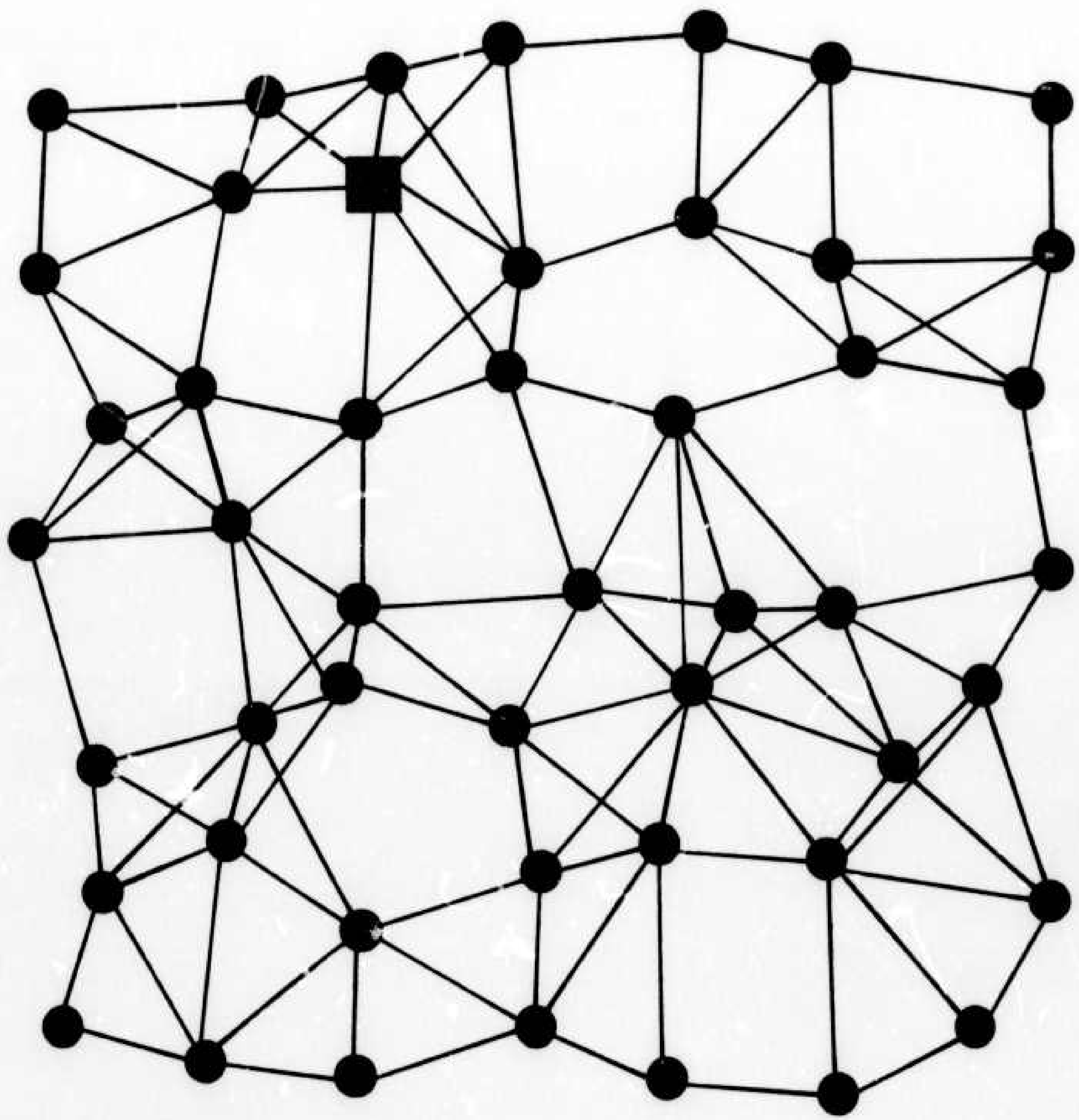


FIGURE 8

CONNECTIVITY OF REPEATERS & STATIONS

X. THE TRADEOFF BETWEEN TRANSMISSION RANGE OF DEVICES AND NETWORK INTERFERENCE

For the experiments discussed in the previous section, it was assumed that Repeater-Repeater range is the same as Terminal-Repeater range. This, however, is not always a realistic assumption since repeaters can be placed on elevated areas and can have more power then terminals, (especially hand held terminals). Thus, if repeaters are allocated for area coverage of terminals, the repeater range will be higher than terminal range and higher network connectivity or device interference will result.

The problem which then arrises is to determine the impact of this interference on system performance. Alternatively, one may seek to reduce repeater transmission power when transmitting in the repeater-station network. To study this issue, two CCSDR systems were simulated, one with high interference CCSDR (HI), and the other with Low Interference CCSDR (LI). The routing labels of the two systmes were the same and are shown in Figure 7. The interference of the CCSDR (LI) system is shown in Figure 8 and the interference of the CCSDR (HI) system in Figure 9. Figure 9 shows only the connectivity of two devices in the network.

The results are shown in Figure 10 and Table 1. Figure 10 shows the throughput of the two systems as a function of time while Table 1 summarizes all other measures of performance. The third row of Table 1 summarizes performance of the high interference system under an improved set of repeater labels. This experiment is discussed in detail in the next section. It is clear that the high interference system is much better than the low interference system. The only measure of the low interference system which is better is terminal blocking which is a direct result of the low interference feature. In fact, CCSDR (LI) is saturated at the offered traffic rate. This can be seen from the fact that the throughput is decreasing as a function of time; the relatively high total loss; and the low station response*. The CCSDR (HI) with improved labels compared in Table 1, is better than the other two systems. This demonstrates the importance of proper labelling. The experiments of this section demonstrate that it is preferable to use high transmitter power to obtain long repeater range, despite the network interference that it results.

* The average number station response packets assumed for these studies is 2.0.

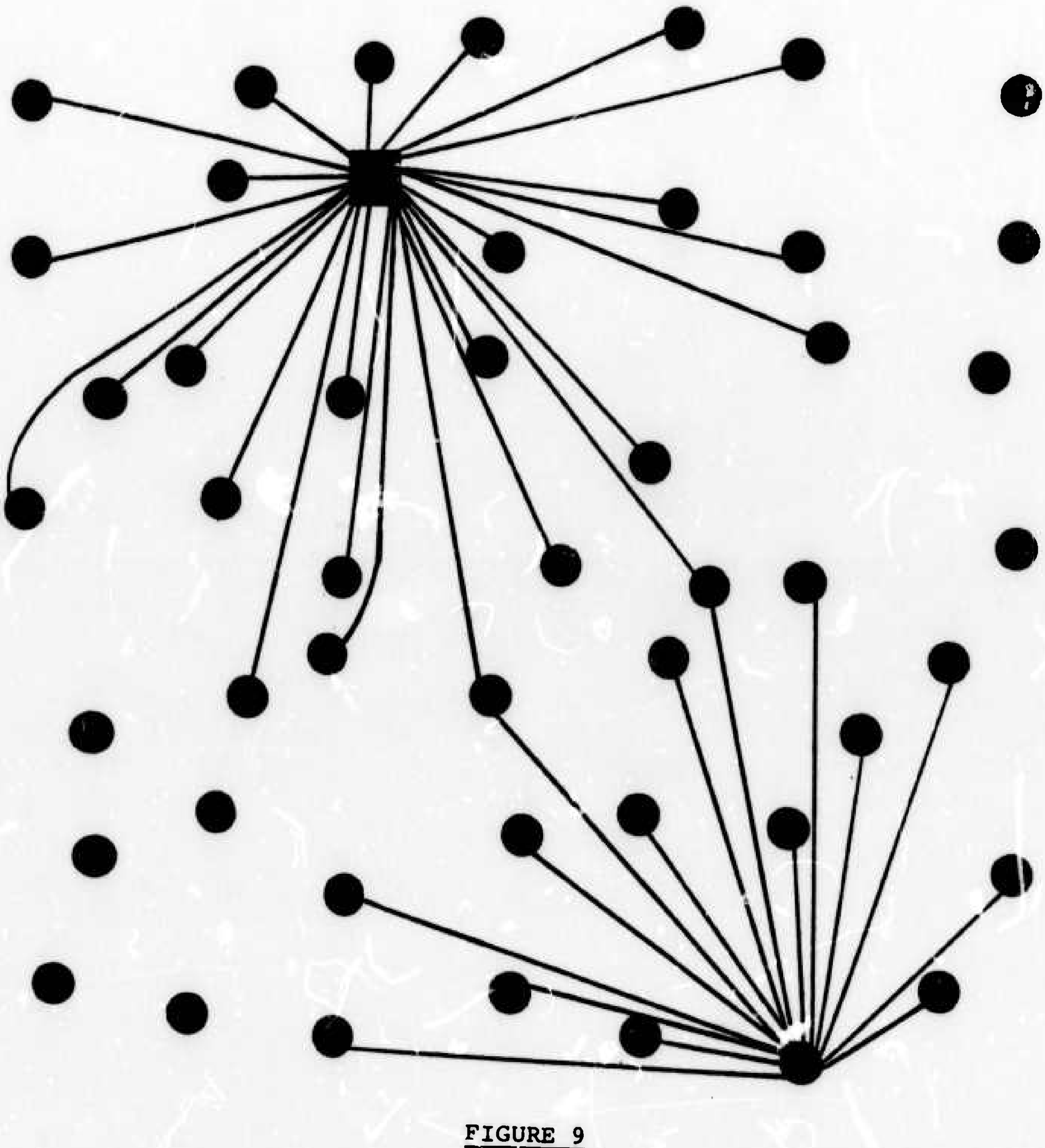


FIGURE 9

INTERFERENCE OF CCSDR (HI) SYSTEM

THROUGHPUT VS. TERMINAL SLOTS: CCDR (HI) & CCSDR

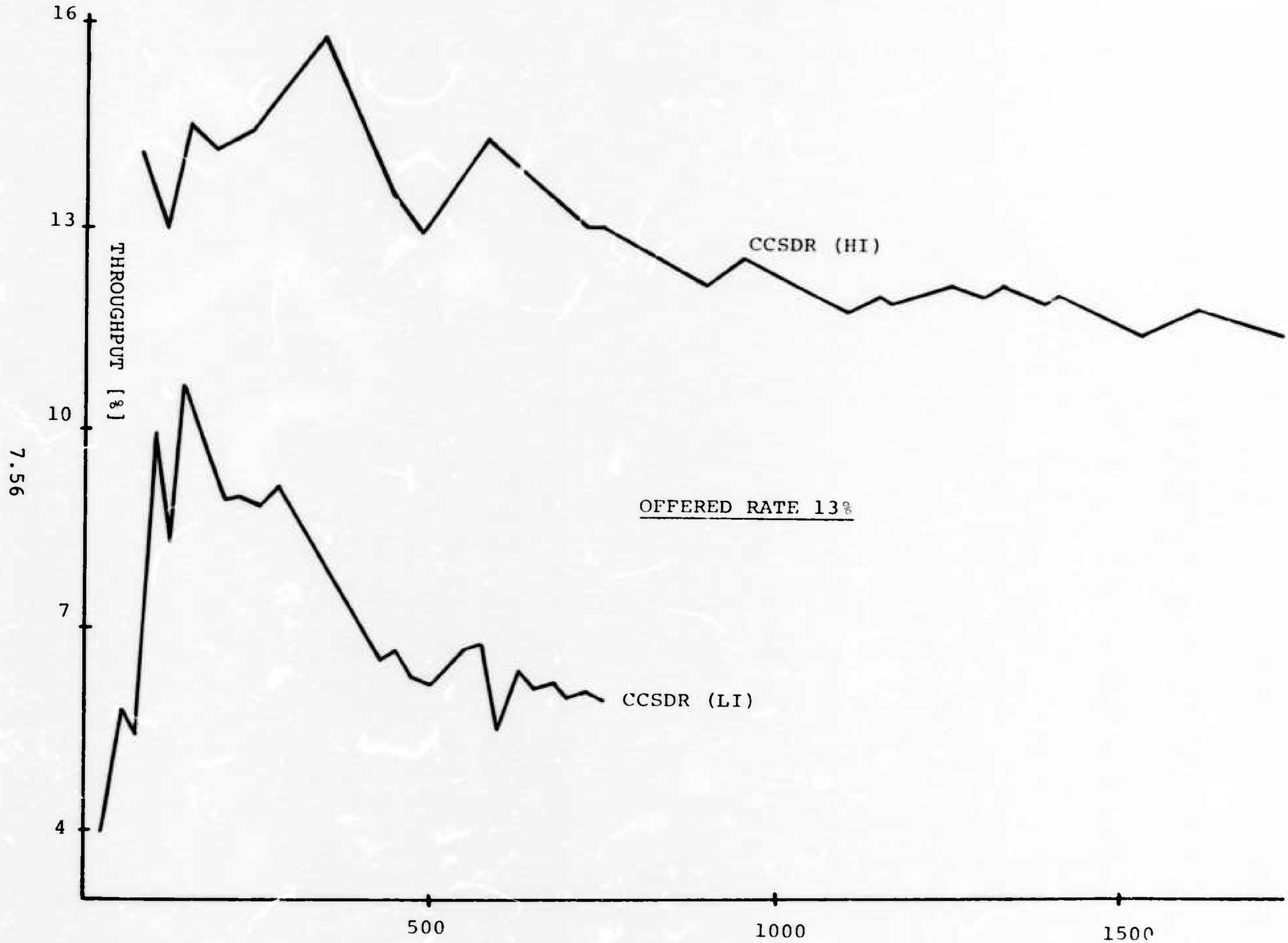


FIGURE 10

	OFFERED RATE [%]	THROUGHPUT [%]	DELAY OF IP TO STATION [Terminal Slots]	RATE OF STATION RESPONSE	PROB. STATION BUSY	% OF IP BLOCKED	TOTAL % OF IP LOSS	TERMINALS REMAINING
CCSDR (LI)	13	5.95	40.11	1.14	.53	2.98	32.53	13
CCSDR (HI)	13	10.55	23.93	1.81	.43	9.83	9.83	13
CCSDR (HI) (Improved Labels)	13	12.14	16.61	2.06	.50	10.63	11.41	10

TABLE 1

XI. SINGLE VERSUS DUAL DATA SIGNALLING RATES NETWORKS

The results of the previous section demonstrate that a better performance system is obtained when repeaters and station use high power to obtain long range despite the interference that results. We now examine the problem of whether repeaters and station should use their fixed power budgets to obtain a long range with a low data rate channel or have a short range with a high data rate channel. The following systems were studies.

- A CCSDR (HI) of the previous section with improved labels, which we denote by CCSDR. That is, we take advantage of the high range to improve the routing labels of repeaters and obtain fewer hierarchy levels. The routing labels used are shown in Figure 11, and the connectivity is shown in Figure 9.

- A Common Channel Two Data Rate (CCTDR) system with the routing labels as in Figure 7 and connectivity as in Figure 8.

In the CCTDR system, the terminal has a low data rate channel, the same rate as in the single data rate system, for communication with a repeater or station. Repeaters and station have two data rates. The high data rate is used for communication in the repeater-station network. The two data rates use the same carrier frequency so that only one can be used at a time.

The two systems are tested with offered rates of 13% and 25%.* The throughput as a function of time for the two runs are shown in Figures 12 and 13, respectively; and the summary of other measures is given in Table 2. The comparison demonstrates that the CCTRD

* In the simulation runs we used the inverse square low for the relation between data rate and distance, rather than the result in [9];this however, favors CCSDR.

system is superior to the CCSDR system, in terms of throughput, delay, and other measures. One can see that the CCSDR system is saturated at an offered rate of about 13%.

- Effect on Blocking Level

In Table 2, one can see that one reason for the relatively low throughput of the CCSDR system at an offered rate of 25% is due to blocking. Furthermore, the fraction of time that the station is busy has decreased. This may suggest that the station may be able to handle more terminals providing they are able to enter the system. To examine this point, we ran the CCSDR system with offered rate of 25%, and relaxed the constraint for entering the system. Rather than resulting in better performance, this step resulted in reduction in blocking and increase in delay. The throughput increased to 12.63%, the blocking decreased to 18.35% and the total loss decreased to 30.73%. On the other hand, the delay increases to 57.82, the fraction of time the station is busy increased to .57, and the rate of station response decreased to 1.32.

To conclude, when we enabled more terminals to enter the system, the throughput increased insignificantly, from 12.20% to 12.63%; on the other hand, the average packet delay increased significantly, from 34.97 to 57.82 terminal slots. This suggests that one of the important design problems in the packet radio system is the blocking level of terminals.

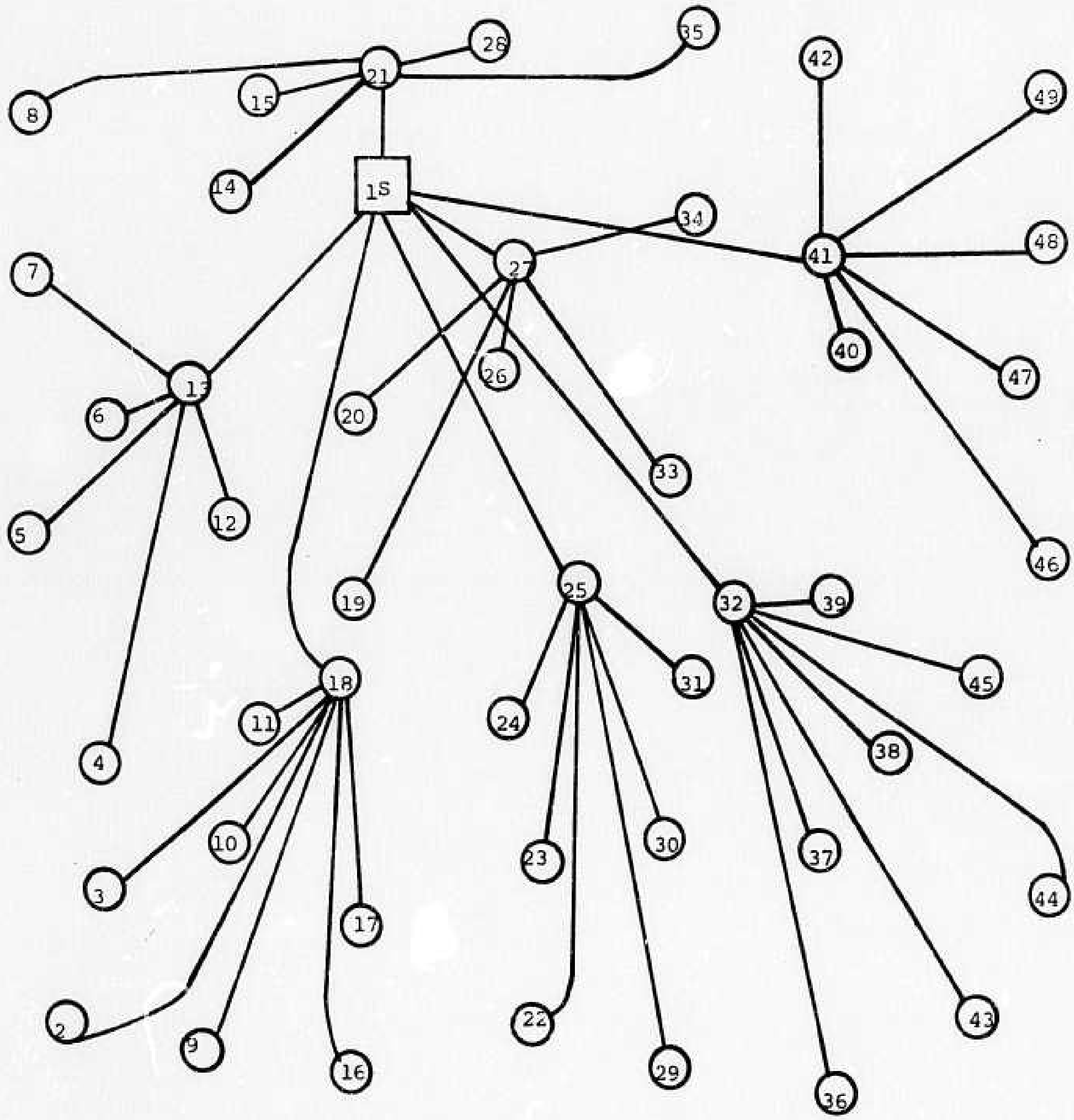


FIGURE 11

CCSDR (HI) SYSTEM WITH IMPROVED LABELLING

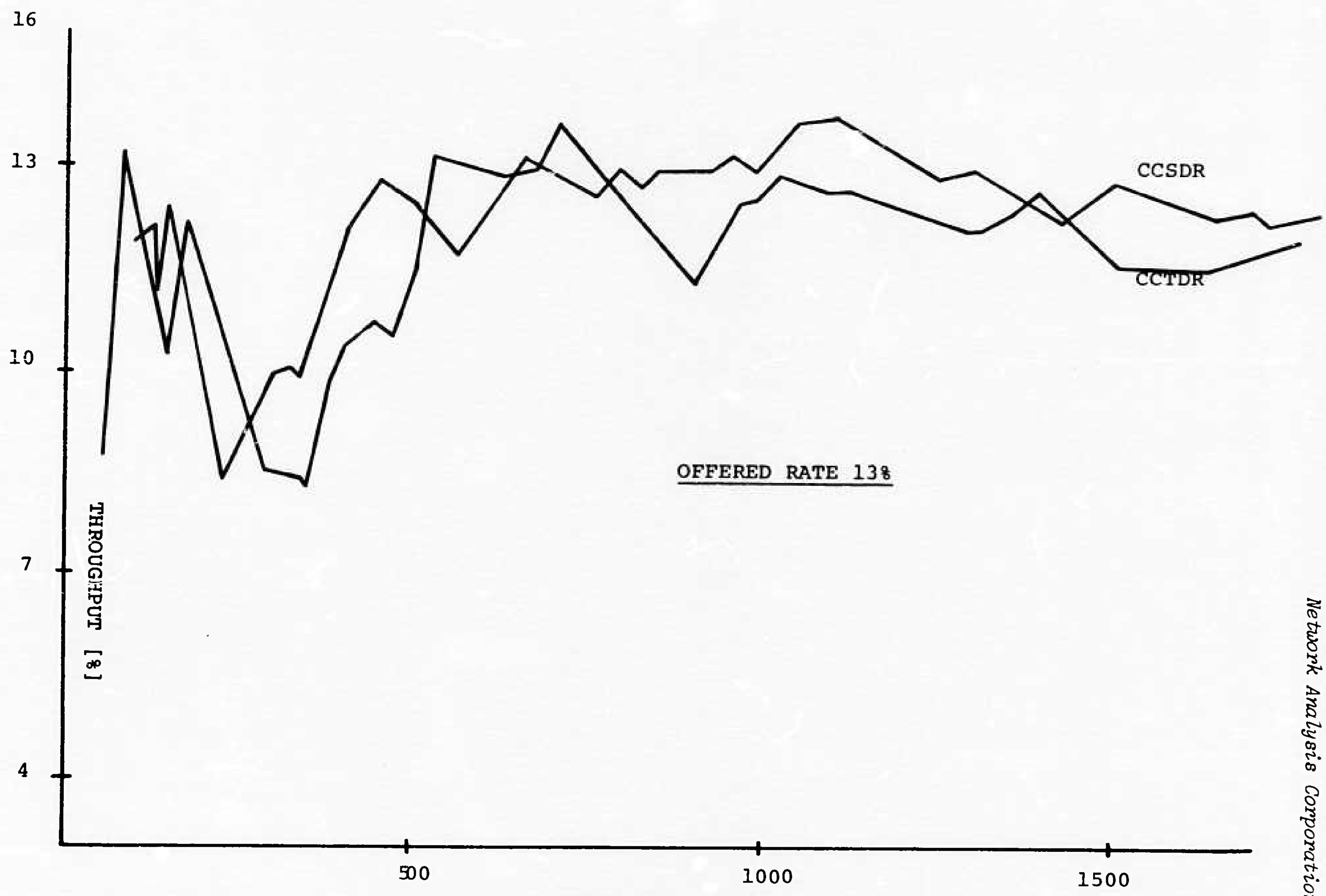


FIGURE 12

THROUGHPUT VS. TERMINAL SLOTS: 13% RATE

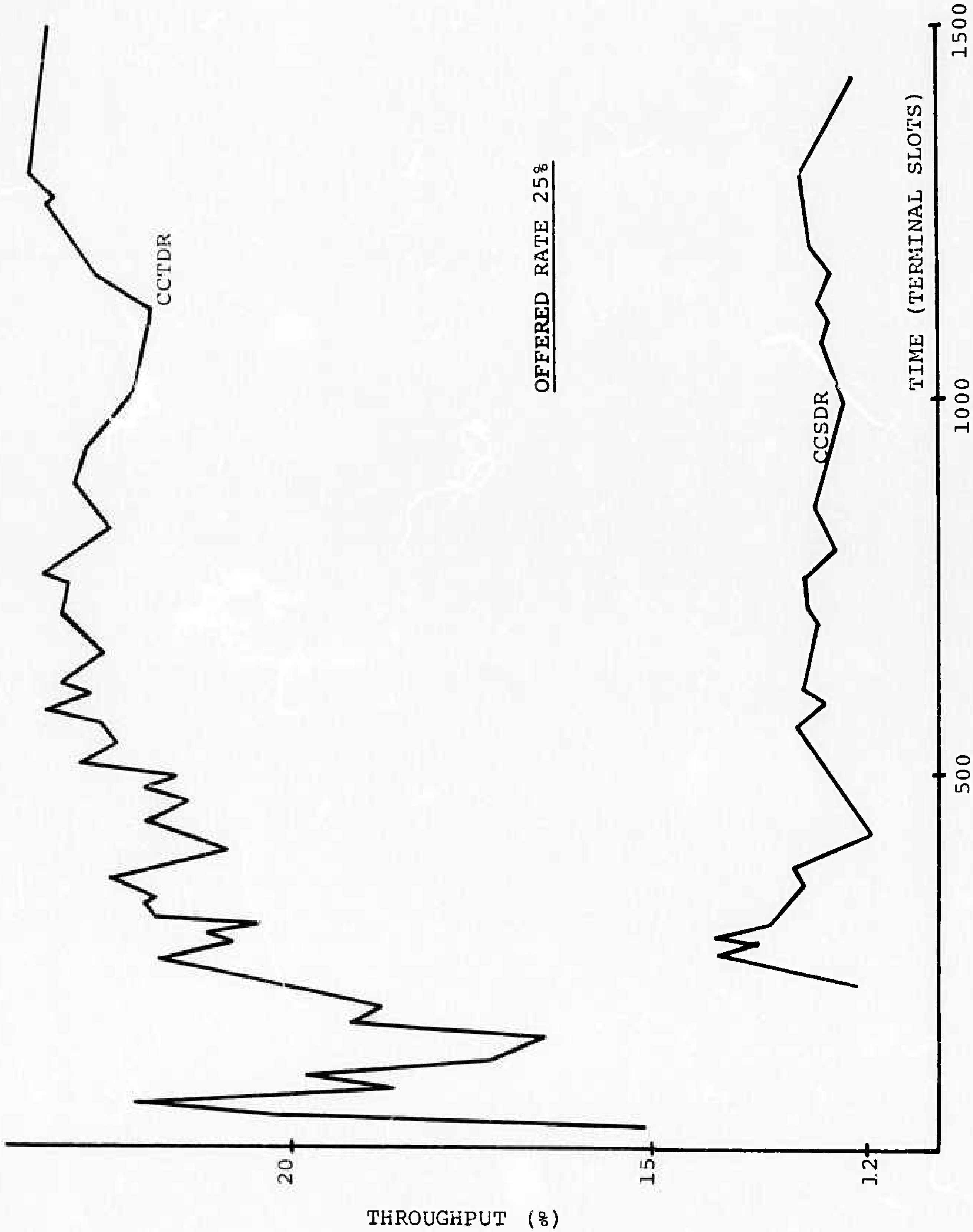


FIGURE 13

THROUGHPUT VS. TERMINAL SLOTS: 25% RATE

	OFFERED RATE [%]	THROUGHPUT [%]	DELAY OF IP TO STATION [Terminal Slots]	RATE OF STATION RESPONCE	PROB. STATION BUSY	% OF IP BLOCKED	TOTAL % OF IP LOSS	TERMINALS REMAINING
CCSDR	13	12.14	16.61	2.06	.50	10.63	11.41	10
	25	12.20	34.97	1.61	.48	29.50	32.95	23
CCTDR	13	12.39	4.91	1.99	.26	1.59	1.59	9
	25	23.33	11.51	1.97	.31	3.31	3.31	34

TABLE 2

XII. PRELIMINARY RESULTS OF MAXIMUM THROUGHPUT, LOSS, AND DELAY OF CCSDR AND CCTDR SYSTEMS

In the packet radio system there is an absolute maximum throughput (independent of loss and delay) because of the interference characteristics. Similar to curves of throughput versus channel traffic, when the relation is known analytically [7], we draw the curves of system throughput vs. offered rate for estimating the maximum throughput. Figure 14 shows the throughput versus offered rate for CCSDR and CCTDR systems. The curves are linear for low offered rates and saturate when the offered rate increases.

For the CCSDR system one can see that the throughput is practically the same when the offered rate is increased from 13% to 25%. This and the other measures (see Table 2), (for example, the rate of station response) show that the system is overloaded at a 25% offered rate. On the other hand, the system seems to operate at steady state at an offered rate of 13% (rate of station response 2.06). A rough estimate of maximum throughput for this system would be between 12% and 15%. Similar observations of the performance measures lead to an "estimate" of between 27% and 30% for the maximum throughput of the CCTDR system.

The average delay of the first Information Packet from terminal to station, and the Total Loss, as a function of offered rate are shown in Figure 15 and Figure 16, respectively.

Remark: There are many parameters in the simulation program which we have not experimented with (or tried to optimize) and which affect the quantities discussed above. One parameter which is significant in determining the maximum throughput is the average number of response packets from station to terminal. The affect of this parameter has been analyzed in [10] for a slotted ALOHA random access mode. It has been shown that the maximum throughput is increased in the Common Channel system when the rate of response increases, and the maximum throughput tends to 100% of the data rate when the rate of response tends to infinity. We expect that this parameter has a similar effect for the mode of access simulated. In the results reported here the rate of response is 2.0 which is small compared to usual estimates for terminals interacting with computers. Furthermore, the relatively short terminal interaction increases the traffic overhead of the search procedure, per information packet.

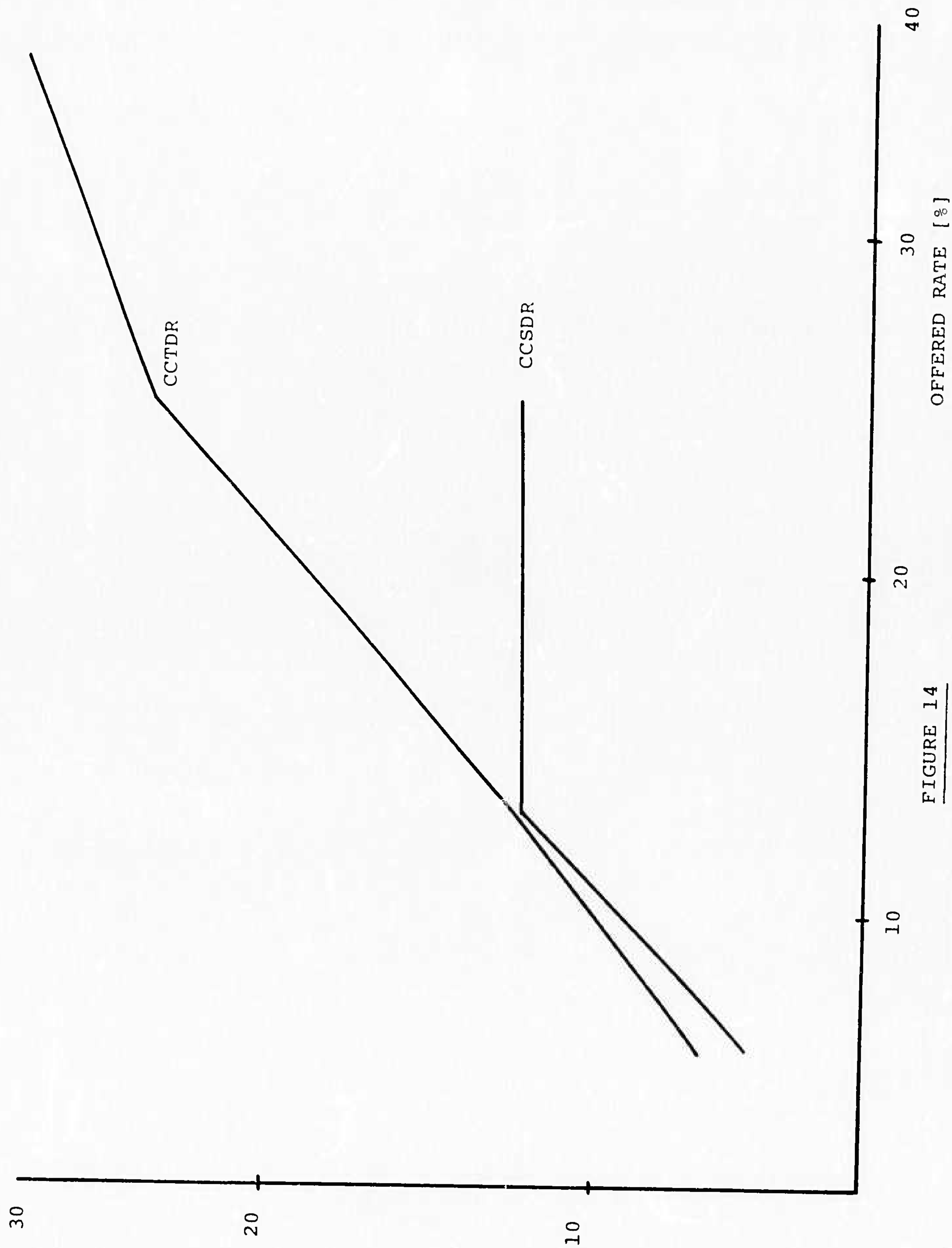


FIGURE 14

SYSTEM THROUGHPUT VS. OFFERED RATE

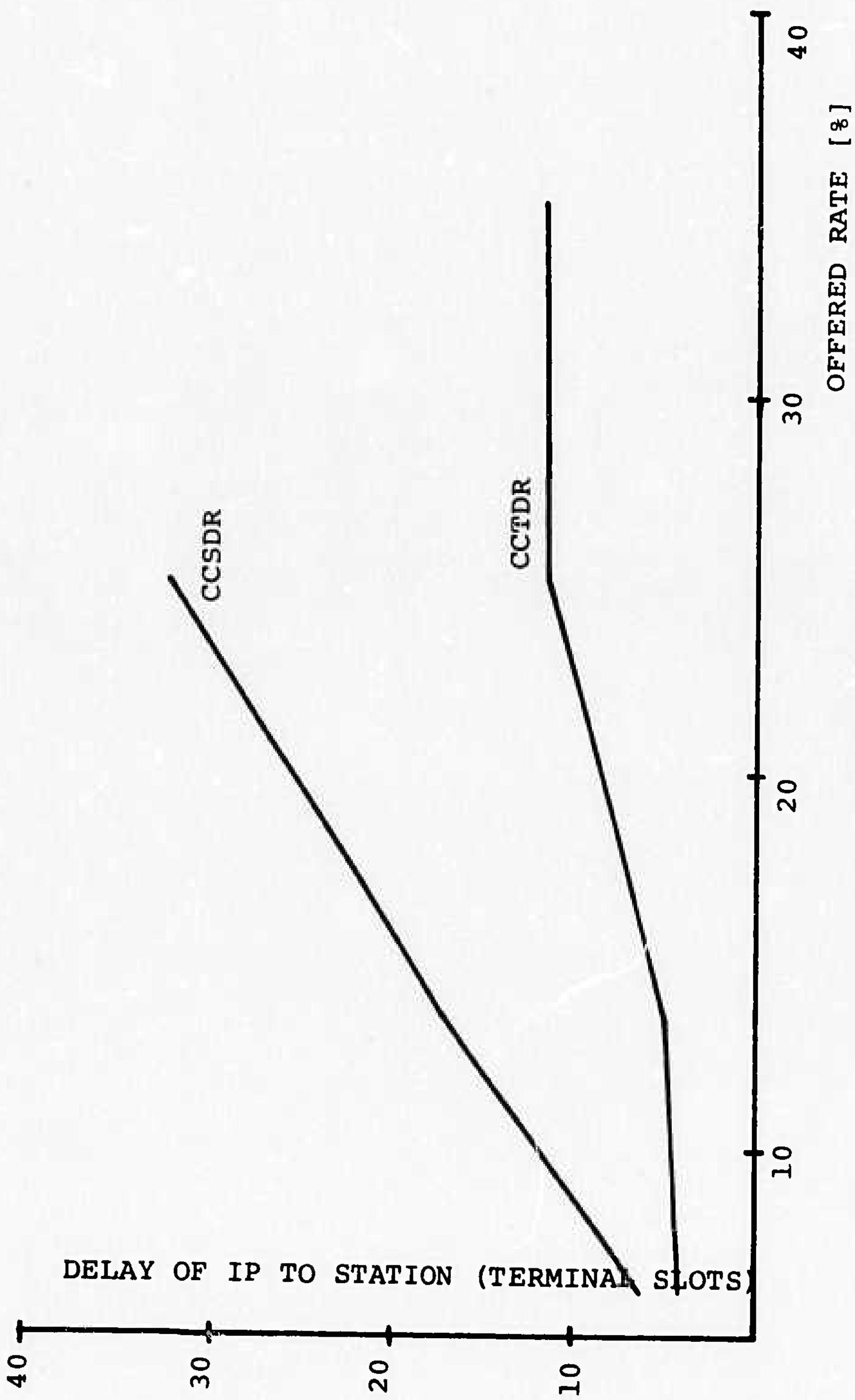


FIGURE 15

TERMINAL-STATION DELAY VS. OFFERED RATE

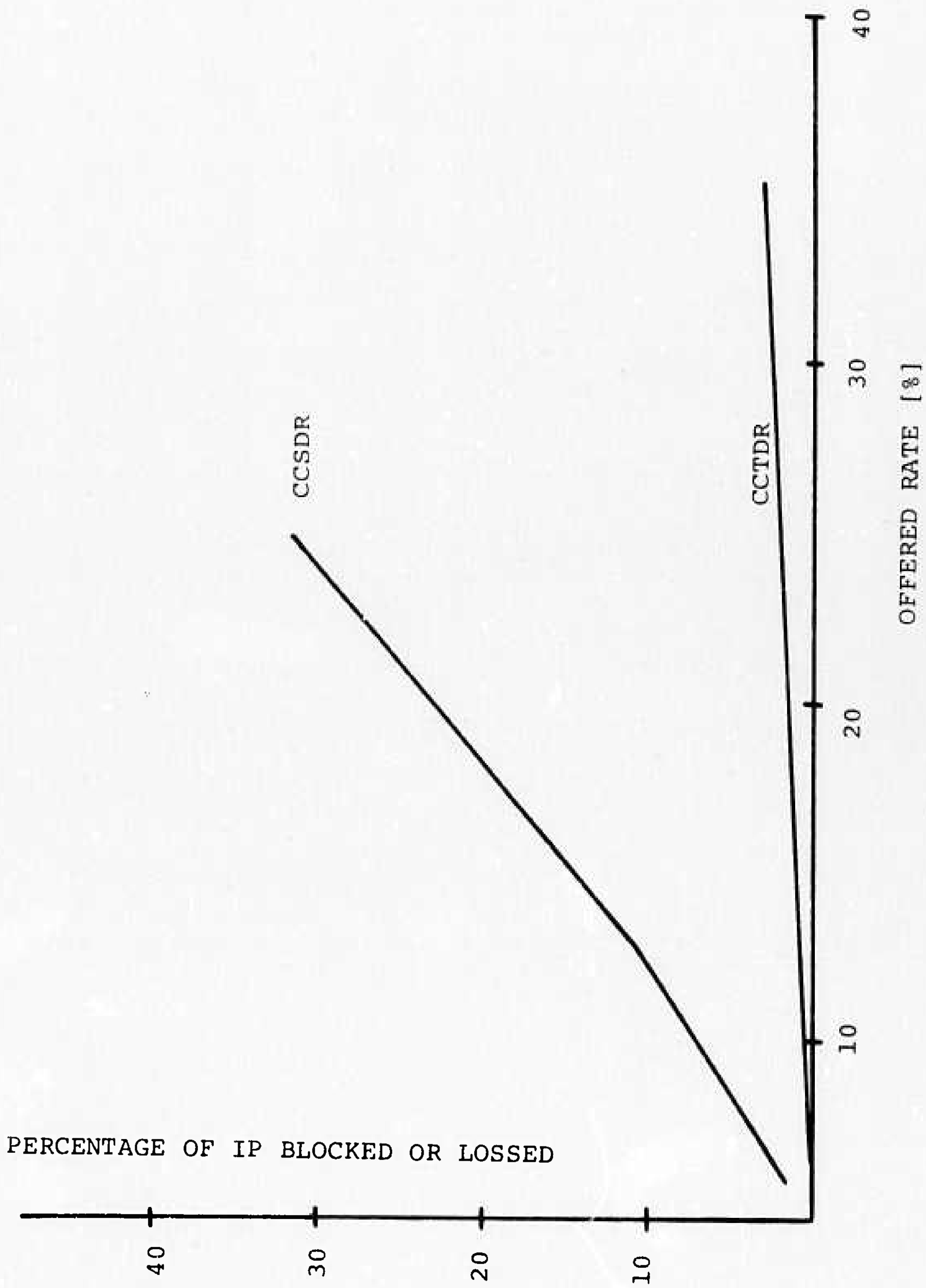


FIGURE 16

IP BLOCKING VS. OFFERED RATE

XIII. FUTURE DEVELOPMENT OF THE PACKET RADIO SIMULATOR

We outline several of the future developments for the simulator; some of these are in the implementation stage.

- <u>Initialization and Labelling of Repeaters</u>

Preliminary experiments have shown that the Hierarchical Labelling algorithm is much more efficient than the other two algorithms of Section VI, subsection B. Consequently, it is recommended for implementation in the Packet Radio System. In many cases however, the connectivity between devices in the network will not be known apriori. For example, in a military application one may wish to establish a network by distributing repeaters at random locations, and one may not have physical access to the repeaters. Furthermore, there may be changes in the "topology" of the network due to variations in transmission power of devices, or when some devices cease to operate. Thus, it is necessary to assign and reassign labels to repeaters in an operating network.

The approach that we adopted is to use the flooding routing algorithm to load repeaters with hierarchical labels. The flooding algorithm was selected because it does not require any knowledge of the topology of the network. A process for repeater initialization and labelling which has been detailed in [6] is currently under implementation. Initially, it is assumed that the station contains a set of fixed identifiers of repeaters which may possibly be connected into a network; three stages are then followed. In stage 1 the station transmits special control packets to the above repeaters, and repeaters respond with control packets from which a connectivity matrix between repeaters is established. The hierarchical labels are determined in stage 2 from the connectivity matrix. In stage 3 the station transmits the labels to repeaters and tests each path in both directions, from station to repeater and from repeater to station.

- Flow Control

Control packets for changing the operating parameters of devices, and algorithms for using these will be implemented. For example, turning repeaters "on" and "off", changing the parameter for the maximum number of transmissions, etc.

- Access Modes

In [7] it has been shown that one of the important parameters which affects the performance of the carrier sense access modes is the ratio of the propagation time between devices to the packet transmission time. Specifically, that the performance (relative) deteriorates when the above ratio increases; which is the case when the data signalling rate is increased and the number of bits in an information packet is kept constant. Thus for some operating parameters the carrier sense access modes may not show a much better performance than the more simple nonslotted ALOHA [11] random access scheme. These problems will be studied in a network environment by simulating the latter, for comparison, and by studying the carrier sense performance as a function of the data signalling rate.

- <u>Directional Antenna at the Station</u>

Analysis has shown [12] that directional antennas at the station may increase the system capacity. This can possibly be verified and quantified by simulating such an antenna.

- <u>Capture</u>

Currently the non-capture system is simulated. Caputre models which reflect the practical performance of hardware are under development and will be simulated.

REFERENCES

1. Kershenbaum,A. and R. VanSlyke, "Computing Minimum Spanning Trees Efficiently," Proceedings of the 1972 ACM Conference, Boston, August, 1972, pp. 518-527.

2. Knuth,D., Sorting and Searching: The Art of Computer Programming, Vol. 3, Addison-Wesley, 1973, Section 5.2.3.

3. Williams, J.W.J.,"Heapsort," Algorithm 232, ACM Collected Algorithms, June, 1964.

4. NAC (a): "Packet Radio System-Network Considerations" ARPA Network Information Center, Stanformd Research Institute, Melno Park, California.

5. NAC (b): "Routing and Acknowledgement Schemes for the Packet Radio System", ARPA Network Information Center, Stanford Research Institute, Melno Park, California.

6. NAC (c): "Packet Radio Broadcast Network-System Operation", ARPA Network Information Center, Stanford Research Institute, Melno Park, California.

7. Kleinrock,L. and F. Tobagi, "Carrier Sence Multiple Access (CSMA)", ARPA Network Information Center, Stanford Research Institute, Melno Park, California.

8. Van Slyke, R., W. Chou, and H. Frank, "Avoiding Simulation in Simulated Computer Communication Networks," National Computer Conference, 1973.

9. Fralick, Stanley, C., "R. F. Channel Capacity Considerations," ARPA Network Information Center, Stanford Research Institute.

10. Gitman, I., R. M. VanSlyke, H. Frank: "On Splitting Random Accessed Broadcast Communication Channels". Seventh Hawaii International Conference on System Sciences, January 8-10, 1974.

11. Abramson, N., "The ALOHA System - Another Alternative for Computer Communications," Fall Joint Comp. Conference AFIPS Conference Proceedings, Vol. 37, 1970, pp. 281-285.

12. Gitman, I., "On the Capacity of Slotted ALOHA Networks and Some Design Problems", ARPANET Network Information Center, Stanford Research Institute, Melno Park, California.

PACKET RADIO SYSTEM CONSIDERATIONS - NETWORK CAPACITY TRADEOFFS

I. INTRODUCTION

Packet Switching over radio channels with random access schemes is of current interest for local distribution systems and for satellite channels. This mode of operation is useful when the communicating devices are mobile and when the ratio of the peak to average data rate requirement of each device is high. Such systems have been analyzed for the case in which all communicating devices are within an effective transmission range of each other; either directly or through the satellite. These analyses were originally done for the ALOHA system [1] and for a satellite channel [4]. The models used do not sufficiently describe the Packet Radio System. The reason being that in the packet radio system there is a network of repeaters which separate an originating device from a destination device (terminals and stations).

In this chapter, we address broadcast networks in which orginating devices cannot directly reach the destination receiver. Thus, repeaters are introduced which receive these packets and repeat them to the destination. The capacity (maximum throughput) of such systems is determined, and design problems related to the number of repeating devices and the usefulness of directional antennas are resolved.

The model used in this chapter can describe systems other than the packet radio system and we discuss it in the more general context.

One way to categorize channel allocation schemes for data transmission is the following:

1. Fixed assignment (FDM, TDMA)

2. Dynamic assignment with centralized control (polling schemes, reservation upon request)

3. Dynamic assignment with distributed control (loop networks, random reservation schemes)

4. No assignment (random access schemes)

It has been recognized that a fixed allocation of channel capacity is extremely wasteful when the traffic of users is of a bursty nature; that is, when the traffic requirements of the users can be characterized as having a high peak to average data rate. Users can be so characterized in an inquiry response application. In fact, if one characterizes a set of users by the number and by the ratio of the peak to average data rate requirements of each user, one can conjecture that when the above number and ratio are increasing, one obtains a higher channel utilization when proceeding (along the categorization) from the fixed assignment to the no assignment allocation schemes. For example, when the time delay to make a reservation or the average time between two consequtive pollings of a user is large compared to the fraction of time that the user wishes to use the channel, then the dynamic assignment with centralized control, becomes inefficient (apart from the need for a system for polling or making reservations).

Roberts [7] has demonstrated the cost advantages of a random reservation scheme and a random access scheme over fixed assignment, when the number of users increases and the average traffic requirement per user is kept constant. A somewhat more formal justification for sharing a channel was given by Kleinrock and Lam [4], we quote: "Rather than provide channels on a user-pair basis, we much prefer to provide a single high-speed channel to a large number of users which can be shared in some fashion; this when allows us to take advantage of the powerful 'large number laws' which state that with very high probability, the demand at any instant will be approximately equal to the sum of the average demands of that population." Gitman, Van Slyke, and Frank [3], have addressed the problem of splitting a channel between two classes of users. It was shown that in almost all cases, sharing the channel results in a higher utilization of the total channel capacity.

In this chapter, we consider a packet switching network in which a single radio channel is shared by all communicating devices. Devices access the channel using the so called "slotted ALOHA" random access scheme. When a random access scheme is used, there is a possibility that more than one packet is simultaneously received by a receiver due to independent transmissions of several devices. In that event, it is assumed that none of the packets are correctly received, and the corresponding devices have to retransmit their packets. One can see that the number of packets transmitted is larger than the number of originating packets. That is, part of the channel capacity is used up by the wasteful collisions and are not considered as effective channel utilization since it does not contribute to the throughput. For example, if each packet is transmitted two times, on the average, before it is successfully received, then the maximum effective utilization of the channel (or the effective channel capacity) is one-half of the given capacity, the other one-half is used up by the non-successful transmissions. The first problem that one faces is to determine the maximum effective utilization (or system capacity) that can be obtained. This is one of the problems addressed in the chapter.

If the channel is offered a higher rate of traffic than its effective capacity, the system becomes unstable in the sense that the number of transmissions increases with time and the throughput decreases with time until zero throughput is obtained. This implies another problem in random access schemes and that is, the control of the offered rate and retransmission strategies so as to obtain the highest possible channel utilization [5], [8]. It is clear however, that retransmissions have to be randomized in time since otherwise once a collision between two devices occurred, it will persist.

The first packet radio channel system where devices use a random access scheme, known to us, was analyzed and implemented at the University of Hawaii [1]. The random access scheme used is the so called "pure" or "unslotted" ALOHA. In this scheme, every terminal transmits its packets independent of any other terminal or any specific time. That is, the terminal transmits the whole packet at a random point in time; the terminal then times out for receiving an acknowledgement. If an acknowledgement is not received, it is assumed that a collision occurred and the packet is retransmitted after an additional random waiting time. Abramson [1] obtained that the capacity (effective) of the channel is 1/2e of the given capacity when the number of terminals is very large and when the point process of the beginning of packet transmission onto the channel is Poisson.

It was realized that a gain in capacity can be obtained if the channel was slotted into segments of time whose duration is equal to the packet transmission time, and when terminals are required to begin the transmission at the beginning of a time slot. The access scheme is random in the sense that terminals transmit into a random slot in time and retransmit after waiting a random number of slots. This scheme is called "slotted ALOHA." Roberts [6] has shown that the capacity of this scheme is 1/e of the given capacity, using the same assumptions as Abramson. The slotted ALOHA random access scheme was further analyzed in [2], [3], [4], and [5] for the case of a small number of terminals and when there is a mixture of traffic from a small number of "big" users and a large number of "small" users.

II. PROBLEM DESCRIPTION

In a network context, the analyses of all the references address a "single hop network." That is, when considering a terrestrial system, it is implicitly assumed that all devices are within an effective distance of each other; the same is true when considering a satellite channel, since the satellite echoes the packets to the destination station. Such a network can be described as one in which there is a single receiver and many transmitters, all of which are within an effective transmission range to the receiver, and to each other.

In cases in which the transmission range of terminals is not sufficient to reach the destination receiver, it is necessary to introduce another device which will receive the packets from the terminals and repeat them to the final destination. Such a network can be used for local distribution and collection of traffic, in which case the station is a gateway to a point-to-point network. We particularly consider a 2-Hop network model in which there is a large number of terminals in the neighborhood of each repeater and that the transmission range of terminals is short so that a terminal can reach only one repeater, as shown in Figure 1. Our model can be useful as a distribution model for a suburban area, where instead of supplying each terminal with a powerful transmitter, we allocate repeaters which collect the traffic from terminals.

In a military application, one can think of a large unit which uses such a 2-Hop network, where each subunit (the terminals of which are relatively close to each other and may be mobile) has its own repeater, and the station is at the Headquarters. In fact, the network can be operative when the whole unit is moving, providing each subunit carries its repeater or station along with it, e.g., a fleet of ships.

This network can also model a satellite case as shown in Figure 2. In the analyses of a satellite channel reported, it was assumed that each terminal is a ground station which originates its traffic or which is connected to a point-to-point network. However, suppose there are clusters of (possibly mobile) terminals, where in each cluster, they are relatively close to each other, and wish to communicate with a remote central computer installation. Then, one can devise a Ground Station Repeater to communicate with the terminals and the ground station repeater may use a satellite channel to transmit (second hop) to the computer installation.

Before we specify the problems to be addressed, we comment on the operation of the repeater. We have indicated that a terminal retransmits its packet after a time out if it does not receive an acknowledgement. The repeater can operate in a very simple manner in which it repeats only once a packet that it correctly receives. In this case, the terminal has to time out for a longer period of time to wait for an acknowledtement from the station (end-to-end ack). Alternatively, the repeater can be made responsible for the successful transmission on the second hop, by acknowledging the terminal, storing the packet, and transmitting it until it receives an acknowledgement from the station (hop-by-hop acknowledgement). This problem has been considered in [9] where it was shown that a hop-by-hop acknowledgement operation is more efficient. Thus, we will assume a repeater of that type.

The first problem that we address is to determine the network capacity as a function of the number of repeaters and the interference between repeaters. Another problem of interest, is to determine the capacity bottleneck (critical hop); i.e., whether the capacity bottleneck is on the hop from terminals to repeaters or on the hop from repeaters to station. The system is assumed to have two channels; one channel is used for transmission from terminals via repeaters to the station and the second channel for transmission from the station via repeaters to the terminals. The traffic of acknowledgement packets is not considered; which implicitly assumes that there is a separate channel for acknowledgements. The two channels are analyzed separately and design questions such as the following are considered: Is it useful to have directional antennas at repeaters when transmitting to the station ? This question is relevant in the terrestrial system, since in the satellite system, the ground station repeaters will presumably be out of range from each other, and use directional antennas. Similarly, in a terrestrial system, one may ask about the usefulness of using a directional antenna at the station when transmitting to repeaters. Other questions relate to the possibility of using several transmitters and antennas at the station when transmitting to repeaters.

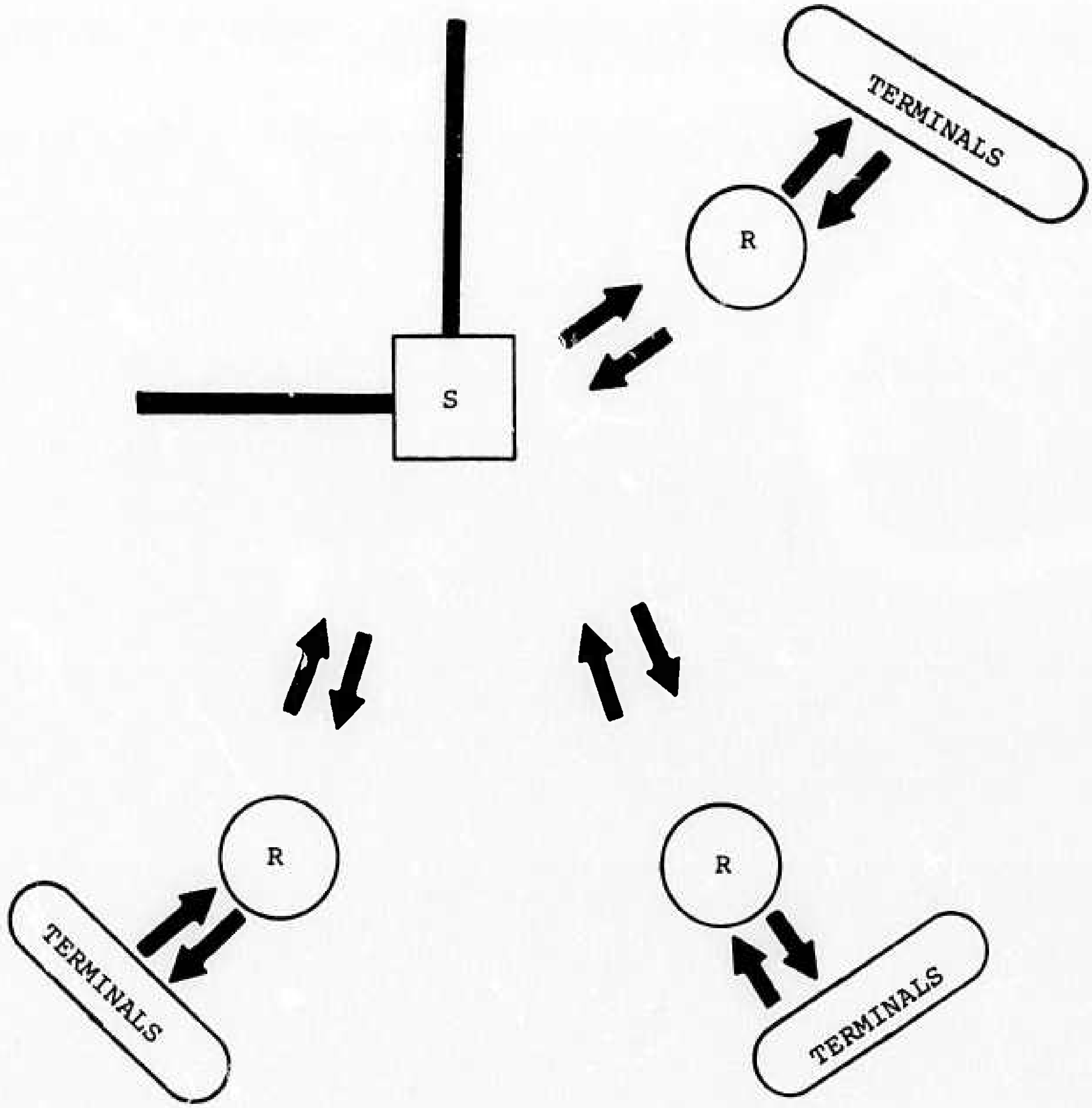


FIGURE 1: A TERRESTRIAL SYSTEM

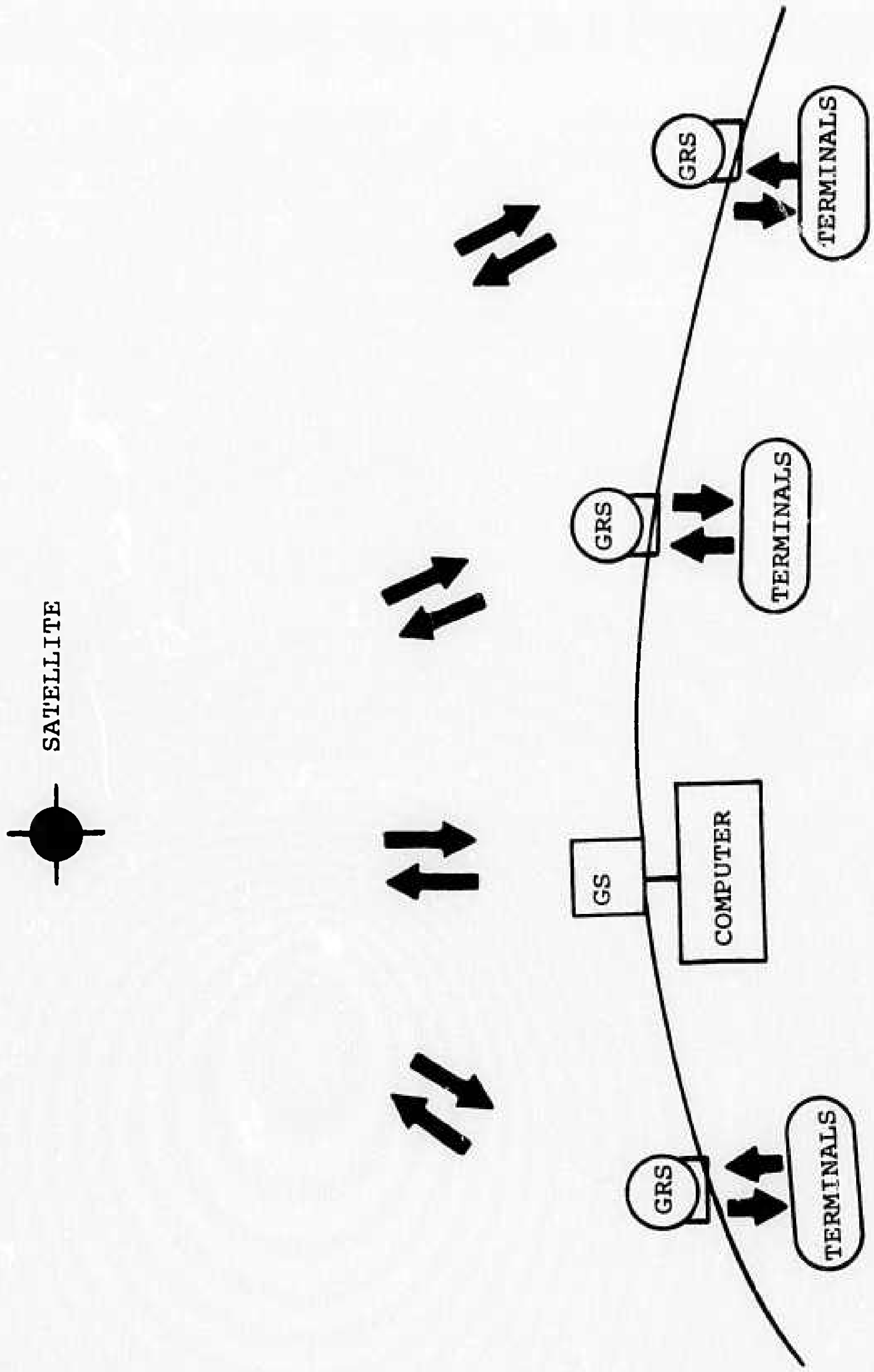


FIGURE 2: A SATELLITE SYSTEM

III. TRANSMISSION FROM TERMINALS TO STATION

Consider a system where m repeaters receive packets from terminals and repeat the packets to a single station, as shown in Figure 3. We donote by G and S the rate of packet transmission per slot and the rate of successful packet transmission per slot, respectively. Specifically, let G_{1i} and S_{1i} be the rates of transmission from terminals to repeater i, and G_{2i} and S_{2i} the rates from repeater i to the station. We wish to obtain the probability that a repeater is idle.

A single hop network is the case in which a set of terminals transmit to a repeater and the repeater is the final destination and does not repeat the packets. Thus, the probability that a transmission from a terminal to the repeater is successful is the probability that no other terminal transmits into that same slot. That is, if we assume that a packet is transmitted into the first slot after it becomes ready for transmission then the probability of success is the probability that no new packet has arrived and that no other packet has been scheduled for retransmission in the interval of time of the preceeding slot. In the network case however, a transmission from a terminal to a repeater will not be successful also in the case when the repeater uses the same slot for transmitting to the station, or if another repeater, within an effective transmission range of the first and which uses an omnidirectional antenna, uses the same slot for transmission to the station.

Throughout the chapter, we use the following assumptions. The combined point process of packet origination and packets scheduled for retransmission, from each set of terminals to a repeater, is Poisson. Thus, the probability of no arrival during a slot time τ is $e^{-G_{1i}\tau}$; we use $\tau \equiv 1$. The probabilities of transmission by a repeater into different slots are independent. The probability of

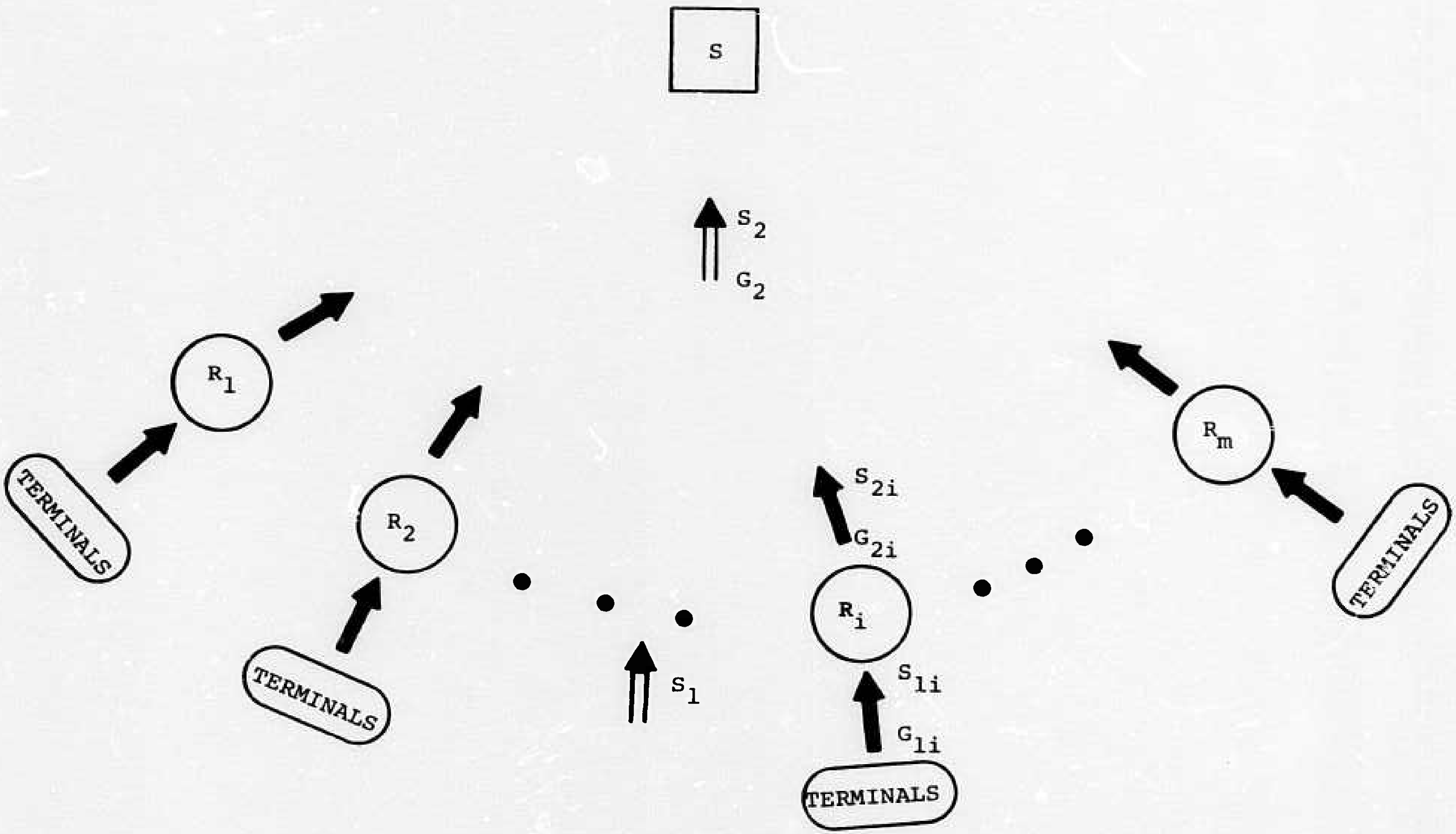


FIGURE 3: TRANSMISSION FROM TERMINALS TO STATION

transmission by two or more repeaters into a randomly chosen slot are mutually independent; and the probability of transmission into a random slot by a terminal and by a repeater are independent. The assumption of a Poisson distribution for the process of packet origination plus retransmission has been questioned in previous publications. The validity of this assumption is important when one considers packet delays, since if one assumes a Poisson point process for packet originations, then the assumption that the combined process, of origination plus retransmissions, is Poisson is valid only when one allows very large packet delays. In this chapter, we do not require finite delays and our interest is in the ultimate system capacity.

Finally, let Q_i denote the set of repeaters which have an effective transmission range to repeater i; Q_i includes repeater i. Under these assumptions we can write:

$$P_r[\text{repeater } i \text{ is idle}] = e^{-G_{1i}} \prod_{j \varepsilon Q_i} (1-G_{2j}) \qquad (1)$$

Similarly, the probability that the station is idle can be written as:

$$P_r\ [\text{station is idle}] = \prod_{j=1}^{m} (1-G_{2j}) \qquad (2)$$

We now make a few assumptions which simplity the computation but enable us to answer the questions of interest. Specifically, we assume that $G_{1i} = G_1$, $S_{1i} = S_1/m$ for all i, that repeaters share equally the load so that $G_{2i} = G_2/m$ and $S_{2i} = S_2/m$ for all i, and that all repeaters have the same number of interfering repeaters, $n\ (Q_i) = I$ for all i. We refer to I as the interference level.

We can now write the throughput equations for hop 1 and hop 2:

$$S_1 = \sum_1^m G_1 \left(1-\frac{G_2}{m}\right)^I e^{-G_1} = mG_1 \left(1-\frac{G_2}{m}\right)^I e^{-G_1} \quad (3)$$

$$S_2 = \sum_1^m \frac{G_2}{m} \left(1-\frac{G_2}{m}\right)^{m-1} = G_2 \left(1-\frac{G_2}{m}\right)^{m-1} \quad (4)$$

Note that when m=1 all transmissions from the repeater to the station are successful since it does not interfere with its own transmissions.

If a repeater were a traffic source and a traffic sink, then G_1 and G_2 could have been considered as independent variables. In our case, however, the intensity of the processes on the two hops are related. In particular, we consider traffic rates in which the system can operate at steady state. That is, if one observes the system for a long period of time, then all the packets which successfully arrive to repeaters also successfully arrive to the station. Thus we can use the conservation law at repeaters, namely $S_1 = S_2$. This results in the relation:

$$G_2 = mG_1 \left(1-\frac{G_2}{m}\right)^{I+1-m} e^{-G_1} \quad (5)$$

One can now study the system performance as a function of parameters m and I, with one independent variable.

A. <u>Complete Interference System, I = m</u>

This case is applicable to terrestrial networks in which repeaters are either placed relatively close to each other or use powerful transmitters; either of which results in the interference among all repeaters. For this case we obtain:

$$G_2 = \frac{m\, G_1 e^{-G_1}}{1+G_1 e^{-G_1}} \quad (6)$$

and

$$S_1 = \frac{m\, G_1\, e^{-G_1}}{(1+G_1\, e^{-G_1})^m} \tag{7}$$

The capacity of this system is given by the maximum of S_1.

$$\frac{d\, S_1}{d\, G_1} = \frac{m\, e^{-G_1}}{(1+G_1\, e^{-G_1})^{m+1}}(1-G_1)\ [1-(m-1)\ G_1\ e^{-G_1}] = 0 \tag{8}$$

By examining (8), one finds that there is one stationary point, a maximum at $G_1 = 1$, when $m<4$. When $m\geq 4$ there are three stationary points; a minimum at $G_1 = 1$, and two maximum points of the same value at G which satisfies $1-(m-1)\ G\ e^{-G} = 0$. Substituting these values into (7), one obtains the capacity of this (complete interference) network as a function of m:

$$\text{Network Capacity} = S_1^* = \begin{cases} \dfrac{m}{e\ (1+\frac{1}{e})^m} & m<4 \\ (1-\frac{1}{m})^{m-1} & m\geq 4 \end{cases} \tag{9}$$

Notice that the network capacity is lower than the capacity of a single hop network, 1/e, when m=1; it is higher than 1/e for m>1, and tends to 1/e in the limit when $m\to\infty$.

Figure 4 shows the network throughput S_1 as a function of the rate of transmission from terminals to a single repeater. It is interesting to observe the rate of change of S with respect to G. For example, when M is large (e.g. m=8) then the system becomes more sensitive to variations in the value of G. On the other hand, there are values of m, for example, m = 2 or 3 in the complete interference case, for which the maximum throughput region is quite flat.

B. The Critical Hop

To determine the critical hop, one has to obtain the capacity of the two hops of the network. The capacity of the hop from repeaters to the station is independent of G_1 and I (see Eq. (4)), and is given by $(1-1/m)^{m-1}$. The capacity of the hop from terminals to repeaters, on the other hand, depends on G_2 and I. Note that $0 \leq G_2 \leq m$; $G_2 > m$ is not realizable since we assume that a repeater has one transmitter and cannot transmit more than one packet per slot. For any G_2 in the above range, the capacity of hop 1 (see Eq. (3)) increases with m.

Thus, there exists an m_o for which the capacity is higher than that on the hop from repeaters to station, and the latter becomes the critical hop. Furthermore, m_o depends on I, and for $I_2 \geq I_1$, $m_o (I_2) \geq m_o (I_1)$. Thus, for $m > m_o$ the critical hop is that from repeaters to station, and for $m \leq m_o$ the critical hop is from terminals to repeaters. For example, for I=m (see Eq. (9)) $m_o = 4$, and for $I = m-1$, $m_o = 3$.

Perhaps a more direct way to answer the question of the critical hop is to obtain the system capacity as a function of m and I. Then, whenever the capacity is smaller than $(1-1/m)^{m-1}$ the critical hop is that from terminals to repeaters and when the capacity equals this expression, the critical hop is from repeaters to station. However, it is difficult to obtain a closed form solution in the general case.

C. Number of Repeaters and Directional Antennas at Repeaters

The considerations of directional antennas at repeaters apply only to the terrestrial system. In the satellite system, ground station repeaters will use directional antennas when transmitting to the satellite and omnidirectional antennas when communicating with terminals.

The effect of directional antennas at repeaters in the terrestrial system is that the transmission from repeaters to the station is directed towards the station and does not interfere with the transmission of terminals to other repeaters. Thus, it is the special case with I=1. We notice, however, that directional antennas do not increase the capacity of the hop from repeaters to station because all antennas are directed towards the same physical location where the station is placed and where the conflicts may occur.

Figure 5 shows the capacity of the system as a function of m, for I=m and I=1, which is equivalent to omnidirectional and directional antennas (or satellite system) respectively. One can see that there is a gain in capacity when using directional antennas only when m=2, and a small gain for m=3. For $m \geq 4$ the capacity does not increase because the critical hop is between the repeaters and the station, so that it does not matter how much one can get through from terminals to repeaters.

As far as the number of repeaters is concerned, one can see from Figure 5 that the maximum system capacity is obtained when m=2 in the non-interference case and when m=3 in the complete interference case. Thus 2 or 3 repeaters would be a good design; any additional repeaters that may be added because of other considerations (such as area coverage) will result in a reduction in the system capacity.

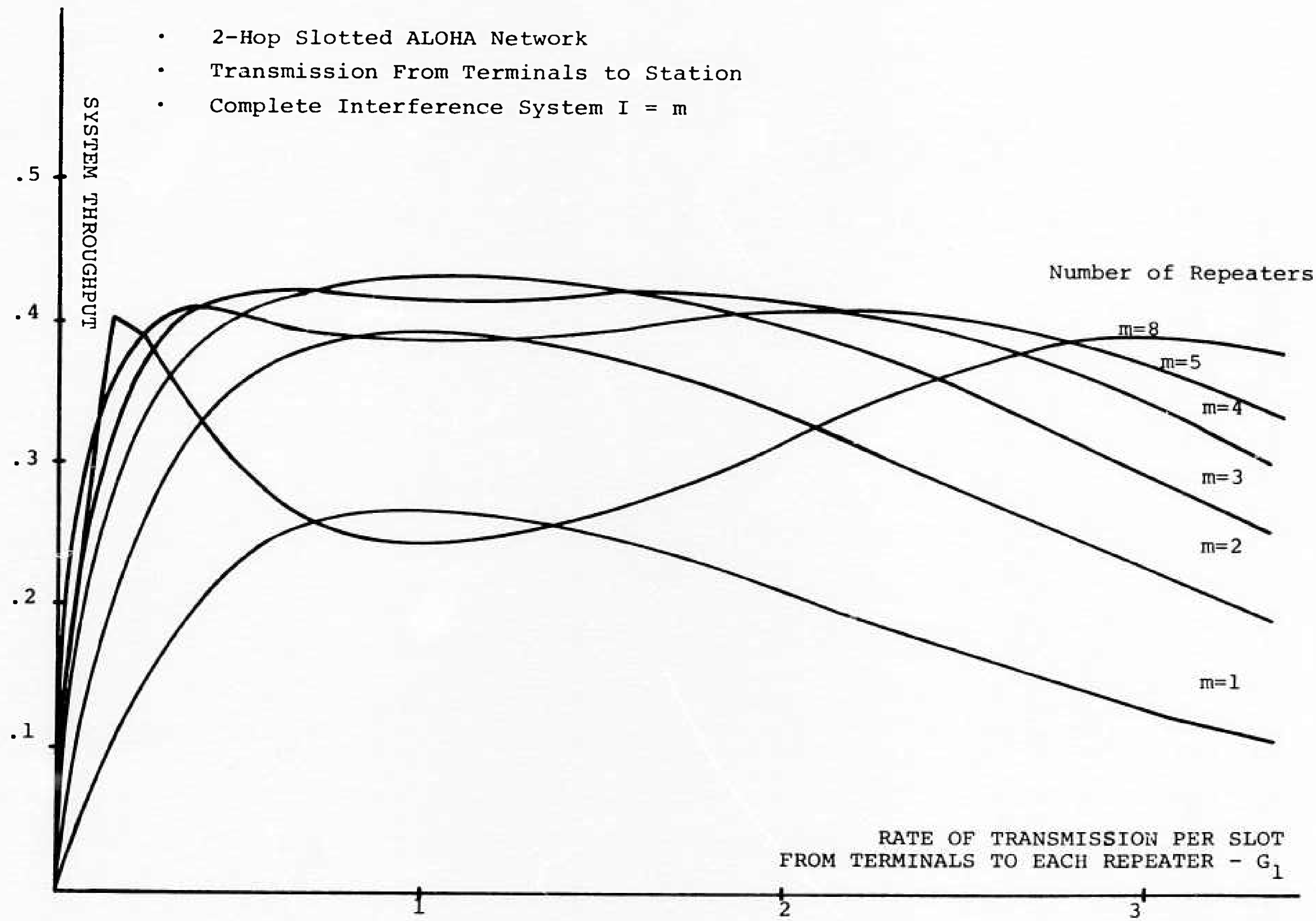


FIGURE 4

NETWORK THROUGHPUT VS. TERMINAL-REPEATER TRANSMISSION RATE

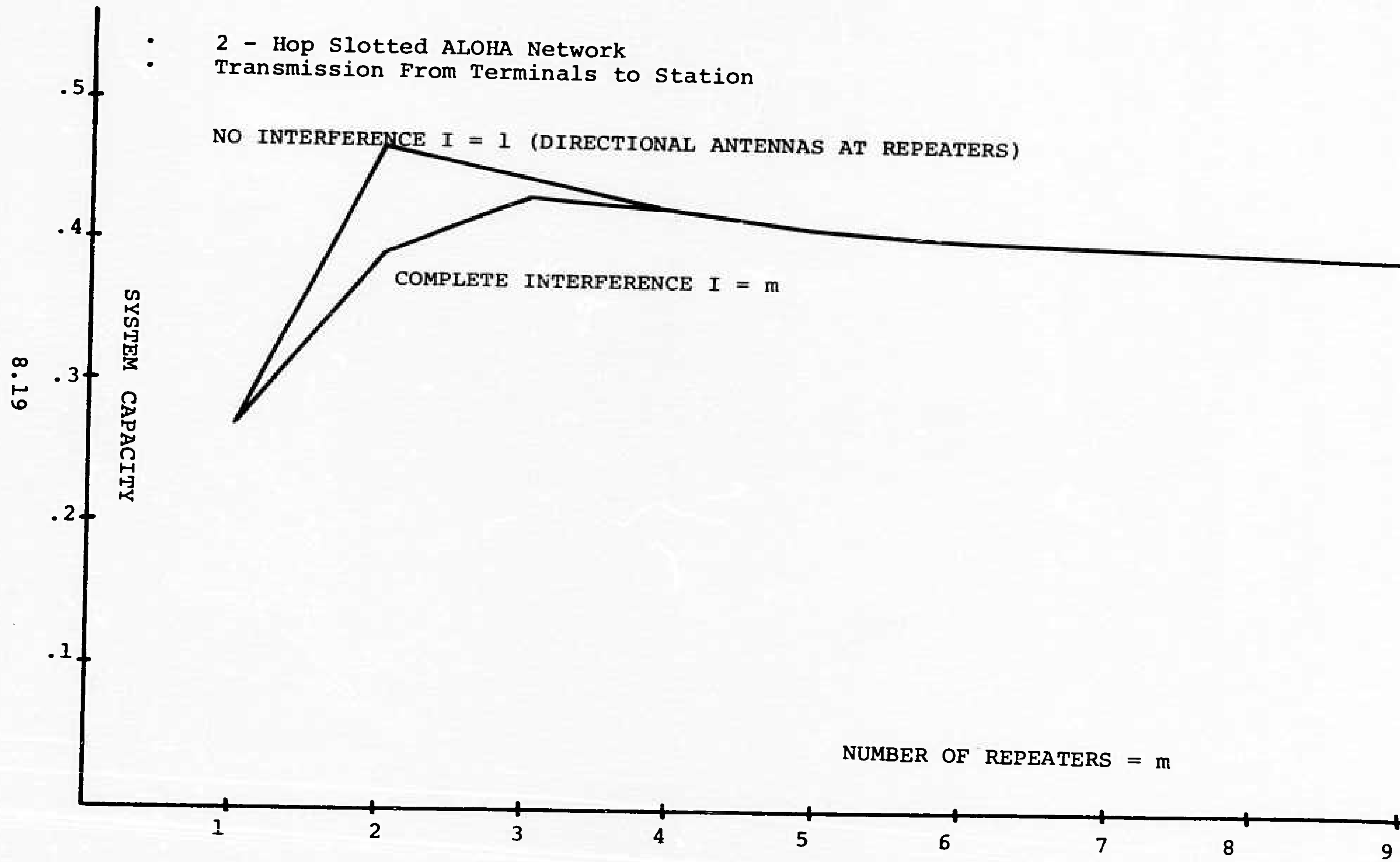


FIGURE 5

NETWORK THROUGHPUT VS. NO. OF REPEATERS: DIRECTIONAL AND NON-DIRECTIONAL ANTENNAS

IV. TRANSMISSION FROM STATION TO TERMINALS

In this section, we consider the second channel which is used for transmission from the station to terminals via repeaters. In the terrestrial system, it is assumed that the effective transmission range of the station is such that it interferes with the transmission from repeaters to terminals, as shown in Figure 6. However, we assume that terminals cannot directly receive from the station or from the satellite (otherwise, it becomes a single hop network and the capacity is 1). We use the notation shown in Figure 6, where the first hop is that from the station to repeaters and the second hop is from repeaters to terminals.

We use similar assumptions to the ones made in the previous section. Specifically, we assume that the probabilities of transmission by a repeater into different time slots are independent, that the probability of transmission of two or more repeaters into a randomly chosen slot are mutually independent, and that the probability of transmission by the station and by repeaters into a random slot are mutually independent. Finally, we simplify by assuming that repeaters share equally the load; by which we mean that $S_{2i} = S_2/m$ and $G_{2i} = G_2/m$, for all i. The equations which relate the rate of transmission to the rate of successful transmission on the two hops can now be written:

$$S_2 = \sum_{1}^{m} \frac{G_2}{m} (1 - G_1) (1 - \frac{G_2}{m})^{I - 1} = G_2 (1 - G_1) (1 - \frac{G_2}{m})^{I - 1} \quad (10)$$

$$S_1 = G_1 (1 - \frac{G_2}{m})^{I} \quad (11)$$

I is the interference level as in the previous section.

For consistency with the interference model of the previous section, we remark the following. Eqs. (10) and (11) are for the case in which the same energy-per-bit-to-noise-density is required

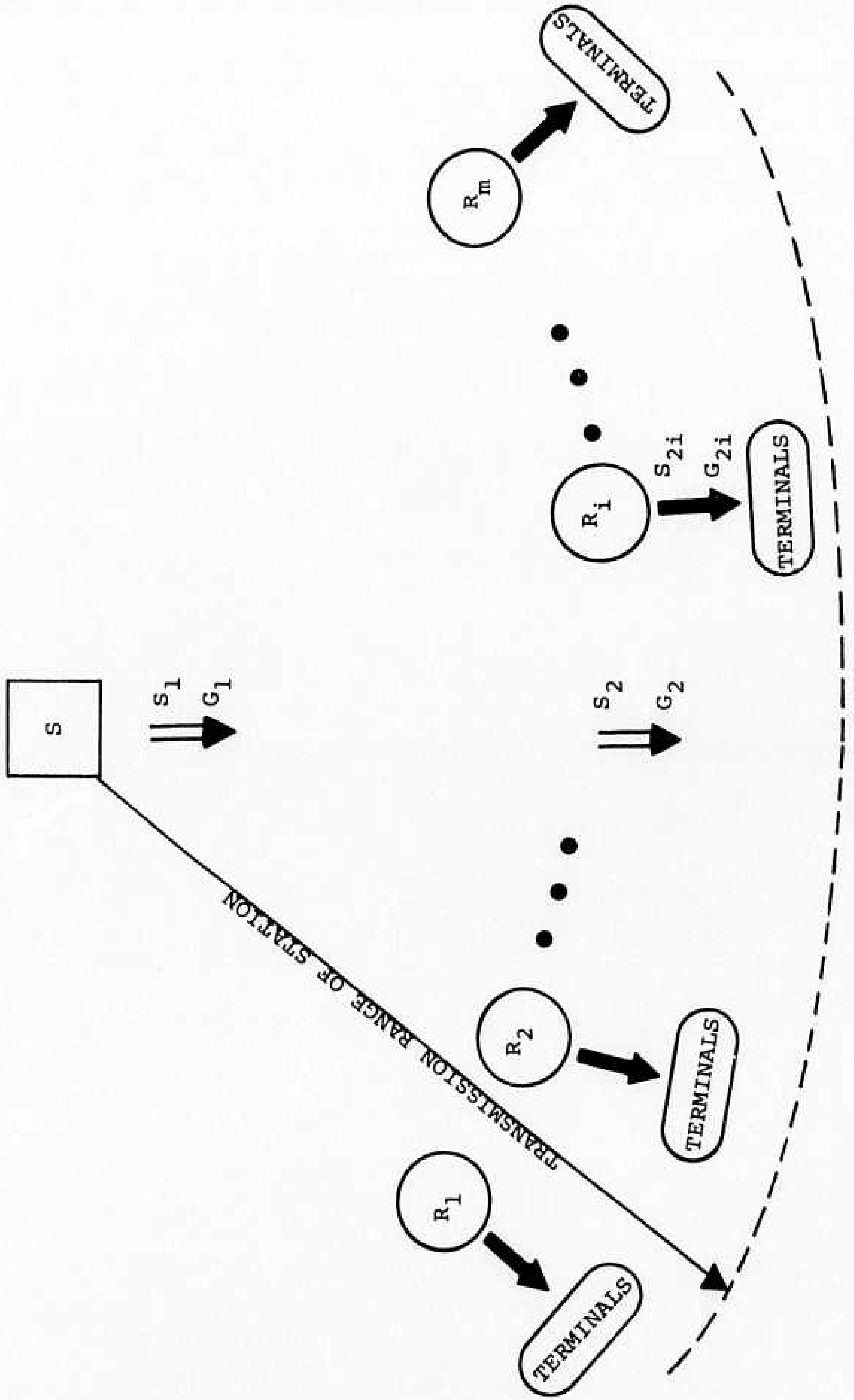


FIGURE 6: TRANSMISSION FROM STATION TO TERMINALS

for detection with equal error rates, by the repeater and by the terminal; and when the repeater used a higher transmitter power. In the more general case, one has to replace I in (10) by I_T which will designate the number of sets of terminals with which a transmission from a repeater may interfere; and replace I in (11) by I_R which will indicate the number of repeaters with which a transmission from a repeater may interfere.

We assume again steady state system operation and use the conservation law at repeaters, i.e., $S_1 = S_2$. This results in

$$G_2 = \frac{G_1}{1 - G_1 \frac{m-1}{m}} \tag{12}$$

One can now substitute (12) into (10) and (11) to obtain the throughput as a function of one independent variable. For S_1, one obtains:

$$S_1 = G_1 \left(\frac{1 - G_1}{1 - G_1 \frac{m-1}{m}} \right)^I \tag{13}$$

To obtain the capacity one can maximize either S_1 or S_2. Equating dS_1/dG_1 to zero, we obtain that there are three stationary points. A minimum at $G_1^1 = 1$ and two maximum values at

$$G_1^{2,3} = \frac{(2m + I - 1) \pm \sqrt{4mI + (I-1)^2}}{2\,(m-1)} \tag{14}$$

We now examine the constraints. For the system to be realizable,

$$0 \le G_1 \le 1 \; ; \; 0 \le G_2 \le m \; ; \; 1 \le I \le m \tag{15}$$

From Equation (14), one can see that the plus sign results in $G_1 > 1$, since

$$2m + I - 1 + \sqrt{4mI + (I-1)^2} > 2m + 2(I-1) > 2(m-1) \tag{16}$$

Furthermore, from Equation (12), one can see that when $0 \le G_1 \le 1$, then $0 \le G_2 \le m$. Thus, the only realizable maximum is given by Equation (13) with G_1 as in Equation (14) when taking the minus sign of the square root.

Figure 7 shows the system throughput as a function of G_2 for m = 6 and I as a parameter. One can see that there is a high degradation in system performance when the interference level I increases. Figure 8 shows the capacity of the system as a function of the number of repeaters m for the non-interference (I = 1) and the complete interference (I = m) cases. When $1 < I < m$, the curve will be between the two shown. It can be seen that there is a large difference in system capacity between the non-interference and the complete interference systems, and that this difference increases with the number of repeaters m. For I = 1, the capacity of the system is $(m(\sqrt{m} - 1)^2)/(m - 1)^2$ which tends to 1 when m tends to infinity.

In the terrestrial system, the interference level depends on the transmission power of the repeaters when transmitting to terminals. Thus, it would be advantageous to use as low transmission power as possible sufficient to reach the terminals, or possibly an adaptive power mechanism.

<u>Directional Antennas and Multiple Transmitters at the Station</u>

This section is addressed only to a terrestrial system. When the station uses a directional antenna, then its transmission to repeater R_i does not interfere with R_j, $j \neq i$. Consequently, the average rate of transmission per slot to a single repeater is G_1/m (assuming equal share of load). The only change that would result in Equations (10) and (11) is the replacement of G_1 in Equation (10) by G_1/m. Doing so and equating S_1 and S_2 results in $G_1 = G_2$ and:

$$S_1 = G_1\left(1 - \frac{G_1}{m}\right)^I ; \quad 0 \leq G_1 \leq 1 ; \ 1 \leq I \leq m \qquad (17)$$

$$S_2 = G_2\left(1 - \frac{G_2}{m}\right)^I ; \quad 0 \leq G_2 \leq m ; \ 1 \leq I \leq m \qquad (18)$$

The capacity of the system is given by

$$S_1^* = \begin{cases} \frac{m}{I+1}\left(1 - \frac{1}{I+1}\right)^I ; \; m \leq I + 1 & \text{(19a)} \\ \left(1 - \frac{1}{m}\right)^I ; \; m > I + 1 & \text{(19b)} \end{cases}$$

Figure 9 shows the capacity of the system as a function of m for I = 1 and I = m. The same curves for an omni-directional antenna are shown as a reference. It can be seen that the capacity of the system with a directional antenna is substantially higher; in particular, when the interference level is low.

We now address the question of multiple transmitters and antennas at the station. Consider the capacity of the system with a directional antenna at the station, Equation (19). The maximum given by (19a) is a stationary point whereas that given by (19b) is a boundary point at $G_1 = 1$. Also note that

$$\frac{m}{I+1}\left(1 - \frac{1}{I+1}\right)^I \geq \left(1 - \frac{1}{m}\right)^I, \text{ for } m \leq I + 1 \qquad (20)$$

Thus, if one increases the domain of G_1, it would result in an increase in system capacity, which will then be given by Equation (19a). This is exactly what happens when adding additional transmitters and antennas to the station. It enables the station to transmit more than one packet into the same "slot in time" and direct the packets to different repeaters. In practice, if the station has several transmitters and directional antennas which enables it to transmit simultaneously in different directions, then one can devise an algorithm at the station, which will properly select the directions to which packets are simultaneously transmitted, so as to further reduce the interference level I (manage its transmissions).

Figure 10 shows the system throughput as a function of the rate of transmission from station to repeaters (or from repeaters to terminals, note $G_1 = G_2$). One can see that for I = 1 and I = 3, the maximum system capacity cannot be obtained with a single transmitter and the value obtained is at the boundary point at $G_1 = 1$ and given by (19b). Notice that when the number of repeaters is large and the interference level is low, then there is a large difference between the maximum capacity and the constrained maximum capacity.

We now determine the minimum number of transmitters and directional antennas needed at the station. If the interference level I is constant then the unconstrained capacity of Equation (19a) is increasing with the number of repeaters m. Moreover, the capacity increases also in the case that I is a linear function of m. For let I = km, $1/m \leq k \leq 1$, then the unconstrained capacity is given by

$$S^* = \frac{m}{km + 1} \left(1 - \frac{1}{km + 1}\right)^{km} \qquad (21)$$

and

$$\frac{ds^*}{dm} = \frac{km}{(km + 1)^2} \left(1 - \frac{1}{km + 1}\right)^{km - 1} \geq 0; \text{ for } k \geq 0 \qquad (22)$$

Since the capacity is increasing as a function of m, one can obtain a capacity greater than 1 (see for example Figure 10, I = 1). The minimum number of transmitters that can realize the unconstrained capacity is given by the rate of transmission from the station to repeaters at which the maximum utilization is obtained. That is:

$$\text{Minimum Number of Transmitters} = \left\lceil \frac{m}{km + 1} \right\rceil = \left\lceil \frac{m}{I + 1} \right\rceil \qquad (23)$$

where $\lceil x \rceil$ is the smallest integer greater than x. Equation (23) also implies that multiple transmitters will not result in an increase in system capacity when the interference level is high, i.e., $m \leq km + 1$. It is easy to verify that when the constraint on G_1 is satisfied, the constraint on G_2 is also satisfied.

By associating a cost value with a repeater and with each additional transmitter at the station, one can formulate a design optimization problem in which the system cost is traded against the increase in capacity which it results.

The results of this section demonstrate that a single slotted ALOHA channel can be used (and reused) spatially to obtain channel utilization higher than 100%.

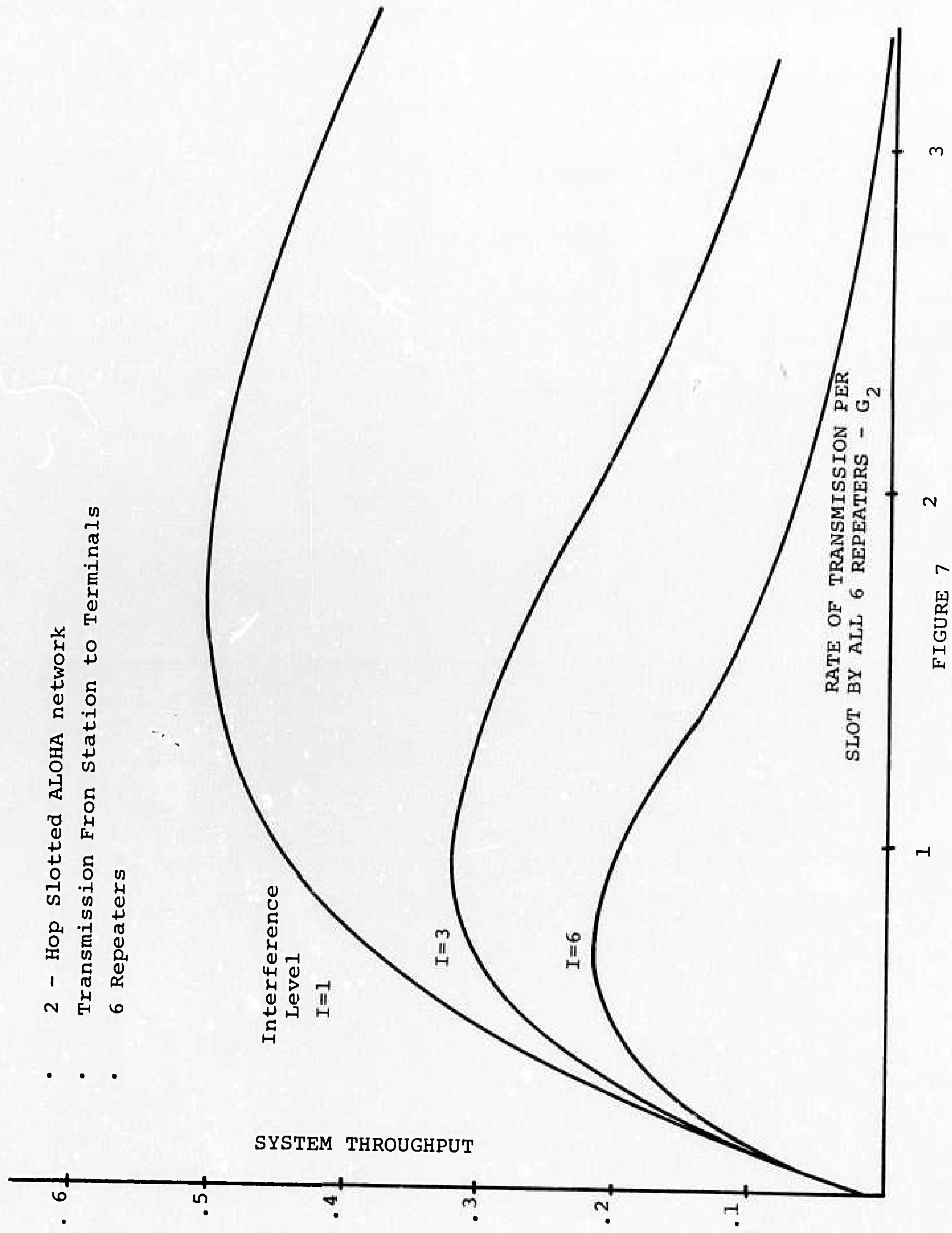
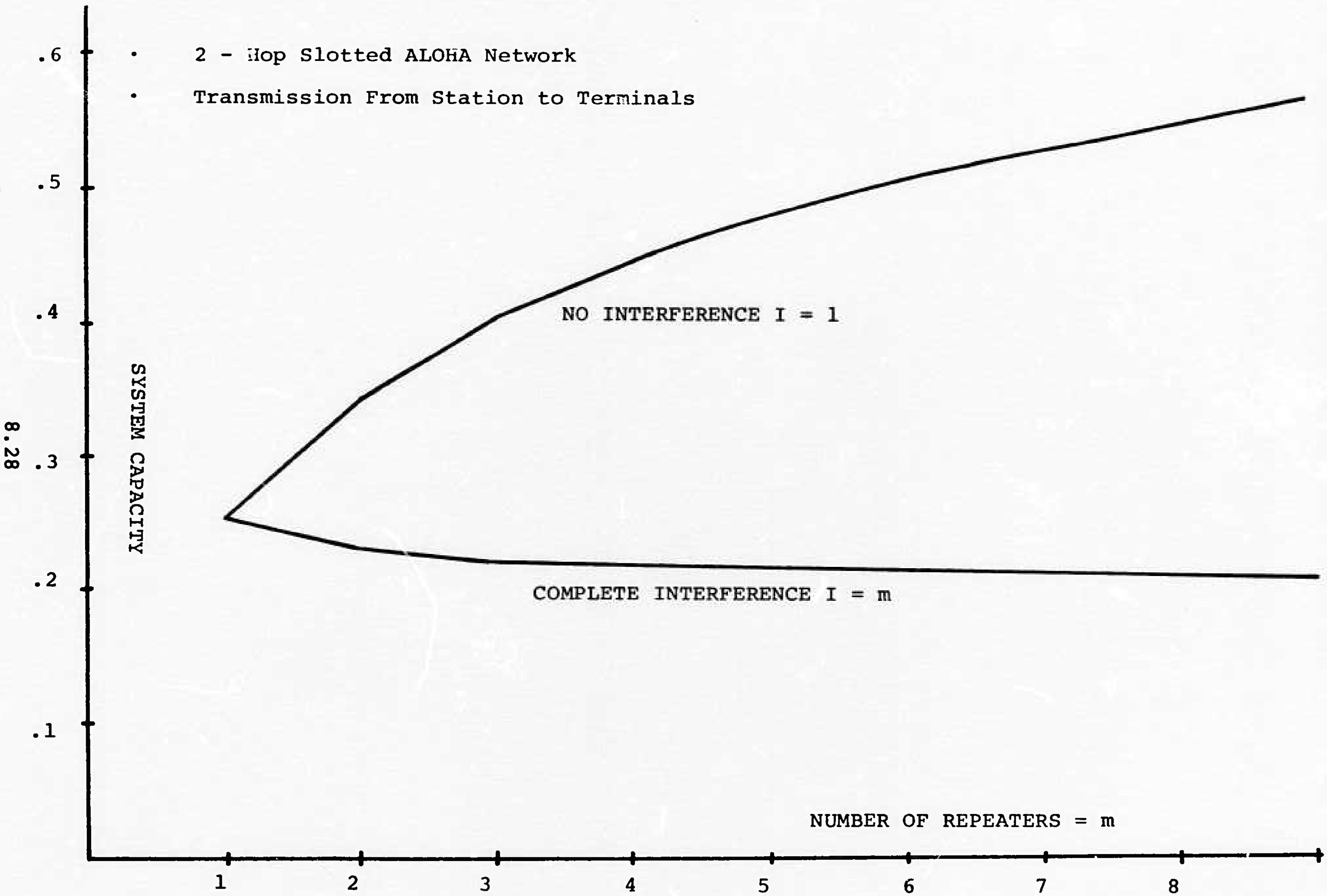
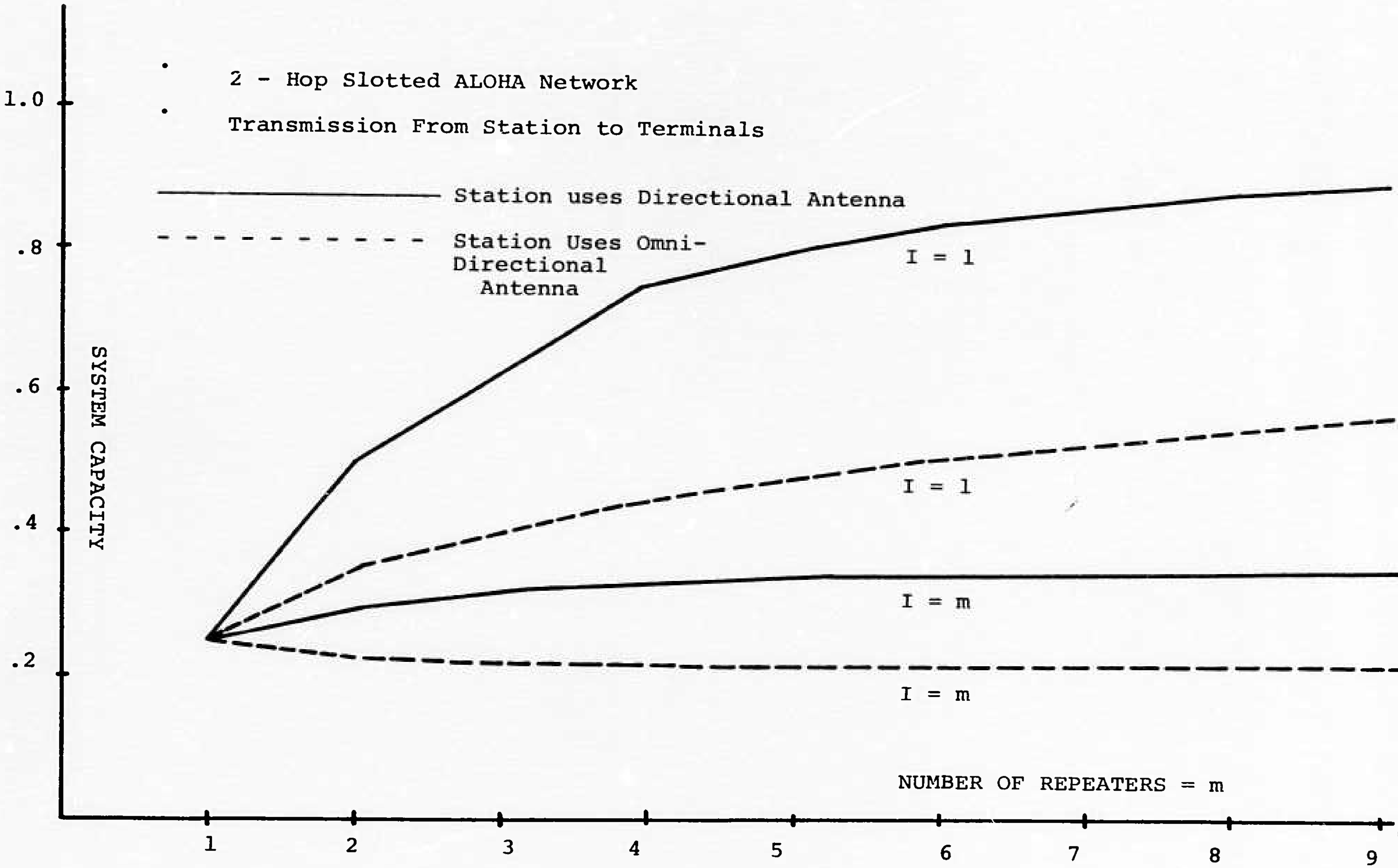

. 2 - Hop Slotted ALOHA network
. Transmission Fron Station to Terminals
. 6 Repeaters
SYSTEM THROUGHPUT
Interference Level I=1
I=3
I=6
RATE OF TRANSMISSION PER SLOT BY ALL 6 REPEATERS - G_2
.1
.2
.3
.4
.5
.6
1
2
3

FIGURE 7



• 2 - Hop Slotted ALOHA Network
• Transmission From Station to Terminals
SYSTEM CAPACITY
.6
.5
.4
.3
.2
.1
NO INTERFERENCE I = 1
COMPLETE INTERFERENCE I = m
NUMBER OF REPEATERS = m
1
2
3
4
5
6
7
8
9

FIGURE 8



2 - Hop Slotted ALOHA Network
Transmission From Station to Terminals
Station uses Directional Antenna
Station Uses Omni-Directional Antenna
I = 1
I = 1
I = m
I = m
SYSTEM CAPACITY
1.0
.8
.6
.4
.2
NUMBER OF REPEATERS = m
1
2
3
4
5
6
7
8
9

FIGURE 9

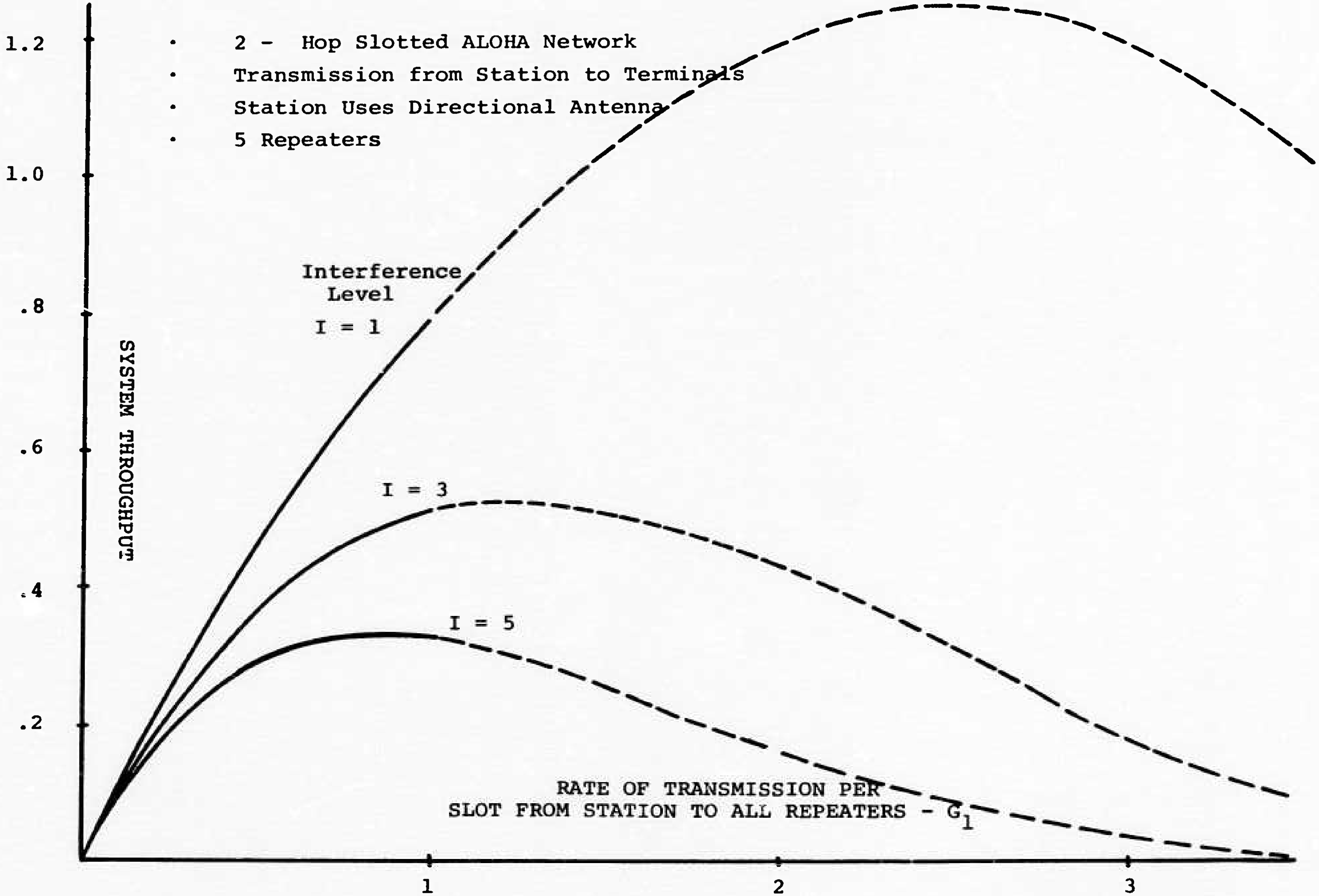

• 2 - Hop Slotted ALOHA Network
• Transmission from Station to Terminals
• Station Uses Directional Antenna
• 5 Repeaters
SYSTEM THROUGHPUT
1.2
1.0
.8
.6
.4
.2
Interference Level
I = 1
I = 3
I = 5
1
2
3
RATE OF TRANSMISSION PER SLOT FROM STATION TO ALL REPEATERS - G1

FIGURE 10

V. CONCLUSIONS

Our conslusions from the analysis are outlined.

A. Transmission to Station

1. When the number of repeaters is small (the exact number depends on the interference level), then the critical hop in the network (the capacity bottleneck) is from terminals to repeaters.

2. When the number of repeaters is large, the critical hop is from repeaters to station.

3. The 2-Hop network design which maximizes the system capacity has 2 repeaters when the interference is minimum, and 3 repeaters when the interference is maximum. Additional repeaters reduce system capacity.

4. The capacity of a 2-Hop network is higher than that of a 1-Hop network when the number of repeaters is 2 or more, and is lower than the capacity of a 1-Hop network when there is one repeater.

5. Directional antennas at repeaters increase system capacity when the critical hop is from terminals to repeaters. The increase is significant only in the case when there are 2 repeaters, which do otherwise interfere with each other. In other cases, the increase in capacity is either insignificant or does not exist.

B. Transmission from Station

1. The interference of the station with the transmission of repeaters to terminals reduces significantly the system capacity. Thus, if possible, it is important to enable terminals to receive such transmissions.

2. The system capacity reduces substantially when the interference level between repeaters is increased. Note that this is not the case when transmitting to the station; compare Figures 5 and 8. Consequently, in a terrestrial system, it is important to reduce the interference factor by a mechanism such as adaptive power.

3. A directional antenna at the station in a terrestrial system increases significantly, the system capacity when the interference level between repeaters is low to moderate. This is not the case when the interference level is high, since the throughput on the hop from repeaters to terminals is limited due to this interference.

4. When the station has directional antennas, then multiple transmitters and antennas may further increase, significantly, system capacity. Note that in this case, one can obtain a capacity greater than 1.

5. An equation for the number of transmitters needed at the station is given. This number increases when the interference level decreases. When the interference level is high, then none of the devices in conslusions B.3, B.4, B.5, are desirable, since the capacity is limited by the throughput from repeaters to terminals.

REMARKS:

Conclusions A.5 and B.3 which relate to directional antennas, imply that directional antennas are generally not useful when devices which use them direct transmissions to a single location in space; because the interference at this location is not avoided. On the other hand, directional antennas are very useful when oriented to different azimuths, because one can take advantage of the spatial districution of receivers.

REFERENCES

1. Abramson, N., "The ALOHA System - Another Alternative for Computer Communications," Fall Joint Computer Conf. AFIPS Conf. Proc., Vol. 37, 1970, pp. 281-285.

2. Abramson, N., "Packet-Switching With Satellites," National Comp. Conf., June 1973, pp. 695-702.

3. Gitman, I., R. M. Van Slyke, H. Frank, "On Splitting Random Access Broadcast Communication Channels," Proceedings of the Seventh Hawaii International Conf. on System Sciences, Subconference on Computer Nets, January 1974.

4. Kleinrock, L. and S. S. Lam, "Packet-Switching in a Slotted Satellite Channel," National Computer Conf., June 1973, pp. 703-710.

5. Metcalfe, R. M., "Steady-State Analysis of a Slotted and Controlled ALOHA System with Blocking," ARPANET Satellite System Note #16, (NIC Document #11624).

6. Roberts, L. G., ARPANET Satellite System Notes 8, (NIC Document #11291), available from ARPA Network Info. Center, Stanford Research Institute, Menlo Park, Calif.

7. Roberts, L. G., "Dynamic Allocation of Satellite Capacity Through Packet Reservation," National Computer Conf., June 1973.

REFERENCES (Continued)

8. Kleinrock, L. and S. S. Lam, "On Stability of Packet Switching in a Random Multi-Access Broadcast Channel," Proceedings of the Seventh Hawaii International Conf. on System Sciences, Subconference on Computer Nets, January 1974.

9. Network Analysis Corporation, "Comparison of Hop-by-Hop and End-to-End Acknowledgement Schemes," PRTN #7, Available from ARPA Network Info. Center, Stanford Research Institute, Menlo Park, Calif.

PACKET RADIO SYSTEM CONSIDERATIONS - CHANNEL CONFIGURATION

I. INTRODUCTION

Consider a communication channel which is shared by two independent sources of traffic in a broadcast mode. Source 1 is generated by an infinite number of terminals, each with an infinitesimal traffic rate, and which collectively form a finite Poisson source. Source 2 is generated by a finite number of terminals each with a finite traffic rate. The terminals transmit fixed size packets and access the channel using the so called "slotted ALOHA" random access scheme. A terminal can transmit to any other terminal in the system.

This model can describe the ALOHA system at the University of Hawaii [1] or the Packet Radio System [6]. In the Packet Radio System there is a large number of terminals which communicate with a small number of stations. The terminals can be modelled by the terminals of Source 1 and the stations by the terminals of Source 2. The model is suitable for a Packet Radio System in an urban area where a terminal can directly transmit to a station and where any transmission from a terminal or station interferes with all other devices.

Roberts [5] has shown that if all the traffic is contributed by Source 1, then the maximum throughput which can be obtained is 1/e of the channel capacity. Abramson [2] and Kleinrock and Lam [3] have shown that the maximum throughput can be increased when the traffic is composed of contributions from both Source 1 and Source 2.

We approach the problem from a synthesis viewpoint. That is, given Source 1 and Source 2, the question is whether one should split the channel so that one part is used by Source 1 and the other part by Source 2; or alternatively, should one use the total capacity in common. The criteria for decision are maximum throughput and average delay. We consider a channel split in which the slots are partitioned between transmissions from Source 1 and Source 2, however, all terminals receive on all time slots. It is shown that the choice of channel configuration depends on the number of terminals in Source 2, n, and on the ratio of packet rate of Source 2 to Source 1, designated by α. Further, given n, there is an interval of α for which a higher maximum throughput can be obtained by splitting the channel.

The problem considered in this chapter was addressed in [7] for n=1 but for several random access schemes; specifically, for the nonslotted and slotted ALOHA and for the carrier sence [8] random access schemes. The qualitative conclusions of [7] are the same as in this chapter.

II. PRELIMINARY ANALYSIS

A. The Slotted ALOHA Channel

In the slotted ALOHA random access mode, a channel is partitioned into segments of time (slots) equal to a packet transmission time. Terminals transmit their packets into random slots in time. That is, there is no coordination among the terminals as far as the choice of the slot is concerned; however, there is a universal clock which enables each terminal to start the transmission of its packet at the beginning of a slot. If two or more packets are transmitted in the same slot, it is assumed that none of the packets are correctly received and each of the terminals will retransmit its packet at some randomly chosen future slot.

One can see that the number of packets transmitted (the channel traffic) is larger than the number of packets offered to the system due to the retransmissions of packets which collide. Let S denote the rate of packet originations per slot offered to the channel, and G the rate per slot of packets plus retransmissions. Assume that the two origination processes are Poisson. Further assume that the probability that a packet is blocked, given the packet is new, equals the probability that a packet is blocked given it is a retransmission [3].

B. The Single Source Case

If all the traffic is contributed from Source 1, Roberts [5] has shown that the relation between S_1 and G_1 is given by:

$$S_1 = G_1 e^{-G_1} \qquad (1)$$

Abramson [2] considered the case where all the traffic is contributed from Source 2 of n identical terminals and has shown that

$$S_2 = G_2 \left(1-\frac{G_2}{n}\right)^{n-1} \tag{2}$$

One can see that Eq. (2) takes the form of Eq. (1) when $n \to \infty$. The maximum values of S_1 and S_2 are given by:

$$S_1^* = \frac{1}{e} \; ; \; S_2^* = \left(1-\frac{1}{n}\right)^{n-1} \tag{3}$$

The operation of the slotted ALOHA channel may become unstable [4]. In this paper we consider the steady state case, in which the offered rate is also the throughput per slot or the utilization of the channel.

C. Mixed Sources on a Common Channel

The performance of a channel which is shared by terminals from Sources 1 and 2 has been analyzed by Abramson [2], and Kleinrock and Lam [3]. The offered packet rate to the channnel is $S_1 + S_2$, and the channel traffic $G_1 + G_2$. The following equations hold:

$$S_1 = G_1 \left(1-\frac{G_2}{n}\right)^{n} e^{-G_1} \tag{4}$$

$$S_2 = G_2 \left(1-\frac{G_2}{n}\right)^{n-1} e^{-G_1} \tag{5}$$

III. MAXIMUM UTILIZATION OF THE SPLIT AND COMMON CHANNELS

Given Source 1 and Source 2, the question is whether one should split the channel so that one part is used by Source 1 and the other part by Source 2; or alternatively, should one use the total capacity in common. In this section we compare the channel configurations in terms of maximum utilization. To obtain an absolute comparison, we introduce the parameter of the ratio of packet rates of Source 2 to Source 1:

$$\alpha = \frac{S_2}{S_1} \tag{6}$$

We shall use the subscripts c and s to denote a common channel and a split channel, respectively; and the superscript * to denote the optimum or maximum values. The total given capacity will be assumed as one unit and (C_1, C_2), $C_1 + C_2 = 1$, will denote a channel split where the fraction C_1 is assigned to Source 1 and the fraction C_2 to Source 2. Given an arbitrary split (C_1, C_2), the maximum utilization of the configuration is given by:

$$S_s^* = \begin{cases} C_1(1+\alpha)S_1^* & ;\alpha \leq \dfrac{S_2^* C_2}{S_1^* C_1} \quad \text{(7a)} \\[2ex] C_2(1+\dfrac{1}{\alpha})S_2^* & ;\alpha > \dfrac{S_2^* C_2}{S_1^* C_1} \quad \text{(7b)} \end{cases}$$

Corresponding to C_1 and C_2 being saturated respectively.

If α is known one can split the channel optimally to obtain the highest maximum utilization of the total capacity. It can be shown that the optimum split (C_1^*, C_2^*) satisfies the following equation:

$$\frac{C_2^*}{C_1^*} = \frac{\alpha S_1^*}{S_2^*} \tag{8}$$

Corresponding to both channels saturating simultaneously.

If the channel is optimally split, then the maximum utilization of the total capacity is given by:

$$S_s^* = C_1^* S_1^* + C_2^* S_2^* = \frac{(1+\alpha)(1-\frac{1}{n})^{n-1}}{e(1-\frac{1}{n})^{n-1} + \alpha} \tag{9}$$

Note from Eq. (9) that when $\alpha \to 0$, $S_s^* \to \frac{1}{e}$, and when $\alpha \to \infty$, $S_s^* \to (1-\frac{1}{n})^{n-1}$.

The total utilization of the common channel configuration is given by the summation of Eqs. (4) and (5).

$$S_c = e^{-G_1} (1-\frac{G_2}{n})^{n-1} [G_2+G_1 (1-\frac{G_2}{n})] \tag{10}$$

To obtain the maximum of S_c, one has to maximize Eq. (10) subject to the constraint $S_2/S_1 = \alpha$. Alternatively, we can use the condition of the channel traffic at the maximum utilization obtained by Abramson [2]. Doing so, we obtain:

$$S_c^* = e^{-G_1^*}(1-\frac{G_2^*}{n})^{n-1}(1-\frac{G_1^* G_2^*}{n}) \tag{11}$$

where,

$$G_2^* = \frac{n + \alpha(n+1) - \sqrt{[n+\alpha(n+1)]^2-4\alpha^2 n}}{2\alpha} \tag{12}$$

and,

$$G_1^* = 1 - G_2^* \tag{13}$$

Figure 1 shows the comparison between the maximum utilizations of the two configurations S_c^* and S_s^*. It is shown as a function of α with n as a parameter. The values of n used are 1, 3, and ∞, and the split $C_1 = C_2 = \frac{1}{2}$.

From Figure 1, one can see that for a given n, there is an interval of α such that within this interval $S_s^* \geq S_c^*$, and outside of the interval $S_s^* < S_c^*$. Furthermore, the interval discussed decreases when n increases and constitutes a single point when $n = \infty$.

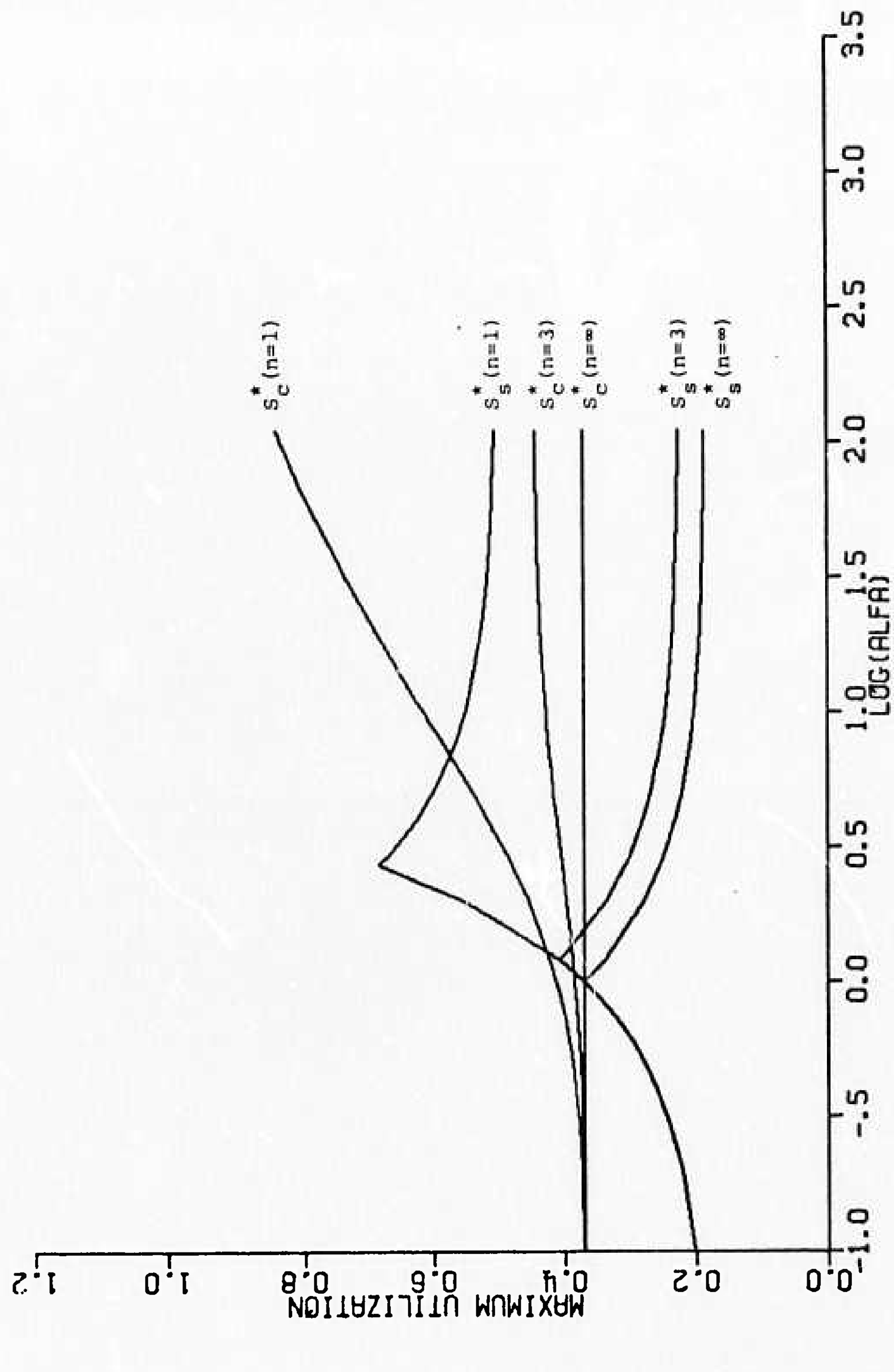


FIGURE 1

MAXIMUM UTILIZATION VS. SPLIT AND COMBINED CHANNEL PARAMETERS

IV. DELAY CONSIDERATIONS

It is clear that when α is within the interval discussed in the previous section, the split configuration is better than the common configuration in terms of maximum throughput and average delay, since an infinite delay is obtained in the common configuration before it reaches the maximum utilization of the split configuration. The average delay of a packet from Source 1 will usually be different from that of a packet from Source 2. In particular, in the split configuration when the maximum utilization is obtained, one channel is saturated (infinite delay) and the other is not; except for the optimum splitting α for which both channels saturate simultaneously.

The average delay in this system is composed of the delay when the first transmission is successful plus the average number of retransmissions times the average delay per retransmission. A terminal in Source 2 also encounters queueing delay.

We show curves of delay vs. throughput for several parameters α. We consider a terrestrial system in which propagation delay is ignored, and use all the assumptions given in Section 1. The average delay equations in units of slot times used are the following:

$$D_1 = 1.5 + \left(\frac{G_1 - S_1}{S_1}\right) \bar{k} \quad (14)$$

$$D_2 = 1 + .5\left(1 - \frac{G_2}{n}\right) + \frac{G_2/n}{2(1 - G_2/n)} + \left(\frac{G_2 - S_2}{S_2}\right) \bar{k} \quad (15)$$

where $\bar{k}$ is the average waiting per retransmission. The value .5 is added to represent that when a packet is ready for transmission, the terminal will wait one half slot, on the average, until the beginning of a slot. The third term in Eq. (15) represents the queueing delay at terminals of Source 2. When writing the queuing delay we assume that packets arrive according to a Poisson distribution and require constant service time equal to the packet transmission time (an M/D/1 queueing system). Note that when n then Eq. (15) takes the same form as Eq. (14).

The delay equations hold for the split as well as the common channel configurations. However, if one assumes the same packet length in each case and the slot time on the common channel is one unit, then the slot time on the split channel will be $1/C_1$ and $1/C_2$ for the respective channels. This has been taken into account when showing the delay curves.

Figures 2, 3, and 4, show the delay as a function of throughput for $\alpha = .5$, $\alpha = 2.5$, $\alpha = 10.$, and for $n = 1, 3,$ and ∞. Other parameters are $C_1 = C_2$ and $\frac{1}{2}$ and $\bar{k} = 4$. From Figure 1, one can see that $\alpha = .5$ and $\alpha =10.$ corresponds to the case where $S_c^* > S_s^*$; on the other hand $\alpha = 2.5$ corresponds to the case where $S_c^* < S_s^*$. The throughput shown is the sum of both sources and the delay is an average of D_1 and D_2 weighted by the throughputs of the sources.

One can see that when $S_c^* > S_s^*$ the common channel configuration results in lower average delay for all values of throughput (for the same value n). On the other hand, if $S_s^* > S_c^*$ (Figure 3), there are operating points, for example when the throughput is .55, where the split channel configuration is better both in throughput and delay. However, even in this case, when the system operates at low throughputs, the common channel configuration results in lower values of delay. This is due to the differences in the slot times.

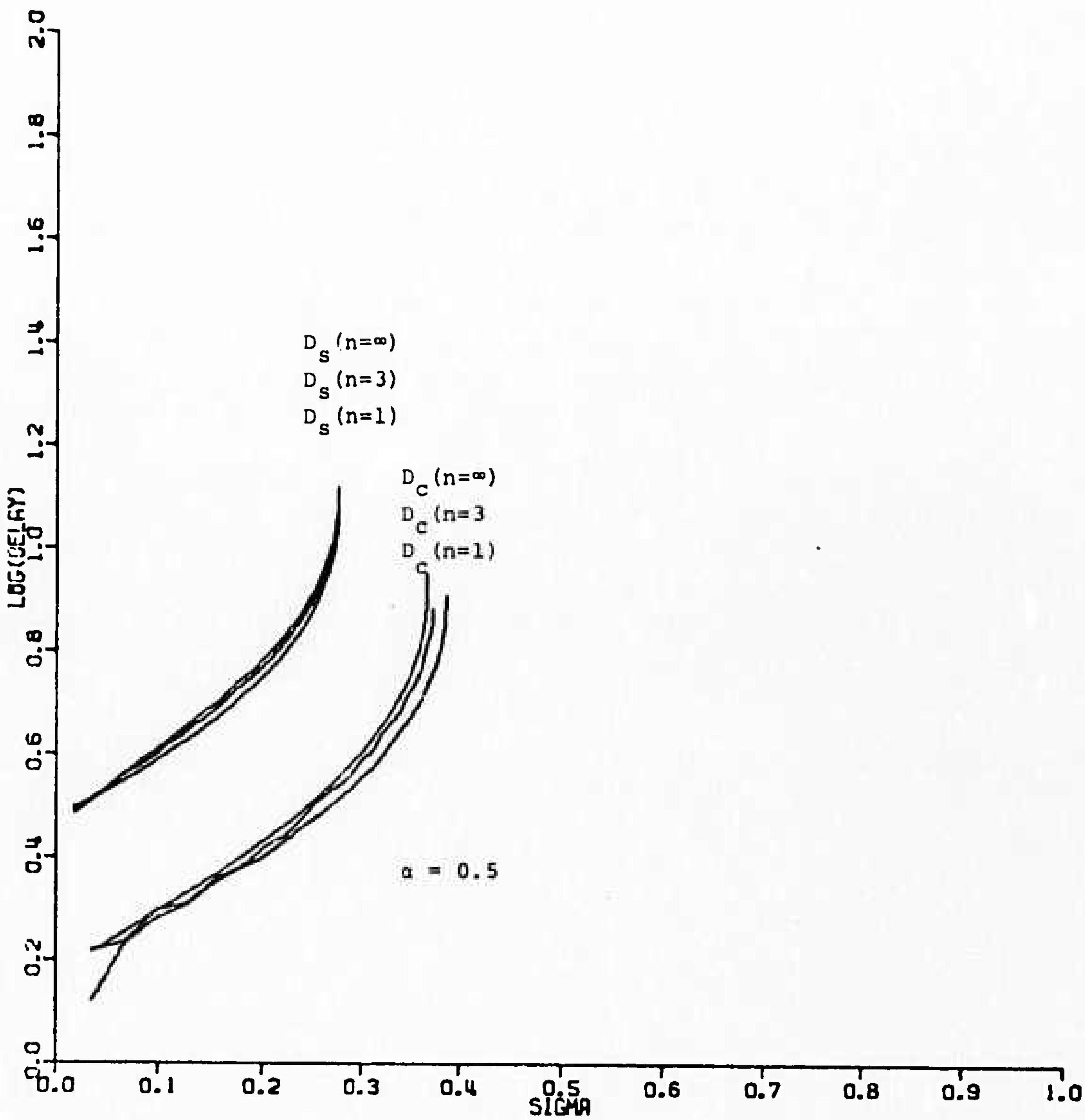


FIGURE 2

DELAY VS. THROUGHPUT, $\alpha = .5$

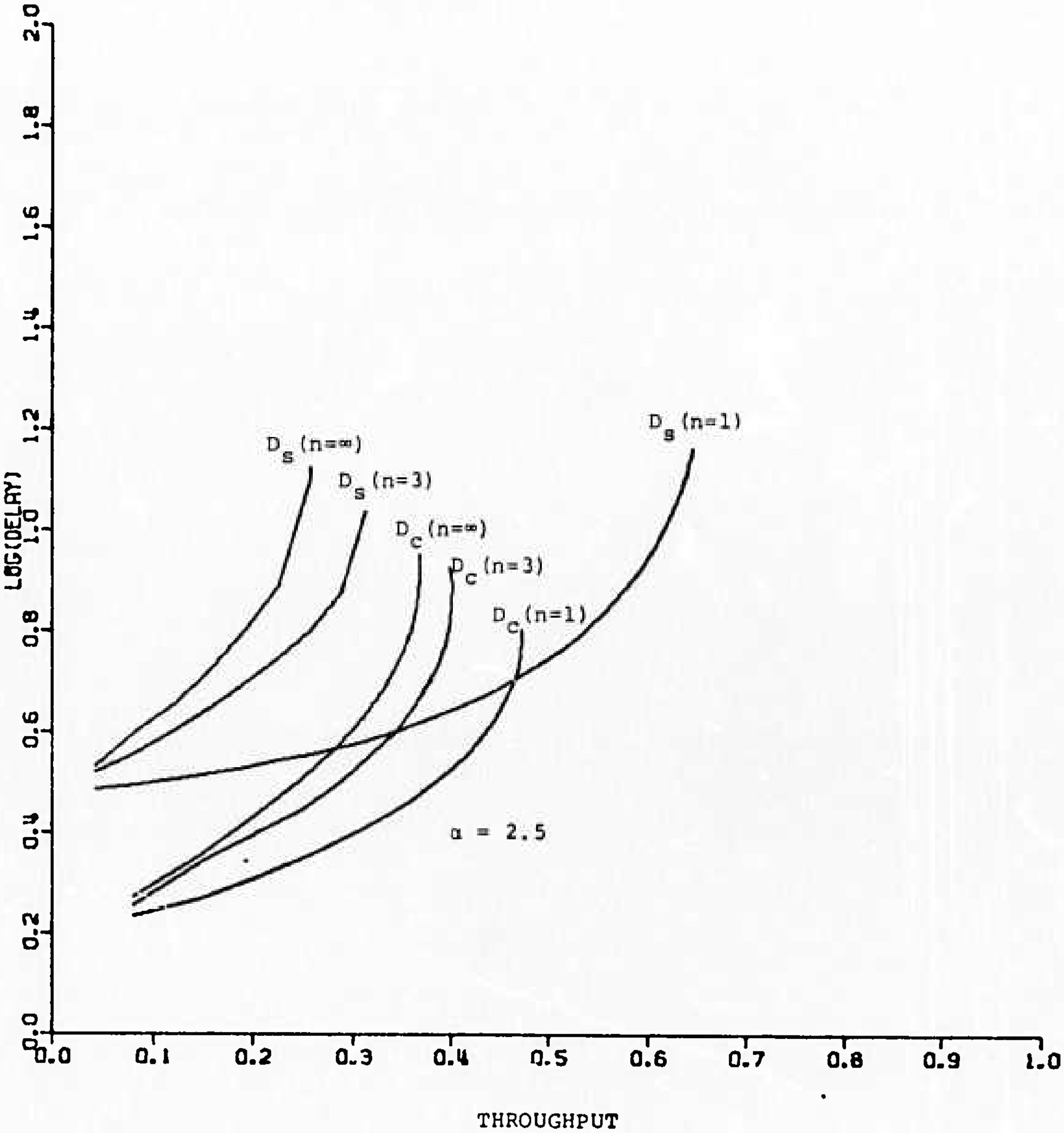


FIGURE 3

DEALY VS. THROUGHPUT, $\alpha = 2.5$

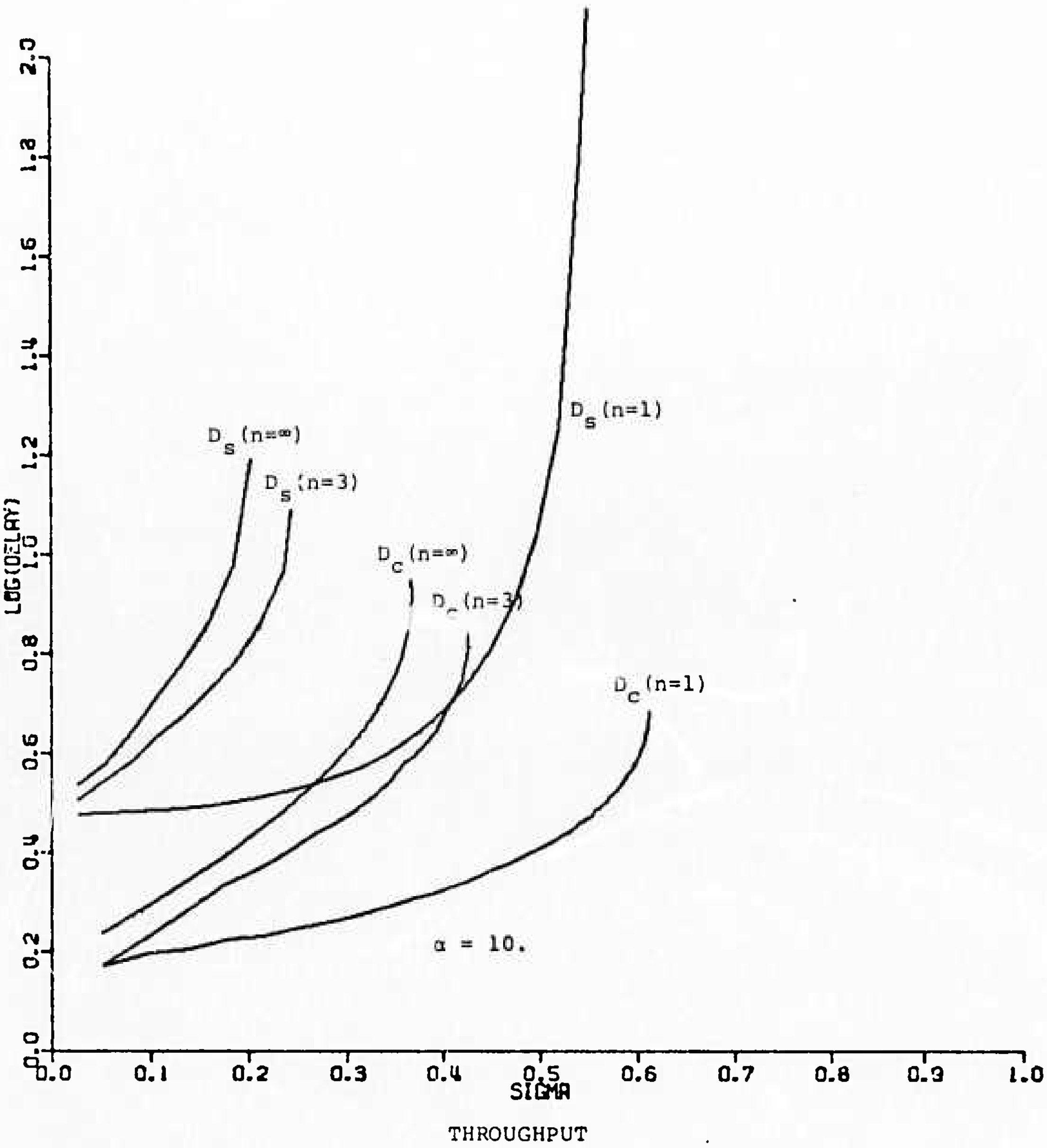


FIGURE 4

DELAY VS. THROUGHPUT, $\alpha = 2.5$

V. CONCLUSIONS

We relate the results obtained to the Packet Radio System. If one assumes that a packet which originates from a terminal will always be directed towards a station then α has the practical meaning of the average number of response packets from station to terminal for every packet which is successfully transmitted from a terminal to a station.

It is demonstrated that if this ratio α is known, then one can split the channel into two to obtain a higher maximum utilization of the total channel capacity. It is also shown how to split the channel. However, if the channel is split and the system operates at low values of throughput, then the average delay is higher than would result from operating in a common channel configuration. Another conclusion is that if α is not known, or if it varies, then the common channel configuration will be better in terms of throughput and delay.

Finally, we note that the system analyzed models the Packet Radio System when there is no coordination among the stations in choosing the slot, so that packets from two stations can collide on the same slot. In practice, however, one may consider having signalling channels among the stations to obtain the above coordination dynamically. In this case, the set of stations has to be considered as a single source; that is, $n = 1$.

REFERENCES

1. Abramson, N., "The ALOHA System - Another Alternative for Computer Communications," Fall Joint Computer Conf., AFIPS Conf. Proc., Vol. 37, 1970, pp. 281-285.

2. Abramson, N., "Packet-Switching With Satellites," National Comp. Conf., June 1973, pp. 695-702.

3. Kleinrock, L. and Lam, S. S., "Packet-Switching in a Slotted Satellite Channel," National Computer Conference, June, 1973, pp. 703-710.

4. Kleinrock, L. and Lam, S.S., "On Stability of Packet Switching in a Random Multi-Access Broadcast Channel," Proceedings of the Seventh International Conference on System Sciences, January 8-10, 1974.

5. Roberts, L., ARPANET Satellite System Notes 8, (NIC Document #11291), available from ARPA Network Information Center, Stanford Research Institute, Menlo Park, Calif.

6. NAC (a), "Packet Radio System - Network Considerations", available from ARPA Network Information Center, Stanford Research Institute, Menlo Park, California.

7. NAC (b), "Channel Configuration for Packet Radio System," PRTN #45, (NIC Document #15692), ARPA Network Information Center, Stanford Research Institute, Menlo Park, Calif.

8. Kleinrock, L., and F. Tobagi, "Carrier Sense Multiple Access (CSMA)", available from ARPA Network Information Center, Stanford Research Institute,Menlo Park, Calif.

AREA COVERAGE BY LINE-OF-SIGHT RADIO

I. PROBLEM FORMULATION

We are concerned with line-of-sight coverage of an area where mobile terminals or fixed terminals are transmitting by radio from unspecified locations. The problem is to locate repeaters so that any such terminal will be in line-of-sight of repeaters and that there be reliable connections between every pair of terminals (and repeaters). More precisely we wish to minimize the installation cost and maintenance cost of the repeaters subject to a constraint on the reliability of service.

In general, determining if line-of-sight micro-wave transmission between two points is possible, involves taking into account many factors including wave-length (Fresnel zones), weather conditions (effective earth radius), antenna design, height, topography, etc.. Nevertheless, we shall assume that there are known functions $\ell(r,t)$ and $L(r_1,r_2)$ that are 1 if a repeater at location r can communicate with a terminal at location t and if a repeater at location r_1 can communicate with a repeater at location r_2 respectively and are 0 otherwise.

From purely topographical considerations it is obvious that the "flat terrain" problem and the "hilly terrain" problem should be handled seperately. For "flat terrain" the problem is homogeneous, i.e. installation costs, maintenance costs can be assumed equal at all locations and the transmission properties are identical at all points. The "flat terrain" problem is discussed in Section 3; a model is suggested for which we compute an optimal solution.

The primary concern in the ensuing paragraphs is with the "hilly terrain" problem. By that one should understand hilly topographies as well as flat topographies that cannot be viewed as homogeneous from a cost or transmission viewpoint.

In "hilly terrain" it is impractical to consider all possible locations of repeaters and terminals, which theoretically are infinite in number. We shall limit ourselves to a finite set R of possible repeaters locations and a finite set T of possible terminal locations. How the set R and T are chosen will be of great computational importance and will probably be chosen adaptively. But for the time being, we assume R and T known and fixed.

The principal and immediate interest is in an appropraite mathematical model of the situation and some indications on how to solve the problem. The first problem is the proper choice of reliability measure or grade of service. We assume that the radio network is for local distribution-collection of terminal traffic with rates small compared to the channel capacity so that throughput capacity is not a constraint. That is, if any path through the network exists for a given pair of terminals we assume there is sufficient capacity for traffic between them. Possible measure of network reliability that have proved useful in the analysis of communication networks [13] are the probability that all terminal pairs can communicate and the average fraction of terminal pairs which can communicate. However, for network <u>synthesis</u> as distinguished from <u>analysis</u> these approaches appear too difficult both from computational and data collection points of view. This suggests the "deterministic" requirement that there exist k node disjoint paths between every terminal pair. This guarantees that at least k repeaters or line of sight links must fail before any terminal pair is disconnected. Let the cost of a repeater at location $r \in R$ be $c(r)$ and $c(R^\circ) = \Sigma \{c(r) | r \in R^\circ\}$ where $R^\circ \subset R$. Then we can formulate:

<u>Problem I</u>: Find $R^* \subseteq R$ minimizing $c(R^*)$ subject to the constraint that for all t T and r R there exist k node disjoint

paths from t to r in the network N (t;R*) where

$$N(t;R^*) = [R^* \cup \{t\}, \{(r_1,r_2) \mid L(r_1,r_2)=1\} \cup \{(r,t) \mid \ell(r,t)=1\}]$$

One might demand only that there be k node disjoint paths between every pair of terminals instead of between each terminal-repeater pair, but we are assumming that communication always takes place through a "station" which could be any of the repeaters. The analysis of the terminal to terminal model is similar in any case.

The following two propositions motivate a new Problem II,

Proposition 1: $|\{r \mid \ell(r,t) = 1\} \cap R^*| \geq k.$

Proposition 2: For all $r \in R^*, |\{r_1 \mid L(r,r_1)=1\}| \geq k-1.$

If for each $r \in R$ there exists $t \in T$ such that $l(r,t)=0$ then $|\{r_1 \mid L(r,r_1)=1\}| \geq k.$

Problem II: Choose $R^* \subseteq R$ to minimize $C(R^*)$ subject to:

1. For all $t \in T, |\{r \mid \ell(r,t)=1\} \cap R^*| \geq k$ (the k-fold set covering problem).
2. For all $r_1, r_2 \in R^*, r_1 \neq r_2$, there exist k node disjoint paths connecting r_1 to r_2. (the minimum cost redundant network problem).

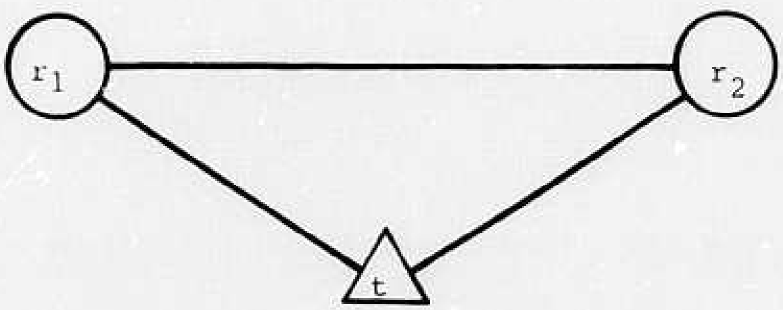
The virtue of II - as compared to I is that II is an amalgam of two well studied network problems, the set covering problem [7] and a problem closely related to the minimum cost redundant network problem [11]. We can then attack the problem using previously developed techniques.

Problem I and II are related by:

Proposition 3: Any R* satisfying the constraints of II also satisfies the constraints of I.

<u>Proof</u>: Suppose R* satisfies the constraints of Problem 2 and violates those of Problem I. Then there exists a terminal t_0 and a repeater r such that there are not k node disjoint paths connecting them in $N(t_0;R^*)$. According to the Menger Graph Theorem [8], there are k-1 repeaters $r_1,\ldots,r_{k-1}$ not including r_0 such that when these repeaters are removed from R* there is no path from t to r_0. But this leads to a contradiction. Removing k-1 repeaters cannot disconnect t_0 from the net by 1 of Problem II. Thus, there is an arc (t_0,r') with $(r',t°) = 1$. Similarly, k-1 repeaters cannot disconnect r' from r_0 by 2. Thus, there is a path from t_0 to r.

The converse is not necessarily true; that is, there may be feasible solutions to I which are not feasible to II. Figure 1 depicts a counter-example to a possible converse for k=2.



<u>FIGURE 1</u>

However, by Proposition 3, solving II is fail-safe in the sense that the solution obtained will always be feasible for Problem 1.

Feasible solutions to I which are not feasible in II will be quite rare. In order for this to happen, there would have to be two repeaters r_1 and r_2 in R* (where R* is an assumed solution to I) for which there are not k node disjoint paths joining them yet for which all terminals in the vicinity of r_1 can communicate by k node disjoint paths with r_2 and conversely. This does not happen unless there are very few terminals and many repeaters near r_1 and r_2 which from the physical nature of the problem is highly unlikely. On the other hand, as pointed out above, artificial counter-examples to a possible converse of Proposition 3 can easily be constructed.

II. COMPUTATIONAL TECHNIQUES

The following problem will be referred to as a k-cover problem. Let N be a bipartitie graph with edges E and nodes (R,T). The edges E connect nodes of R to nodes of T. Then the following is a k-cover problem:

Find $R^* \subset R$ such that valence of each T-node in N^* is at least k and such that the cardinality of R^* is minimum (number of elements of R^*), where N^* is the subgraph obtained by deleting all nodes $R \setminus R^*$ (where $\setminus$ indicates set difference) and all edges connected to these nodes. This is also known as the α-width problem [4], [10].

The 1-Cover Problem is the classical set covering problem. Extensive research has been and is being done on the 1-cover problem, see e.g., [7] and references mentioned therein, see also [1] (A special case of the 1-cover problem is the simple covering problem where each R-node is connected to exactly 2 T-nodes. For this problem algorithms are known that are "efficient" i.e., with known polynomial bounds on the number of operations [7]. So far, practice does not seem to have singled out a "best" algorithm to solve the general 1-cover problem. But in any case, those available seem to be much more efficient than solving the problem as a straight integer programming problem.

a. Every k-cover problem is equivalent to a 1-cover problem. To establish the assertion let us consider the case when k=2. The generalization to arbitrary is important but it is easier to see the proof for the case k=2.

Let us consider a bipartite graph constructed in the following manner: Start with a 2-covering problem. Let L be the number of elements in R. Take L copies of the T-nodes. Connect node r_1 to all copies of node t_j (if r_i, t_j) E of the 2-cover problem, except for the copy of t_j in the i^{th} copy of T. (See Figure 2.)

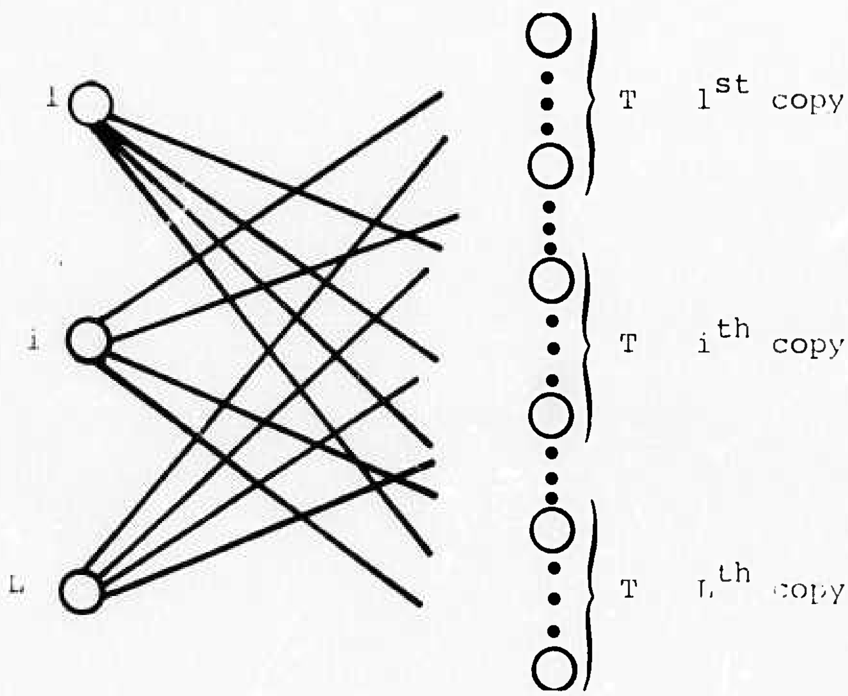


FIGURE 2

We call this problem the 1-cover problem <u>generated</u> by the 2-cover problem.

<u>Proposition</u>: Every solution to the 2-cover problem yields a solution to the generated 1-cover problem and conversely.

In particular, the optimal solution of the 2-cover problem yields an optimal solution to the generated 1-cover problem and conversely.

<u>Proof</u>: The first part (a 2-cover solution yields a solution to the generated 1-cover) is trivial. In the other direction, suppose the n^{th} copy of t_j is "covered" by node r_i the $r_i \varepsilon$ optimal 1-cover of the generated 1-cover problem, but in the i^{th} copy, t_j must be "covered" by another node in R. So every node is "covered" by at least 2-elements of R. The optimality statements follows readily.

For k-covers (k>2) a similar proposition can be found but then the number of copies of T needed (L in these case) go up exponentially. (We suspect that this type of transformation is similar to reducing general integer variables in integer programs to 0-1 variables.)

<u>Important remarks</u>: If the problem is formulated as a k-cover problem, one should realize that by the above results one might suspect that the k-cover problem is <u>much</u> more complicated than the 1-cover problem. This is corrobarated by the experience with integer programming algorithms. The only class of problems which can be solved with any form of computational success are those involving only 0 and 1's.

b. <u>The Covering Problem as an Integer Program</u>. Let A be a (T,R) matrix (where the rows correspond to terminals and the columns to repeaters). The size of A is $|T|x|R|$. We have that a_{ij} = 1 or 0 depending on whether terminal i is "visible" or not to repeater j . In terms of the bipartite graph N, the entry $a_{ij} = 1$ if there is an edge between nodes r_i and t_j, the entry $a_{ij} = 0$ otherwise. We can then formulate the k-cover problem as follows:

$$\text{Min } \Sigma x_j$$
$$\text{(I.P) such that } \sum_j a_{ij} x_j \geq k \quad i = 1,\ldots,|T|$$
$$\text{with } \quad x_j = 1 \text{ or } 0$$

This is an integer program (sometimes called a program in boolean variables). A number of algorithms to solve integer programs are known. See [7] for a survey. These algorithms offer little hope in solving the location problem arising in the Packet Radio project in that $|R|$ and $|T|$, i.e., the number of variables and the number of constraints respectively, are much too large for existing methods for k-cover problems.

c. <u>Approximate to the k-cover Method by Linear Programming</u>:

The integer programming problem formulated above can be replaced by a linear program

$$\text{Minimize } \sum_j x_j$$

$$\text{(L.P.) such that } \sum_j a_{ij}\, x_j \geq k \quad i = 1,\dots,|T|.$$

$$0 \leq x_j \leq 1$$

Where the constraint $x_j = 0$ or 1 has been replaced by the constraint $0 \leq x_j \leq 1$. The difference is obvious, the solution to (L.P.) will contain fractional values, but we note that if a variable is "profitable", it will usually be made as large as possible. The constraints $\sum_j a_{ij}\, x_j \geq k$ do <u>not</u> generate upper bounds on the x_j's. Thus one might expect many 0 and 1's in the optimal solution of the L.P. problem.

An upper bound for the optimal solution to the (I.P.) integer program can be obtained by "pushing" all the fractional x_j's appearing in the optimal solution to 1. Obviously, that yields an upper bound on Min Σx_j. The optimal solution of (L.P.) yields a lower bound.

There are many ways to improve the above solution. (without actually going all the way to integer programming). Let us suggest two.

Scheme 1. Let x^{opt} be the optimal solution of the (L.P.) problem. Set $x^*_j = 1$ if $x^{opt} = 1$ and $x^*_j = 0$ otherwise.

Let $k^*_i = \text{Max}\ \{0, k - \Sigma a_{ij}\ x^*_j\}$. Construct A* as follows: Remove from A all columns j such that $x^*_j = 1$ and remove all rows with $k^*_i = 0$. Set $k^* = \{k^*_i | k^*_i \neq 0\}$. You have now a reduced problem.

It is again an integer program.

$$\begin{aligned} &\text{Min } \Sigma x_j \\ \text{(I.P.S.1)}\quad &\text{s.t } A^* x \geq k^* \\ &x_j = 0 \text{ or } 1 \end{aligned}$$

The problem is substantially reduced in size. We can now use an integer programming algorithm.

Scheme 2. Once the solution to the L.P. is obtained. A branch-and-bound algorithm can be developed to be continued until a sufficiently small difference exists between the best solution and the smallest upper bound obtained so far. Note that by the remarks preceeding Scheme 1 it is always very easy to obtain upper and lower bounds.

There is also the possibility to use "integer cuts", see [7]. This is integer programming. However, there is a possibility that for k-cover problem, good cuts can actually be constructed and recognized. That is one direction of research (that might become imperative) but which has not been pursued. Success in characterizing cuts has been achieved in the past for highly structured problems, see [2] and [5].

d. A Network Flow Problem with Concave Cost: For the sake of completeness we record one other formulation. The problem is to find a feasible flow (in the network to be described below) at minimum cost. One possible advantage is the potential use of algorithms specifically developed for fixed charge problems or branch-and-bound methods on nets, see e.g., [9] and [14]. We consider the bipartitie graph N as described in the beginning of this section (2), Figure 3 to which we add a source s and a link S. Let f (x,y) be the blow between nodes x and y, we have the following constraints:

$f\ (x,y) \geq 0$	for all x,y
$f\ (s,y) \leq c(s,y)$	for all $y \in R$
$f\ (x,S) \geq p(x,S)$	For all $x \in T$
and f (x,y) integer	

The upper bound c (s,y) is set equal to the valence of $y \in R$ minus 1 and for all $x \in T, p(x,S) = k$. The cost $a[f(x,y)] = 0$ for all arcs except for arcs (s,y), $y \in R$ where.

$a[f(s,y)] = 0$	if $f(s,y) = 0$
$a[f(s,y)] = 1$	otherwise.

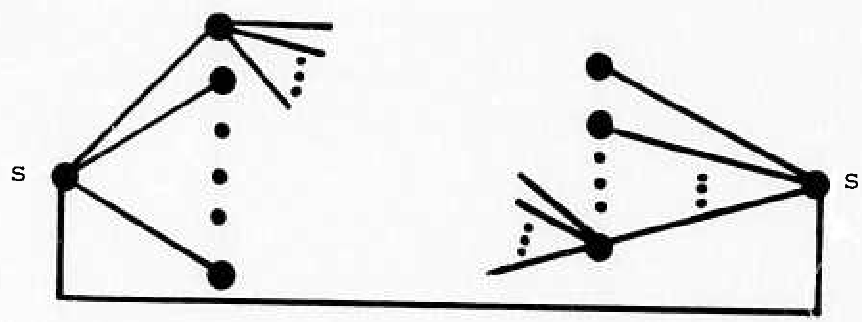


FIGURE 3

BIPARTITE GRAPH WITH SOURCE AND LINK

e. A Heuristic Approach to the k-cover Problem. Given the limited success of integer programming algorithms in solving large scale problems, we have been led to consider heuristic methods to find good solutions to the k-cover (of terminals by repeaters) problem which is typically large scale. It is intuitively appealing to consider a terminal as particularly critical if it is adjacent to few repeaters. (In the extreme cases, if a terminal has fewer than k adjacent repeaters, the problem is infeasible and if it has exactly k adjacent repeaters, all of them must be chosen for any feasible solution.) Similarly, a repeater is desirable if it is adjacent to a large number of terminals, especially if the terminals are highly critical. The heuristic algorithms described below systemizes these intuitive notions in the search of a "good" solution.

Again, consider the problem in matrix form, but this time we imbed the problem in a more general class where we can require a different cover multiple for each terminal. The more general problem is:

(I.P.H.) $\text{Min } \Sigma_j x_j$,

such that $\Sigma_j \, a_{ij} \, x_j \geq k_i, \; i = 1,\ldots,|T|$,

with $x_j = 0$ or 1,

where k_i represents the cover multiple required for terminal i. If $k_i \leq 0$, then no repeater is needed to cover terminal i.

One iteration of the heuristic method consists in passing from a problem of the type (I.P.H.) to an "equivalent" problem by fixing one of the variables, say x_s, as its upper bound 1, and putting the index s in an index set J. This implies the selection of the corresponding repeater. The new problem, denoted by $(I.P.H.)_n$ is obtained from (I.P.H.) by adjusting the matrix $A = [a_{ij}]$ and the cover requirement vector $k = [k_i]$ as follows: the column $A^s = [a_{is}]$ is deleted from A and the adjusted vector $\bar{k}$ is given by $\bar{k}_i = k_i - a_{is}$. The variable x_s no longer appears in $(I.P.H.)_n$. The algorithm terminates if at any iteration it is recognized that for the adjusted A and k, for some i, $\Sigma_j a_{ij} < k_i$, the problem is then <u>infeasible</u>; or as soon as all the adjusted k_i, i = 1,...,T, become non-positive. In the latter case, the problem is <u>feasible</u>, the adjusted problem is optimized by setting all remaining $x_j = 0$, (for $j \notin J$). A feasible solution $\bar{x}$ to the original is obtained by setting $\bar{x}_j = 0$ if $j \notin J$ and $\bar{x}_j = 1$ if $j \in J$. The vector $\bar{x}$ is called the <u>heuristic solution</u>.

The equivalence of the new problem $(I.P.H.)_n$ and the earlier version (I.P.H.) depends naturally on the choice of the variable x_s. If x_s is 1 in an optimal solution to (I.P.H.), then the two problems are equivalent in the sense that;

$$\text{Min of (I.P.H.)} = [\text{Min of } (I.P.H.)_n] + 1$$

Equivalence of the set of optimal solutions is guaranteed only if $x_s = 1$ in every optimal solution of (I.P.H.) (we must ignore x_s when considering optimal solutions to (I.P.H.)).

The selection criterion to choose a variable at each iteration x_s (or equivalently the index s) to be fixed at value 1 can be viewed as a function, called σ, of the adjusted matrix A and vector k with values in the index set {j}, i.e.,

$$\sigma: (A,k) \mapsto s \ , \ s \in \{j\}.$$

There obviously exists a function σ - a selection criterion which will guarantee the equivalence of $(I.P.H.)_n$ and (I.P.H.) at each iteration and consequently, the optimality of the resulting heuristic solution. However, applying this function σ to [A,k] might involve no less than solving (I.P.H.) We use a heuristic motivated by the considerations mentioned at the beginning of this section. The adjusted A and k are used to compute the "probability" that a given x_j belongs to the optimal solution, by this we mean that to each column of the adjusted matrix A we associated a nonnegative number ω_j such that;

$$P_j = \omega_j . (\Sigma\ \omega_j | j \notin J)^{-1}$$

represents very loosely speaking - the probability that x_j belongs to the optimal solution. We select x_s, $s \notin J$ if $P_s \geq P_j$ for $j \notin J$ or equivalently if $\omega_s \geq \omega_j$ for $j \notin J$. The selection criterion σ will be completely determined if we specify a method to compute the ω_j.

In the selection of these weights ω_j, we must take into consideration the effort involved in the computation as well as the reliability of the resultant selection. We have used four such weights:

We first define;

$$k_i^* = 1/2\ [|k_i| + k_i]$$

Observe that $k_i^* = 0$ if k_i 0 and $k_i^* = k_i$ otherwise.
We set:

$$\omega_j^{(1)}\ (A,k) = \Sigma_i \left(\frac{k_i^*}{\Sigma_\ell a_{i\ell} - k_i^*}\right) a_{ij} \qquad j \notin J$$

$$\omega_j^{(2)}(A,k) = \sum_i \left(\frac{k_i^*}{\sum_\ell a_{i\ell}}\right) a_{ij} \qquad j \notin J$$

$$\omega_j^{(3)}(A,k) = \sum_i \left(\frac{1}{\sum_\ell a_{i\ell} - k_i^*}\right) a_{ij} \qquad j \notin J$$

$$\omega_j^{(4)}(A,k) = \sum_i \left(\frac{1}{\sum_\ell a_{i\ell}}\right) a_{ij} \qquad j \notin J$$

The entries between parentheses ( ) in the definition of weights ω_j, measures how critical terminal i is. If $k_i^* = 0$, the terminal does not need covering then $k_i^*/(\sum_{j\varepsilon J} a_{ij} - k_i^*) = 0$ (for our purposes $0/\infty = 0$). On the other hand, if $\sum_j a_{ij} = k_i$, namely there are exactly enough repeaters to cover the terminal then the weight is infinite. A repeater will be preferred to another one if it covers more critical terminals.

Computational Experience: The size of test problems solved varies from problems with as few as 5 repeaters and 5 terminals to problems with as many as 400 repeaters and 400 terminals. Roughly speaking, the computation time was directly proportional to the size of the adjacency matrix A and the cover multiple required. The computer used was a PDP-10 (time sharing). The larger problems (400 repeaters, 400 terminals, 2 - cover) were solved in 70 sec. or less. The time, as may be expected, is dependent on the density of 1's in the incidence matrix A. Thus, the maximum time recorded arose from terminals - repeaters configuration where each repeater covers many repeaters. The running time is of the order of $|T| x |R|^2$ where $|T|$ and $|R|$ and the number of terminals and repeaters respectively.

We ran a number of problems with the heuristic code and for comparison with the Ophelie mixed integer programming code running on a CDC 6600 computer. The Ophelie code uses the branch-and-bound method. In the case of very simple problems (8 repeaters, 9 terminals, 2 - cover) there was essentially no difference in running time (presumably most of the time, less than .5 sec, was spent in setting up the problem). Running experience with the Steiner triples' problem described in the next section, yields a ratio of 500 to 1 between the Ophelie time and the heuristic code time when solving the smaller problem A_{27} (117 terminals, 30 repeaters, 1 - cover), and no comparison is available for the larger problem A_{45} (330 terminals, 45 repeaters) since for example the MPSX code failed to reach a solution in more than one half hour on a IBM 360-91. *

Comparison in running time is naturally not completely valid, since most of the computation time in the Ophelie code can be spent just checking if a given solution is optimal. The heuristic method does not try to check the optimality of its solution. However, in general, results with the heuristic code have been extremely good When the heuristic solution deviated from the optimal solution, the problem usually involved numerous ties for the maximum $\omega_j^{(\ell)}$ ℓ=1,2,3,4, such as in the Steiner triples' problems. In all problems that were generated to resemble the packet radio terminal - repeater problem, the heuristic algorithm reached the optimal solution (in those problems for which we are able to determine the optimal solution).

The running time was unaffected by the choice of any of the selection criteria but for "hard" problems, we obtained consistently better solutions when using $\omega_j^{(1)}$ and $\omega_j^{(2)}$ rather than $\omega_j^{(3)}$ and $\omega_j^{(4)}$.

The Steiner Triples' Problem: Fulkerson, Nemhauser and Trotter [6] report on two covering problems which they characterize as computationally difficult. In each problem, the matrix A is the incidence matrix of a Steiner triple system. The first problem, labelled A_{27} is a 1-cover problem with 117 terminals and 30 repeaters.

---

* Private Communication, R. Fulkerson, June, 1974.

The second problem, labelled A_{45}, has 330 terminals and 45 repeaters and is also a 1-cover problem. Data for both problems can be found on pages 9 and 10 of [6]. The problems are considered to be difficult because the large number of verifications (branching in branch-and-bound, cuts in cutting methods) required to establish that a given solution is in fact optimal.

In our runs, the variable to be fixed at 1 (the selected repeater) at each iteration was selected by using the criterion resulting from using weights $\omega_j^{(2)}$. In the case of ties for the maximum weight ω_j, the variable with smallest index was chosen. Due to the inherent symmetries present in these problems, numerous ties did occur. For example, all weights are equal in the first iteration. Thus, the tie breaking rule plays a relatively important role in the selection of a solution. We solved both problems 100 times breaking ties by random selection among all tied variables.

The frequency of the values generated by the heuristic solutions is recorded in the table below. In all cases the heuristic obtained the optimum solution for the smaller problem A27.

Heuristic Minima	30	31	32	34
A_{45}	3	44	29	24

The total running for 100 solutions for A_{27} (including the generation of random numbers to break ties) required less than 1/5 of the time required to solve A_{27} by branch-and-bound (even giving the optimal solution as a starting solution as recorded in [6]. Approximately 5 sec. were necessary to obtain a heuristic solution to the larger problem A_{45}. (The branch-and-bound algorithm failed to produce a solution to A_{45}. H. Ryser has conjectured that the optimal solution to A_{45} has 30 repeaters [10].

How Accurate is the Heuristic Method: Unfortunately, we can not obtain significant bounds on the error for the heuristic method using any of the weights $\omega_j^{(\ell)}$ $\ell=1,\ldots,4$ (that determine the selection criterion.) We give here an example, developed in collaboration with Professor Robert Bixby which shows that the error can be arbitrarily large. The example is a 1 - cover problem. Let $T_o,\ldots,T_n$ be disjoint sets of indices with the cordinality of $|T_i| = 2^i$. Set $T = \cup T_i$. Terminals are all pairs of indices (t, 1) and (t, 2) with $t \varepsilon T$. There are n + 3 repeaters. Repeaters R_i, i=0,1,...,n are connected to all terminals with indices (t,1) and (t,2) with $t \varepsilon T_i$. Repeaters R_{n+1} and R_{n+2} are connected to terminals with indices $\{(t, 1) | t \varepsilon T\}$ and $\{(t, 2) | t \varepsilon T\}$, respectively. For n = 3, the matrix A appears on the next page.

Observe that each row contains exactly 2 nonzero entries, thus $\sum_j a_{ij} = 2$ for all i. It is easy to verify that

$$\omega_j^{(1)} = \sum_j a_{ij} \left(\frac{1}{2-1}\right) = 2|T_j| = 2^{j+1} \qquad j = 0,\ldots,n$$

and

$$\omega_{n+1}^{(1)} = \omega_{n+2}^{(1)} = \sum_j a_{ij}(1) = \sum_{j=0}^{n} |T_j| = 2^{n+1} - 1.$$

Selecting the variable (index) with maximal weight implies that the choice will be repeater R_n. Eliminating x_n and the corresponding column as well as the rows corresponding to terminals covered by R_n, we obtain a new problem of exactly the same type as the original problem. The previous argument is independent of the value

	R_0	R_1	R_2	R_3		R_4	R_5
$(T_0,1)$	1					1	
$(T_1,1)$		1				1	
		1				1	
$(T_2,1)$			1			1	
			1			1	
			1			1	
			1			1	
$(T_3,1)$				1		1	
				1		1	
				1		1	
				1		1	
				1		1	
				1		1	
				1		1	
				1		1	

= A

	R_0	R_1	R_2	R_3		R_4	R_5
$(T_3,2)$				1			1
				1			1
				1			1
				1			1
				1			1
				1			1
				1			1
				1			1
$(T_2,2)$			1				1
			1				1
			1				1
			1				1
$(T_1,2)$		1					1
		1					1
$(T_0,2)$	1						1

MATRIX A

of n, which implies that repeater R_{n-1} will be selected next, and so on. The heuristic solution is thus given by

$$x_n = x_{n-1} = \ldots\ldots = x_o = 1 \text{ and } x_{n+1} = x_{n+2} = 0$$

It is, however, easy to see that the optimal solution is

$$x_n = x_{n+1} = \ldots\ldots = x_o = 0 \text{ and } x_{n+1} = x_{n+2} = 1$$

The value of the optimal solution is 2 whereas the value of the heuristic solution is n + 1. (The same arguments apply for all the weights $\omega_j^{(\ell)}$ $\ell=1,\ldots,4$, which we use to determine the selection criteria).

The example shows that there are problems for which this heuristic method fails to produce an optimal. By itself, that is not surprising, since it is a heuristic method with no guarantee to generate the optimal solution. But more interesting is the fact that it is possible to find problems for which the error is arbitrarily large.

Typically, however, the example is not in the category of problems that one expects to encounter in the location of repeaters for a packet radio system. There are no ties, at <u>any</u> iteration, but near-ties. At each step we almost choose R_{n+1} and R_{n+2}, in the sense that for all n, R_{n+1} and R_{n+2} would be the second choice.

f. <u>The PaAl Example</u>. One of the test problems used to compare various techniques to solve the k - covering problem, is the PaAl problem. This problem was generated by using real data obtained from a topographical map for the region of Palo Alto. This part of the U.S. was selected because it contained many interesting topographical attributes: a flat terrain (salt flats, the region surrounding the Bayshore Freeway), an urban center (Palo Alto and neighboring communities) on slightly sloping terrain and finally a hilly region (with valleys, small plateaus, etc.). Moreover, at this time, it appears that a reduced scale experiment of a packet radio network will be installed in the Palo Alto area. The purpose of this section is to give a description of the design of this model.

Location: We decided to limit the investigation to the area covered by the topographical map known as the Palo Alto Quadrangle, California, 7.5 minute series (topographic), U.S. Department of the Interior, Geological Survey or equivalently to the area lying between meridians 37° 22' 30" N. and 37° 30' N. and longitudes 122° 15' W. and 122° 07' 30" W, see Figure 6.

Terminals and Repeaters: The map was divided in 180 cells (squares) obtained by dividing the meridian direction (height) in 15 equal parts and the longitudinal direction (width) in 12 equal parts, see the map reproduced below. Each rectangular subregion measures .9356 km in height and .9356 km in width which yields a total surface area of .87533 km^2 (or approximately .35 $miles^2$). Forty two (42) locations were singled out as potential locations for repeaters. In the hilly part of the map, the Southwest region, the high-points were selected, such as top of hills, location of water towers, smaller but prominent points overlooking valleys, etc. In the city, a number of high rise constructions were singled out as potential location such as radio towers, high rise apartment or office buildings, etc.

Connections Between Terminals and Repeaters: By definition, a cell i was declared to be covered by repeater j if a terminal located at the worst possible location in that cell i was in line of sight (LOS) of the repeater j. (In some cases, it turned out that a repeater located in a given cell k covers cell j but a repeater located in cell j did not cover cell k).

<u>LOS Computation</u>: To determine if a terminal at location j can be seen from a repeater at location k, we proceeded as follows. It was assumed, that if no particular high construction (building, water tower, etc.) was available to install the repeater's antenna, it would be installed at 30 feet above the ground level (making use of a tree, telephone pole, etc.). The terminals were assumed to be 5 feet above ground level. The points were said to be in LOS if the first <u>Fresnel Zone</u> associated with transmission between these two points was free of any obstacle.

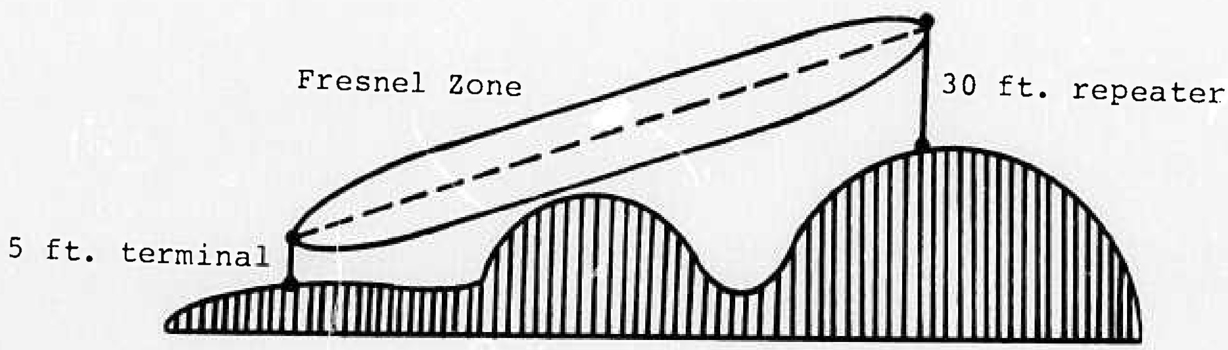


FIGURE 4

To compute the Fresnel Zones, we sssumed that transmission would occur at 1500 MHz corresponding to a wave length λ = .2m (7.87in.). We give here an example of such a Fresnel Zone, transmitter and repeater are assumed to be 5 km apart. In the figure on the next page, we give the radius for certain cross-sections of the Fresnel Zones (λ = .2m).

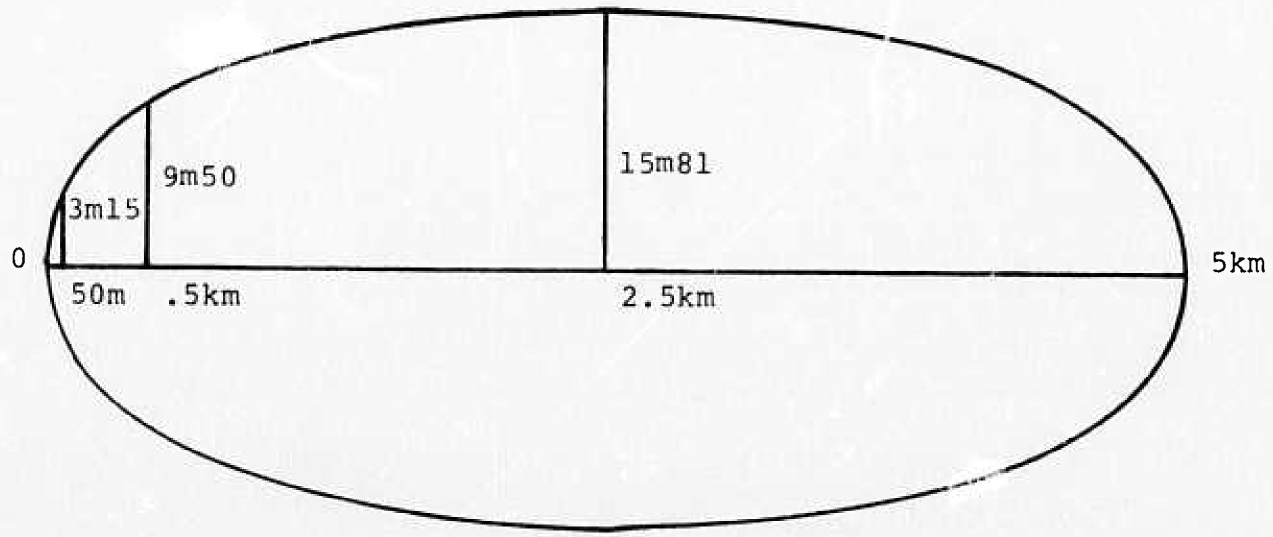


FIGURE 5

FRESNEL ZONE CROSS SECTIONS

Transmission radius is supposed to be less than 20 km (not an upper bound here since the greatest distance between any two points in this region is less than 18 km). In the urban area the maximum transmission radius was assumed to be 7 km.

Cover Multiple: In view of the fact that the region selected was a subregion of the area to be covered eventually by the packet radio network, we made some arbitrary decisions as to the boundary cells. Since they will probably also be covered by repeaters located outside the Palo Alto Quadrangle, we are requiring that these boundary cells be 1-covered rather than 2-covered as the other cells.

Computational Results: The PaAl problem, described above was solved by the heuristic algorithm, given the code name SETCOV, and by OPHELIE. (A rapid analysis of the terminal-repeater adjacency matrix shows that none of the optimal solutions would have been generated if one had used the more simplistic approach of selecting the repeater with highest adjacency degree. Such a selection yields quite different answers requiring a larger number of repeaters).

For the PaAl problem, the optimal solution requires the installation of 14 repeaters (different runs with SETCOV showed that there were in fact a number of optimal solutions with 14 repeaters). The total running time for OPHELIE was approximately 12 CPU sec. excluding set up time. The SETCOV required 3 sec. to produce a solution. The relative success of the OPHELIE code must, at least in part, be attributed to the fact that the linear programming solution (which is used to initiate the branch-and-bound part of the code) is actually the optimal solution. (If this is just an isolated phenomena to be associated with this particular problem or is actually a characteristic of this whole class of problems is not known). One optimal allocation of repeaters consists in selecting sites: 4, 5, 7, 11, 16, 17, 18, 27, 28, 30, 34, 36, 37, and 42.

We also solved a variant of the model described above. The presence of (small) ridges in cells 8, 23, and 54 combined with our model design rule - a cell is covered by a repeater if the worst location is that cell (subregion) can communicate with that repeater - renders these three cells "critical" in the sense that there are exactly one (for cell 8) and two (for cells 23 and 54) repeaters covering these cells. This results in the automatic selection of certain repeaters. To avoid this somewhat peculiar situation, we formulated a variant of the PaAl problem requiring no lower bound on the number of covers for cells 8, 23, and 54. This problem was also solved by OPHELIE and SETCOV. Running times were of the same order than before. The optimal solution only requires 12 repeaters this time. Both codes produced the optimal solution, with OPHELIE obtaining again the optimal solution in the LP part of the problem and the SETCOV using weights $\omega_j^{(3)}$. One optimal allocation of repeaters consists in selecting sites: 4, 7, 11, 17, 18, 26, 28, 30, 31, 36, 37, 42. An Ophelie solution is depicted in Figure 7.

g. <u>The Generalized k-covering Problem</u>. It is not always plausible to assume that the installation and maintenance costs associated with various repeaters at different locations is the same. This shortcoming of the previous model is overcomed by associating different cost to repeaters in the objective (of (I.H.P.)). An obvious adaptation of the heuristic method described in Section 3 replaces the weights $\omega_j^{(1)}$ (i) = 1, ..., 4 used in before by $\omega_j^{(i)}/c_j$ where c_j is the cost ($\geq$ 0) associated with repeater j.

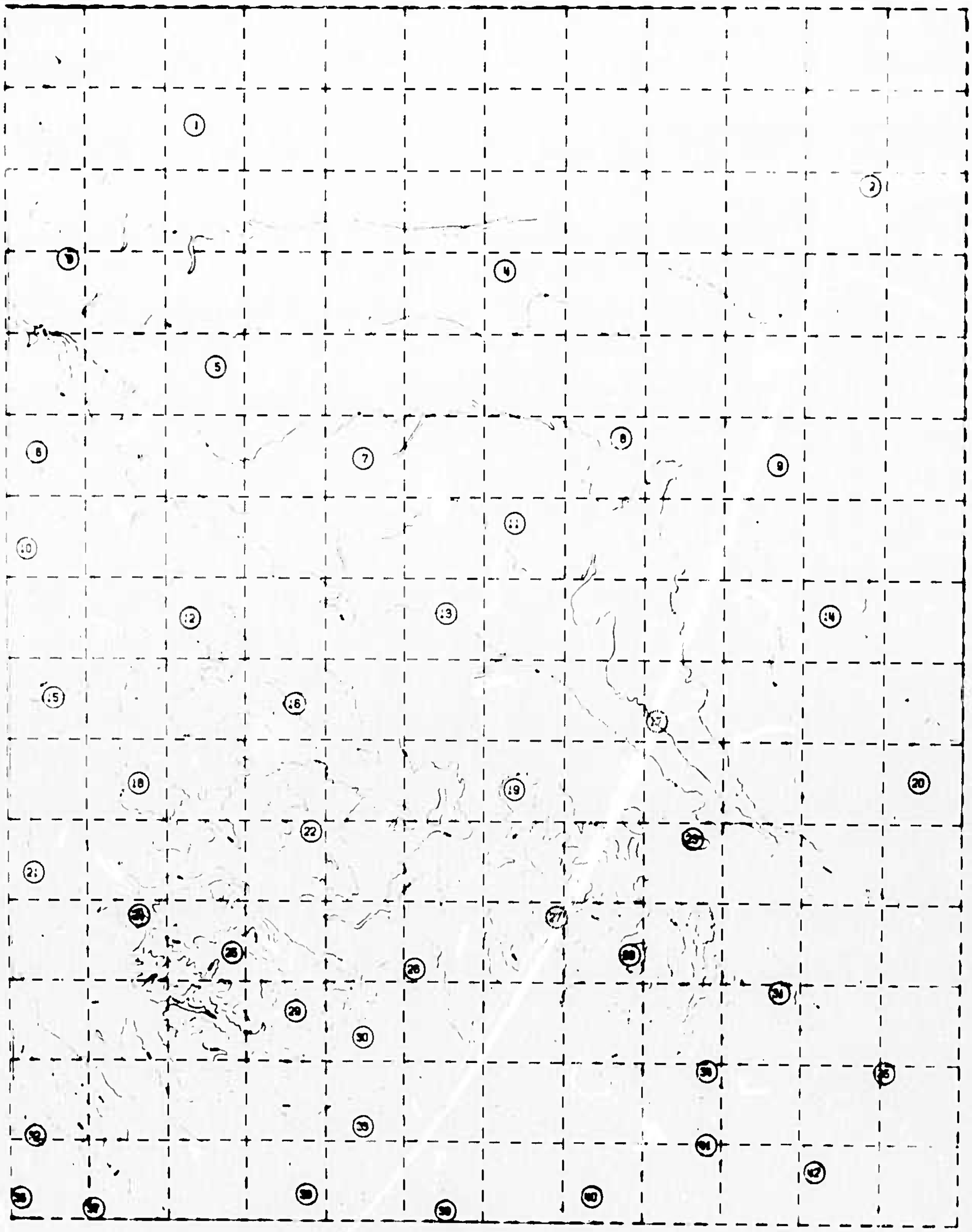


FIGURE 6

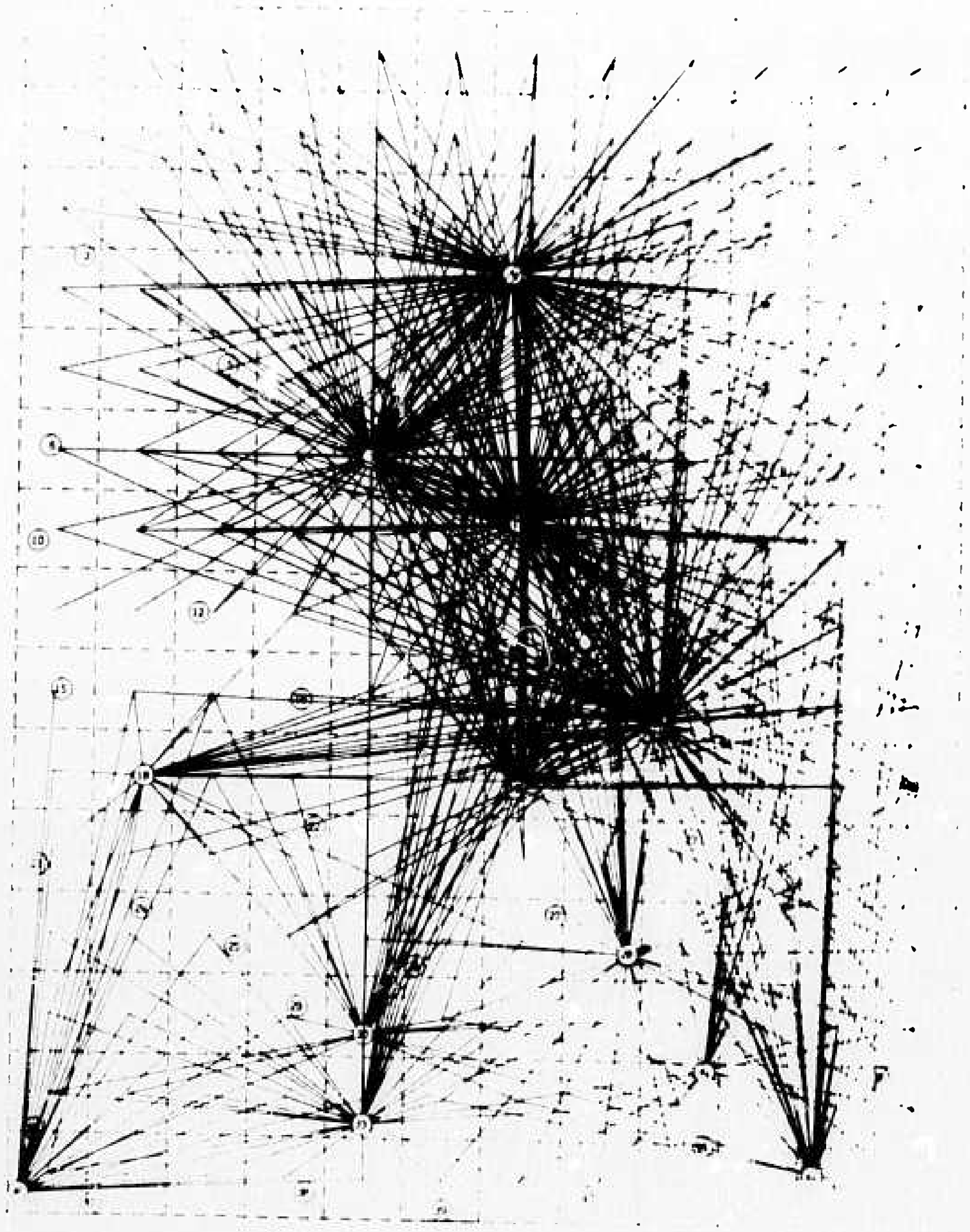


FIGURE 7

III. THE "FLAT TERRAIN" MODEL. A PROPOSED SOLUTION

a. Problem Formulation.

It is assumed that:

(i) The terrain is nearly flat

(ii) The installation (and maintenance) expenses for repeaters are independent of location.

(iii) Transmission characteristics are invariant with location. A repeater communicates with a terminal if and only if they are less than a fixed distance d_t apart. The maximum distance for communication between repeaters is assumed to be d_r (In practice d_r is substantially larger than d_t because repeater antennas are higher than terminal antennas). The area coverage by L.O.S. radio can again be separated into two parts, a covering problem and a connectedness problem (see Section 1 Problem II). Let P be the 2-dimensional plane.

Covering Problem. Find a minimal covering of P by discs of radius d_t such that every point of P is covered at least k times.

Connectedness Problem. Let G = (N,E) be the graph obtained as follows: The nodes N are the centers of the discs used in the minimal covering of P. The edges E of G are obtained by connecting two nodes if their distance is less or equal to d_r. The graph G is to be q-connected (reliability).

Since we are considering an infinite plane one can no longer define minimality of a cover in terms of its cardinality. There are various proceedures to define minimality, for example, the cover with the smallest percentage of area wasted or if $\lim_{r\to\infty} \frac{1}{r^2} \delta_C(r)$ is minimized over the space of all covers C of P that satisfy the covering and connectedness constraints, where δ_C (r) is the number of discs of C whose interior intersect a circle of radius r centered at the origin (an arbitary but fixed point of P).

b. Solutions, A conjecture. An optimal solution to the above problem is known for k = 1, $q \leq 6$ and $d_r/d_t \geq \sqrt{3}$ (that is the maximum distance for transmission between repeaters is at least 73.3% larger than that between terminals and repeaters). The problem is then the standard covering of the plane by discs of fixed radius and with least overlap. In [12], it is shown that the optimal solution is given by arrangement found below, which consists in placing a circle of radius d_t at each vertex of a regular triangular tessalation whose grid points (vertices) are $d_t \sqrt{3}$ apart. Since $d_r \geq d_t \sqrt{3}$ it follows that the resulting graph G contains as subgraph the regular triangular tesselation whose grid points (vertices) are $d_t \sqrt{3}$ apart which is 6-connected. In Figure 8 the area of the discs covered twice is shaded.

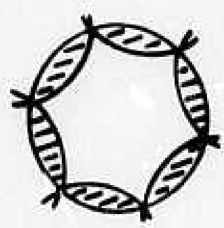


FIGURE 8

We do not consider any other cases for k = 1 since $q \leq 6$ and $d_r \geq d_t \sqrt{3}$ will always be satisfied in practice.

For k = 2, the optimal solution is not known (whatever be q and d_r/d_t). However, due to the inherent symmetry of the problem, we are ready to conjecture that the centers of the optimal solution produces a regular grid of points in P. Making this conjecture our working hypothesis, there are only three cases to consider, the three regular tesselations: (i) the tesselation of P by equilateral triangles, (ii) by squares, (iii) by hexagons [2].

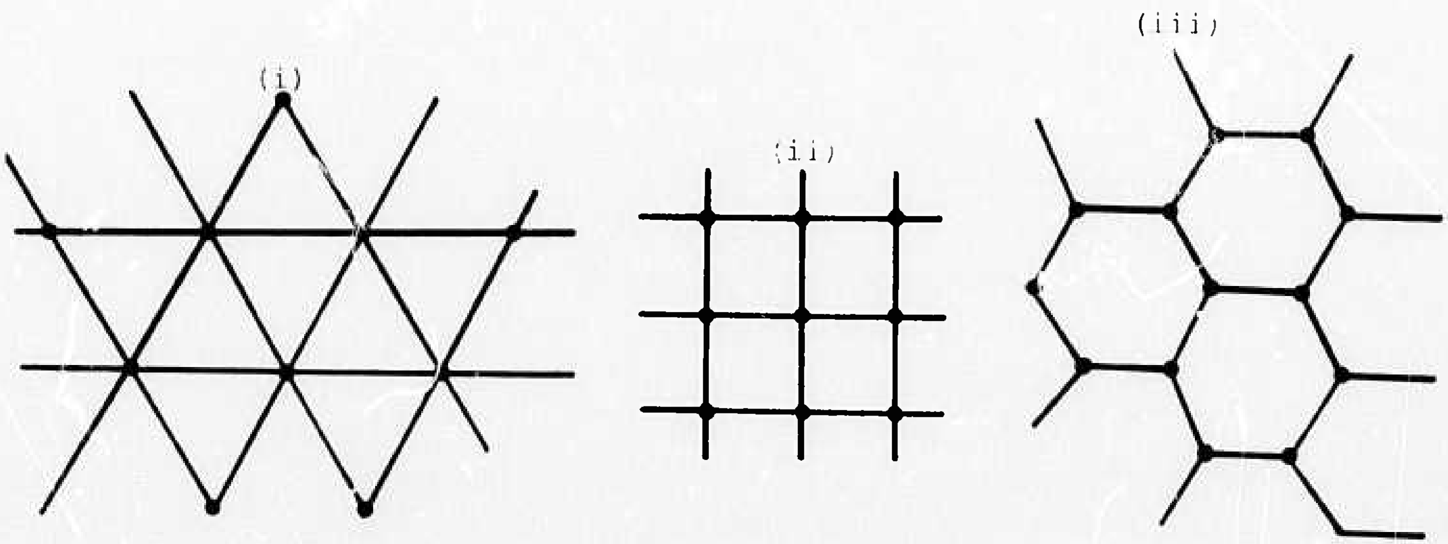


FIGURE 9 THE REGULAR TESSELATIONS

Each vertex of the tesselation corresponds to a repeater and if every point in P is to be covered twice, then the distance between two adjacent repeaters should not exceed d_t. Assuming that d_t is the distance between two adjacent vertices of the tesselations, then in the triangular tesselation one finds 6 repeaters at distance d_t, 6 repeaters at distance $\sqrt{3}d_t$, 6 repeaters at distance $2d_t$ ... from any given repeaters. In the square tesselation one finds 4 repeaters at distance d_t, 4 repeaters at distance $\sqrt{2}d_t$, 4 repeaters at distance $2d_t$ ... from any given repeaters. Finally, for the hexagonal tesselation, there are 3 repeaters at distance d_t, 6 repeaters at distance $\sqrt{3}d_t$, 3 repeaters at distance $2d_t$, ... from any repeater. With each tesselation one can associate a <u>repeater density</u>. It is easy to see that there is 1 node:

per $\sqrt{3}/2\ d_t^2$ units of area in case (i)

per d_t^2 units of area in case (ii)

per $\frac{3\sqrt{3}}{4}\ d_t^2$ units of area in case (iii)

Assuming that $\frac{3\sqrt{3}}{4}\ d_t^2$ is 1 unit of area it follows that <u>the repeater density</u> is

1.5 for triangular tesselations

1.293 for square tesselations

and 1 for hexagonal tesselations

In other words, design (i) requires 50% more equipment than (iii) and (ii) require ~30% more equipment than (iii). Obviously, if a regular grid yields the optimal solution then the one created by the hexagonal tesselation (iii) is the optimal solution. If $q \leq 3$ and $d_t \leq d_r$ then the corresponding graph G contains the subgraph given by the tesselation (iii) which is obviously 3-connected. If $d_r \geq \sqrt{3}\ d_t$ and $q \leq 6$ again the connectedness constraint is satisfied.

One should observe some inefficiencies in this covering. In particular some areas are covered by three separate repeaters rather than 2 (as would ideally be the case). If we define the thickness of a cover as being the average "thickness" of the layer of discs covering the plane, then the optimal solution for k=1 has thickness 1.209 whereas the thickness of the conjectured optimal solution for k=2 is 2.418. Thus, in both cases, we have a 21% "waste". In Figure 10, a solution is shown with the area covered three times shaded. The question for k>2 is open.

<u>Remark</u>. The "flat terrain" results yield obvious lower bounds for the "hilly terrain" problem.

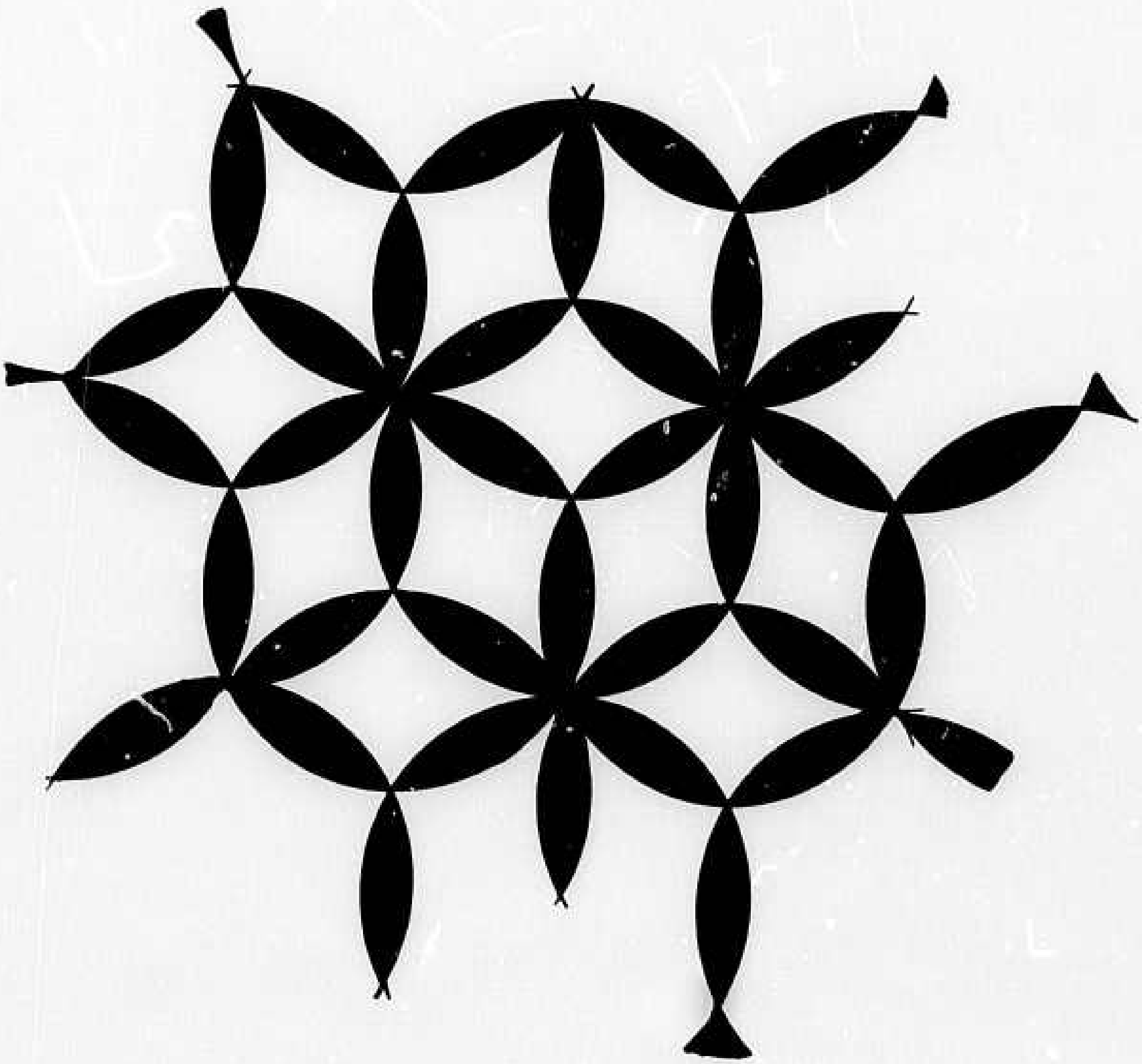


FIGURE 10

SOLUTION TO TESSELTATION OF P BY HEXAGONS

REFERENCES

1. Balinski, M. "Integer Programming: Methods, Uses, Computations", Proceedings of the Princeton Symposium on Mathematical Programming, Princeton University-Press, Princeton, 1970, pp. 199-266.

2. Coxeter, H., Introduction to Geometry, J. Wiley, New York, 1961.

3. Edmonds, J., "Covers and Packing in a Family of Sets" Bulletin American Math Soc., 68, pp. 494-499.

4. Ford, L.R. Jr. and D.R. Fulkerson, Flows in Networks, Princeton, Chapter II. 1962.

5. Fulkerson, R., "Blocking and Anti-Blocking Pairs of Polyhedra", Mathematical Programming, 1, pp. 168-194, 1971.

6. Fulkerson, D., G. Nemhauser, and L. Trotter, "Two Computationally Difficult Set Covering Problems That Arise in Computing in 1-Width of Incidence Matrices of Steiner Triple Systems," Tech. Report 903, Cornell University, Department of Operations Research, 1973.

7. Garfinkel, R. and G. Nemhauser, Integer Programming, J. Wiley, New York, 1972.

8. Hu, T.C., Integer Programming and Network Flows, Addison-Wesley, New York, 1969.

9. Martin, G., "Solving Large Scale Mixed-Integer Programs of the Fixed Cost Type", SIGMAP Newsletter, 15, 1973, pp. 27-29.

10. Ryser, H.J. Combinatorial Mathematics, Mathematical Association of America, Chapter 6., 1963.

11. Steiglitz, K.P., Weiner and D. Kleitman, "The Design of Minimum Cost Survivable Networks". IEEE Trans. Circuit Theory, CT-16, pp. 455-460, 1969.

12. Toth, Fejes L., Lagerunger in der Ebene auf der Kugel und im Raum, Springer-Verlag, Berlin, 1953.

13. Van Slyke, R., and H. Frank, "Network Reliability Analysis-I," Networks, Vol. 1, No. 3, 1972, pp. 279-290.

14. Walkup, D., "On a Branch-and-Bound Method for Seperable Concave Programs," Boeing Scientific Research Laboratory, Report D1-82-0670 - September, 1967.