12

B.S.

SEVENTH SEMIANNUAL TECHNICAL REPORT

AUGUST, 1976

INTEGRATED DOD VOICE AND DATA NETWORKS
AND GROUND PACKET RADIO TECHNOLOGY

network analysis corporation

nac

NETWORK ANALYSIS CORPORATION

Seventh Semiannual Technical Report

Aug~~ust 1~~976

(11) (12) 116 p.

For the Project

(6)

INTEGRATED DOD VOICE AND DATA NETWORKS AND GROUND PACKET RADIO TECHNOLOGY

Principal Investigator
and Project Manager:
(10) Howard Frank
~~Co-Principal Investigator~~:
Israel Gitman

ARPA Order No. 2286

Contractor:
Network Analysis Corporation
Beechwood, Old Tappan Road
Glen Cove, New York 11542
(516) 671-9580

Contract No. (15) DAHC-15-73-C-0135, ARPA Order-2286
Effective Date: 13 October 1972
Expiration Date: 30 June 1977

Sponsored by:
Advanced Research Projects Agency
Department of Defense

389 161

SEVENTH SEMIANNUAL TECHNICAL REPORT

TABLE OF CONTENTS

INTEGRATED DOD VOICE AND DATA NETWORKS AND GROUND PACKET RADIO TECHNOLOGY

NETWORK ANALYSIS CORPORATION
Seventh Semiannual Technical Report
Integrated DOD Voice and Data Networks and Ground Packet
Radio Technology
August 1976

EXECUTIVE SUMMARY

PROGRAM ELEMENTS AND OVERALL OBJECTIVES

NAC's current project, for which this Semiannual Report is an interim report, has three major components and sets of objectives:

1. Integrated DOD Voice and Data Networks -

   - The project's specific concern is to identify the appropriate mix of switching technologies (e.g., circuit, packet, and hybrid switching) that can best meet DOD data and voice communication requirements in separate or integrated future networks. To perform the analyses necessary for this determination, the project's goals are to identify key issues and parameters, to develop cost/performance tradeoffs and technology assessments, and to provide detailed recommendations for best meeting projected DOD requirements for voice and data communications in the 1980's and beyond.

2. Gateway Topological Optimization for Interconnecting Packet Switched Networks -

- Growth of different packet switching networks within the DOD is leading to a number of questions concerning the most effective means to connect these networks when necessary. A major goal of this project is to develop techniques for optimizing the number and locations of gateways required to interconnect packet networks. Additionally, these techniques will be utilized to identify fundamental parameters influencing gateway locations, and as a concrete example, gateway strategies for connecting ARPANET and AUTODIN II will be recommended.

3. Ground Packet Radio Technology -

ARPA has developed a ground packet radio system that is currently undergoing a sequence of experimental tests and performance verifications. Tests are constrained because of the limited number of repeaters which presently exist and because of the complexity of simulating all possible environments and stresses experimentally. The goals of this project are:

- To determine (via simulation and analysis) the performance profile of packet radio networks as a function of key system parameters such as maximum capacity, error rates, and equipment limitations.

- To provide specific recommendations for the enhancement of the experimental packet radio system in order to improve factors such as the time required to stabilize after element failures and system capacity under noisy conditions.

- To determine the speed and characteristics of devices within a packet radio system whose elements are mobile. To determine the range for which the present network procedures become infeasible or introduce unacceptably high performance degradations, and to propose feasible system design alternatives for use within mobile networks.

This document contains only those interim results which are both complete and of utility as stand alone items. At the completion of the contract, these results will be integrated into the final report.

Results given in this interim document are reported in stand alone chapters and are related to one or more of the major program elements with their associated tasks. Chapter 1, "Topological Design of Mixed Media Networks," is relevant to both the Integrated Voice and Data and the Gateway Projects. Chapter 2, "An Algorithm for Design of Non-Hierarchical Circuit-Switched Networks," is a major ingredient of the Integrated Voice and Data Project. Chapters 3 and 4 are part of the Packet Radio effort: Chapter 3, "On Connectivity in Mobile Packet Radio Networks," relates to system mobility issues; Chapter 4, "An Approximate Analytical Model for Initialization of Single Hop Packet Radio Networks," is part of the effort to develop a performance profile for the Packet Radio System.

The project plan for each of the three program elements is directed towards: identifying fundamental issues and problems related to overall objectives; structuring specific tasks whose completion will solve these problems; accomplishing the tasks; and integrating the results of the tasks to meet the overall program objectives. Major tasks identified are given below:

1. Integrated DOD Voice and Data Network Tasks

-1.1 Develop performance measures for packet voice and determine the impact on switch architecture (hardware and software).

-1.2 Develop techniques for the design and analysis of packet switched networks for voice and integrated voice and data.

-1.3 Investigate the impact of priority structures for voice and data in packet switched networks.

-1.4 Apply techniques developed above to AUTOVON and AUTODIN II traffic and generate performance profiles for the packet switching technology.

-1.5 Develop approaches for the design and analysis of circuit switching networks.

-1.6 Develop a methodology for assessment of circuit switch technology with regard to hardware, software, cost and capacity.

-1.7 Apply the circuit switching design methodology to DOD voice, data, and combined voice and data requirements, and generate performance profiles for the circuit switching technology.

-1.8 Develop algorithms for the design and analysis of integrated circuit/packet switching networks. (Such mixed switching strategies are called hybrid strategies.)

-1.9 Conduct an assessment of integrated switches being developed, in terms of cost, modularity, reliability, and impact of priority structures.

-1.10 Determine the cost-effectiveness of incorporating satellite subnetworks into integrated packet/circuit switched networks.

-1.11 Determine partition criteria for classes of voice and data to be served by the circuit and packet switching components of a hybrid switching network.

-1.12 Apply algorithms to AUTOVON and AUTODIN II data bases and generate performance profiles for the hybrid switching technology.

-1.13 Integrate the findings of Tasks 1.4, 1.7, and 1.12 and provide recommendations, sensitivity studies, and cost and performance comparisons between the candidate packet, circuit, and hybrid alternatives.

2. Gateway Topological Optimization Tasks

-2.1 Determine issues, parameters, and performance criteria for interconnecting packet-switched networks.

-2.2 Develop a methodology and computer programs for determining the number and location of gateways for interconnecting either terrestrial or terrestrial and satellite networks.

-2.3 Apply the methodology to a case study - the interconnection of ARPANET and AUTODIN II.

3. Ground Packet Radio Technology Tasks

-3.1 Determine the performance and capacity profile of packet radio networks for different error rates, code rates and topologies via simulation techniques.

-3.2 Develop analytical models and study network initialization time as a function of system parameters.

-3.3 Compare analytical results with simulation.

-3.4 Upgrade the packet radio simulator to contain functions and capabilities such as error correction, protocols, and packet handling techniques being implemented in the experimental system.

-3.5 Perform simulation studies for the elements included in Task 3.4 above parallel to experimental tests and provide guidelines for enhancement (e.g., capacity increases or delay reduction) of the experimental packet radio system.

-3.6 Determine the speed and mobility characteristics of devices in the packet radio system for which the present routing and initialization algorithms degrade performance or become infeasible.

-3.7 Propose routing and initialization procedures for mobile packet radio networks which will mitigate degradations.

RESULTS

CHAPTER I: TOPOLOGICAL DESIGN OF MIXED MEDIA NETWORKS: DETERMINATION OF SATELLITE AND TERRESTRIAL BACKBONE NODES

A computer methodology is described for optimizing the location of packet switches and satellite ground stations in a mixed terrestrial satellite DOD network. The techniques described, which have been programmed and tested for the examples discussed below, combine heuristic, combinatorial, and analytic elements. The algorithms developed have direct relevance to the general gateway location problem and will be used in forthcoming efforts for this problem. Results reported here include:

- Basic parameters which influence cost and performance in a mixed network are identified. Critical parameters include: Ratios of shortest path lengths within network to direct distances, line overhead, average link utilization, unit terrestrial and satellite channel costs, and average channel length. A fundamental relationship between cost and traffic requirements is developed which can provide network cost estimates, obtained without detailed layout, to within 10% of the best design. This relationship, which is an analytic combination of the basic parameters, is the first such result for packet-switched systems.

- The techniques developed are applied to the AUTODIN II data network problem. Results include:

- Network cost (excluding operations, encryption and other add on cost factors) without satellites is on the order of \$9.2 million per year. This is a significant fraction of the AUTODIN II cost.

- Satellites used in backbone network reduce communications cost by about 7% (i.e., over \$0.6 million per year) for DCA anticipated user traffic requirements. This relationship is expected to be true for other similar applications.

- Critical cost elements which influence cost tradeoffs are terrestrial line and satellite ground station costs. The satellite space segment and other costs do not have significant impact. Therefore, further reductions over 7% for the given data levels are achievable primarily through reduced cost ground stations and/or lower tariffs.

- For this application, the best number of satellite ground stations is small and constant (at 4) as the number of backbone packet switches increase from 7 to 16. We expect that this will be true for other applications unless ground station cost is significantly lower (e.g., 1/5 to 1/10 of current costs).

- For this application, and for designs with the same number of backbone nodes, those with lower local access cost nearly always have lower total cost (i.e., minimum local access cost nearly always implies minimum total network cost).

CHAPTER 2: AN ALGORITHM FOR DESIGN OF NONHIERARCHICAL CIRCUIT SWITCHED NETWORKS

A general computerized procedure for the minimum cost design of circuit switched networks is reported and studies to be conducted with the procedure are described. The procedure, which was developed to enable the performance of these studies, needed because no organization has previously required efficient tools with which to examine all of the basic assumptions involved in traditional circuit switched network design unfetered by the constraints that have evolved piecemeal for existing circuit switched systems such as the U. S. telephone network. In this chapter we report on the design of the procedure. Future studies with the system will identify cost performance tradeoffs for voice and data as a function of traffic load and mix, routing strategy, and structure of the circuit switch (switch set up time, signalling scheme, etc.)

Relevant features of the procedure are:

- Computationally efficient and thus useful for a wide range of studies, including investigation of switch design, signalling scheme, and network layout.

- Inputs are switch locations, traffic requirements, and system constraints such as available tariffs and desired user call rejection probability.

- Outputs are network link layout, link capacity assignment, routing plan and network cost to meet traffic requirements and constraints.

- Contains separate modules to enable performance analysis, alternate routing, trunk sizing, and line layout. Thus, useful for studying impact of variations in switch and network operating procedures.

CHAPTER 3: ON CONNECTIVITY IN MOBILE PACKET RADIO NETWORKS

This chapter reports the first results on Packet Radio System performance with mobile elements. An important issue for such networks is the amount of overhead data that must be sent through the network for the routing and control procedures to operate. Speed becomes a factor that can degrade network performance because of this overhead. Performance can eventually deteriorate to the point where operations become infeasible. Consequently, it is important to design networks which exhibit graceful degradation over as wide a range as possible.

The techniques described in this chapter will be used in forthcoming efforts to study performance of current and proposed network operating rules to identify mobility factors leading to infeasibility. Procedures are developed to calculate the following quantities in systems whose element trajectories are known or can be computed:

- Times during which a repeater can communicate with the station.

- Time interval over which the entire network is connected (i.e., all repeaters have paths to the station).

- Connectivity profile of the network (time spans when "at most," "at least" or "exactly" a specified number of repeaters are unable to communicate with station).

The trajectory computability assumption will be relaxed in future studies and the performance projections currently being calculated will serve on bounds on system performance.

CHAPTER 4: AN APPROXIMATE ANALYTICAL MODEL FOR INITIALIZATION OF SINGLE HOP PACKET RADIO NETWORKS

This chapter deals with initialization in Packet Radio Systems and is part of the effort to develop a performance profile of the system. An analytic model is first developed to calculate initialization times (the time required to stabilize the network given a disturbance such as the entry or failure of repeaters). The model is then used to identify efficient operating parameters such as those listed below for the Packet Radio System. Future efforts will extend the model to general packet radio systems and be used to investigate the effect of initialization procedures on the time required for network stabilization under various operating conditions. Current numerical results include operating parameters for:

- Station transmissions of connectivity information (e.g., a label transmission every four maximum packet times is most effective).

- Repeater initialization packets (called ROP's) (e.g., if the network has M repeaters, a repeater should transmit a ROP every m(e) maximum packet times).

CHAPTER 1

TOPOLOGICAL DESIGN OF MIXED MEDIA NETWORKS: DETERMINATION OF SATELLITE AND TERRESTRIAL BACKBONE NODES

CHAPTER 1

TABLE OF CONTENTS

PAGE

CHAPTER 1

TABLE OF CONTENTS: FIGURES

PAGE

CHAPTER 1

TABLE OF CONTENTS: TABLES

CHAPTER 1

TOPOLOGICAL DESIGN OF MIXED MEDIA NETWORKS: DETERMINATION OF SATELLITE AND TERRESTRIAL BACKBONE NODES

1. INTRODUCTION

In this chapter, we address the topological design of large distributed data networks in which both terrestrial and satellite technologies are employed - which we called mixed media data networks. The particular problem considered is the determination of terrestrial and satellite backbone switch locations which minimize total network cost.

In the following, we first briefly consider the various options in packet switched satellite communication. We then consider the general topological design problem of large distributed networks, and indicate how the switch location problem ties into the general design problem.

1.1 Packet-Switched Satellite Communication

Until now, the packet-switched networks implemented use mainly terrestrial communications links. However, with the advancement of satellite communication technology, the possibility of using satellite links to reduce total network cost and increase network growth flexibility has been receiving increasing attention [ABRAMSON, 1973], [GERLA, 1974b], [HUYNH, 1976], [KLEINROCK, 1973], [LAM, 1974], [ROBERTS, 1973].

Satellite communications can be used in the following modes in a packet-switched environment. (Here, we only consider alternatives at the packet-switched backbone level.)

1. Dedicated point-to-point connections between earth stations at some of the backbone node sites, replacing more costly terrestrial channels. This mode is not very attractive unless there are very few earth stations (say 2 or 3).

2. Dedicated up-links from earth stations located at some of the backbone node sites to the satellite, and broadcast down-links from the satellite to all earth stations.

3. Multiple-access (ALOHA, slotted ALOHA, reservation) up-link from the earth stations to the satellite channel, and broadcast down-links from the satellite to the earth stations. This mode is attractive when the users (earth stations) have different busy hour, or when there is a cost-effective high bandwidth satellite channel option.

1.2 Design Approaches for Large Distributed Data Networks

Large distributed networks generally result from the integration of existing centralized and/or distributed data communications systems. The purpose of integration is to achieve line economies, intersystem communications capability, higher network bandwidth, improved reliability, improved growth flexibility, and more efficient network control and management [NAC, 1976]. Consolidation and integration of communication requirements is presently being considered by several large corporations as well as government and military agencies. For example, the most substantial packet network under development is the AUTODIN II System whereby the Department of Defense is in the process of integrating many of its ADP systems into a large, hierarchical network.

Although specific network implementations vary considerably depending on the application, the common structure of large distributed networks is the multilevel hierarchical structure with a backbone network at the higher level, and local access networks at the lower levels. Backbone and local access nets may be further subdivided into hierarchical sublevels.

The backbone network is characterized by distributed traffic requirements and is generally implemented using the packet-switching technology. Local access networks, on the other hand, have a centralized traffic pattern (all the traffic is to and from the gateway backbone node) and are, therefore, implemented with conventional teleprocessing techniques such as multiplexing, concentration, and polling.

The selection of the most effective network architecture (i.e., number of levels and type of access at each level) is the first problem that must be attacked in the design of a large network. A discussion on alternative architectures can be found in [NAC, 1976].

Having selected the architecture, we must then design a cost-effective topology that minimizes line and nodal processor costs within such architecture. For the multilevel case, the problem may be subdivided into four subproblems:

a. Preliminary clustering of the user installations;

b. Selection of backbone nodal processor locations;

c. Local access design;

d. Backbone topology design for the higher level network.

One approach for the topological design of mixed media networks is, in subproblem b of the above, determining not only the backbone nodal processor locations, but at the same time which of them are satellite switch locations. However, as indicated in Section 3.2, the tradeoffs involved here are quite complex. Thus, we are content with only determining the backbone switch locations in subproblem b, and then selecting the satellite switches from the set of backbone switches in subproblem d. Consequently, subproblem d is further broken into the following two parts:

d.1 Selection of satellite switch locations;

d.2 Topology design of the satellite subnet (satellite channel and earth stations) and terrestrial subnet.

Subproblem a, the partitioning of a large terminal population into "minimal cost" clusters satisfying given constraints is a classical problem in data network design. An efficient technique for clustering a population of multidropped terminals was presented in [MCGREGOR, 1975]. Similar techniques may be constructed for other local access strategies.

Subproblem c, the design of local access topologies, has also been well-studied. Several algorithms are found in the literature [CHOU, 1973], [ESAU, 1966], [WOO, 1973]. Typically, the algorithms solve the problem of optimal location of concentrators, and optimal layout of multidrop lines to connect terminals to concentrators.

Subproblem c, the selection of the number and location of backbone switches has recently been studied extensively for the terrestrial network design in [NAC, 1976]. Two algorithms: ADD and CLUSTER, are proposed. In this chapter, we propose a simplified backbone cost estimate for the ADD algorithm (Section 3.2), and apply the simplified ADD algorithm to the backbone switch selection problem for mixed media networks. It is experimentally demonstrated that the cost function proposed is a fairly good approximation to that used in [NAC, 1976], and the saving in computation time is significant.

Efficient heuristics for the topological design of terrestrial backbone networks have been reported in [FRANK, 1972], [GERLA, 1974a], [LAVIA, 1975]. Once the satellite switch locations have been determined, these procedures can be adopted to subproblem d.2, the design of mixed media backbone networks [GERLA, 1974b].

In this chapter, a simple cost function which formulates subproblem d.1 and an efficient exhaustive search algorithm are developed. Experimental results demonstrate that the cost function used is a good approximation to the total network cost. Extensive experimental results which study total network cost as a function of number of earth stations, traffic levels, and sensitivity to cost variations are also presented.

The work presented in this chapter is within the effort of the general topological design of interconnecting communication networks. One of the subproblems in the latter is to select, in each network to be interconnected, a subset of nodes for "gateway" nodes. It is expected that the algorithm presented for solving subproblems b and d.1 will be used for the gateway selection problem, using a different cost function.

2. REVIEW OF TERRESTRIAL BACKBONE SWITCH LOCATION ALGORITHMS

In the previous phase of ARPA contract [NAC, 1976], we have investigated the switch location problem for terrestrial packet-switched networks. Briefly stated, the problem is; given user traffic requirements, facility cost functions, and candidate sites for backbone switches, determine the number and location of backbone switches so that with appropriate optimum backbone link topology, the overall communication cost is minimized while satisfying constraints on backbone delay, switch capacity, etc. The basic underlying assumptions are as follows:

1. The user sites are preclustered, so that each site is connected to a backbone switch directly.

2. Only one type of backbone switch is used.

3. Only one type of backbone trunk is used (though several trunks in parallel may connect the same two switches).

Two solution procedures, MODULARIZED ADD and CLUATER, have been proposed in [NAC, 1976]. Of the two, ADD is found to consistently give better results, though it is also more time-consuming. The following guidelines apply in selecting the most effective algorithm: if the number of candidates is much smaller than the number of user sites, the ADD algorithm is the best choice; if, on the other hand, the number of candidate sites is very large (e.g., the candidate site set consists of user sites), the CLUSTER algorithm is preferrable.

2.1 Backbone Network Cost Estimation

Since the optimal node location strategy is the result of the tradeoff between local access cost and backbone cost, the backbone topology should in principle be redesigned at each iteration to evaluate the backbone cost. This approach, however, is computationally too time-consuming, especially if the switch location algorithm requires at each iteration the design and comparison of several backbone configurations, one for each candidate node selection. We need, therefore, an approximate cost estimate which is both computationally efficient and consistent (in the sense that the error introduced is "systematic", without severe jumps corresponding to perturbations in switch number and location). The estimate need not be very accurate in the absolute sense, since we are considering only cost variations relative to insertion or removal of one node at a time. This estimate, and the procedure used to obtain it, can play a major role in the overall problem of the characterization of networks via simple parameters for use in the internetwork gateway location problem.

We first introduce the concept of nondirect routing penalty P. The route R used by the traffic from i to j depends on the topology, the traffic conditions, etc. However, we would expect the mileage on R to be proportional to the distance between i and j, and, therefore, may approximate the length |R| as:

$$|R| = P \times d_{ij} \tag{1}$$

Where P is the "nondirect routing penalty" defined as the ratio between the shortest path distance and the direct distance, for the average source-destination pair (P>1), and d_{ij} is the direct distance from i to j.

With this estimate, the backbone line cost D can be expressed by:

$$D = \sum_i \sum_j \frac{c r_{ij} d_{ij} (1+b) P}{\rho} + NN \times F \qquad (2)$$

where:

ρ = Average link utilization ($\rho \leq 1$),

F = Fixed cost per node (= average number of line terminations per node x line termination cost),

c = Cost/mile x unit bandwidth x month,

r_{ij} = Traffic requirement from i to j,

b = Line overhead (protocols, etc.), ($0 \leq b \leq 1$),

NN = Number of switches.

The coefficients P, ρ and F are determined experimentally for various values of NN. Application results show that such coefficients depend solely on NN, and are insensitive to switch relocations [NAC, 1976]. Moreover, experimental results [NAC, 1976] show that the estimate, Eq.(2), is adequate for the purpose of switch site optimization.

In the switch location problem for the mixed-media networks, due to the added complexity of the backbone transmission cost tradeoffs, instead of the estimate developed in this Section, a much simpler estimate is adopted (see Section 3.2). However, the transmission cost estimate developed here is used in the selection of satellite switches (Section 4).

2.2 ADD Approach

In the ADD approach, a "figurative" switch (center-of-mass) is created (and dynamically relocated), such that each user that is not yet assigned to any selected switches is assigned to this node. Each remaining switch candidate location is then evaluated by determining the cost reduction which would be achieved by placing a switch at the location and profitably assigning some of the unassigned users to that switch. The cost tradeoff is based on the three cost components: backbone line cost (which is estimated by Eq.(2)), local access cost, and switch cost. The location giving the greatest cost reduction is then selected as the next switch site, and the user sites contributing to its selection are assigned to it. When no further cost reduction can be achieved, the process halts, and the switch candidate location closest to the "figurative" switch is selected as the location for the last switch.

The ADD approach can be improved (at the expense of increased computation time) by considering each switch as composed of several modules. During each iteration of the ADD procedure, instead of adding one more switch, only one more module is added to the backbone. We call this the MODULARIZED ADD procedure. The module cost can be made such that the i-th module at a location costs only a fraction of the (i-1)-th module at the same location. This modification is particularly appropriate when the backbone switches are very powerful but relatively inexpensive as compared to the other network components. Experimental results [NAC, 1976] indicate that the MODULARIZED ADD approach does generate much better results.

Below we develop in some detail the cost trade off evaluation for selecting each new switch. Suppose k candidates $C_1,\ldots,C_k$ have already been chosen as backbone switches (k may be 0, in which case no switch has been chosen yet).

If all user locations are assigned to one of the C_i's, $i = 1,\ldots,k$, then we are done. Suppose on the other hand, that not all the user locations are assigned. Let,

Φ_0 = Set of user locations not yet assigned.

If we select only one more backbone switch, then a reasonable choice for this node, C_0, is the traffic-weighted center of mass of the nodes in Φ_0.

Let Q be a backbone candidate not yet used. If we use Q as C_{k+1}, then we would have to take some of the nodes in Φ_0 (nodes not yet assigned) away from C_0, and assign them to Q. Thus, the basic tradeoff is between homing nodes to Q, or homing nodes to C_0.

Let T be a user site in Φ_0. The saving of homing T to Q, $S_T(Q)$, is given by:

$$S_T(Q) = LAS_T(Q) + BLS_T(Q)$$

where;

$LAS_T(Q)$ = Local access saving of homing T to Q.

$BLS_T(Q)$ = Backbone line saving of homing T to Q.

The local access saving is given by:

$$LAS_T(Q) = K_T \times [d(T,C_0) - d(T,Q)]$$

where;

K_T = Mileage cost of the local access line connecting T to the backbone.

The backbone line saving can be estimated, with Eqs. (1) - (2) and other simplifying assumptions [NAC, 1976], by:

$$BLS_T(Q) \approx t_T \times \frac{C_B}{t_{Total}} \times \sum_{i=1}^{k} t_i \times [d(C_0,C_i)-d(Q,C_i)]$$

where;

t_T = Total traffic (sum of transmit and receive traffic) at user site T.

t_i = Total traffic of users assigned to switch C_i, $i = 1,\ldots,k$.

t_{Total} = Total traffic.

C_B = Cost factor as defined by the approximation method ($C_B = c \times P/\rho$, where c, P and ρ are as defined in Section 3.2).

We proceed by assigning to Q the user sites with positive savings, starting from the site with the largest saving, until the switch capacity is fully utilized. The saving of selecting Q as C_{k+1}, S(Q), is then given by;

$$S(Q) = [\sum_{\substack{T \text{ assigned} \\ \text{to } Q}} S_T(Q)] - C_F,$$

Where C_F is the fixed cost of adding a backbone switch.

After evaluating S(Q) for all the switch candidates Q, the candidate with the largest positive saving is selected as the $(k+1)^{-th}$ switch, the process continues until no candidate can give a positive saving or if all users are assigned. In this case, the candidate nearest to the center C_0 is selected as the last switch location.

2.3 CLUSTER Approach

The CLUSTER approach works to determine, for a given value of N, the best set of N locations at which to place the switches. The basic underlying assumption is that the backbone switch locations that minimize local access cost also minimize total network cost. Based on the experimental evidence reported in Section 5.2, this assumption is quite valid. The switch location problem is thus reduced to one of optimally partitioning the user sites into N subsets. This is heuristically accomplished by forming larger and larger clusters. The clusters are formed by "rolling snowballs" in a rather "balanced" fashion. First, the two nodes closest together are selected. These nodes are then replaced by a single node at their "center-of-mass" with the combined weighting factors of the first two nodes. The merging process continues on a closest node pair basis until only N nodes remain.

In practice, the approach outlined above is too simple to produce a set of N good locations for the backbone switches. The primary deficiency is the possible dramatic difference in size of the clusters, which in practical situations usually results in poor designs. To compensate for this tendency, a parameter Z is defined as a cluster capacity, and a parameter α is defined as a size threshold expressed as a percent of Z. Clusters will be grown until they reach a size greater than αZ and then they will be stopped. Clusters are not permitted to merge if the merger exceeds Z. The first N cluster to be stopped will be selected as the backbone nodes.

By not permitting the size Z to be exceeded, the process of developing clusters may stop prematurely for lack of feasible mergers. In this case, simply the N largest clusters are used. Furthermore, mergers may occur over extraordinary distances because of feasibility issues. This is certainly not desirable, and so a parameter of maximum distance between mergeable nodes is introduced. Finally, if modular capacity switches are available, then several clusters in the immediate vacinity of one another are probably better served by one large switch. Consequently, a parameter is available to define a minimal separation between clusters. Clusters closer than this minimum distance are combined to be served by one switch.

3. LOCATING BACKBONE SWITCHES IN A MIXED MEDIA NETWORK

In this section, we consider a model for the mixed terrestrial-satellite network, and investigate solution approaches to the corresponding backbone switch location problem.

Our network model is as follows:

1. The terrestrial network consists of a set of store-and-forward switches (e.g., IMP's) interconnected by ground channels (a distributed subnet). For reliability, the terrestrial net is usually required to be 2-connected.

2. The ground stations are colocated with the satellite switches. (This assumption is used here for convenience. However, it is not required for the satellite switch selection procedure developed in Section 4.) Moreover, the satellite switches form a subset of the set of store-and-forward switches.

3. The satellite system is used either in a dedicated access/broadcast mode, or in a multiple access/broadcast mode. Thus, the satellite subnet effectively forms a complete graph.

4. All the switches (regular and satellite) have the same capacity and cost.

5. The earth station cost is fixed, while the satellite channel cost is proportional to the bandwidth.

6. The low delay traffic which cannot be routed over the satellite links constitutes only a small portion of total traffic requirement, and has higher priority.

3.1 Backbone Transmission Cost Estimate For Mixed Media Networks

Based on the terrestrial line cost estimate developed in Section 2.1, we can approximate the backbone network cost for a mixed media network as follows: For any two switches A,B in the network, let

$C_{A,B}^{TER}$ = Cost of routing a unit flow from A to B along terrestrial links,

$C_{A,B}^{SAT}$ = Cost of routing a unit flow from A to B through some satellite links,

S(A) = Nearest satellite switch to A. (S(A) may coincide with A.)

Then by Eqs. (1) and (2), $C_{A,B}^{TER}$ can be estimated by

$$C_{A,B}^{TER} = C_T \times d(A,B) + \text{(estimated unit line hardware cost for links on routes from A to B)}, \qquad (3)$$

where C_T is the backbone line cost approximation factor as given in Section 2.1 (in fact, $C_T = c.P/\rho$, where c, P and ρ are defined in Section 2.1). The unit line hardware cost can be calculated by estimating the number of links on an average path between A and B. Since the hardware cost usually constitutes only a small fraction of the total line cost, a very rough estimate suffices.

Specifically, in the present study, the hardware cost is estimated in the following fashion. Similar to the heuristic estimate developed in Section 2.1, for a given set of backbone switches and requirement matrix, we estimate the channel-miles and the number of channels required for a terrestrial backbone network. From this we obtain an estimate for the average channel length, L_C. The number of links on an average path between two switches A and B is then estimated by $P \times d(A,B)/L_C$ (P is defined in Section 2.1), from which the unit line hardware cost for paths between A and B follows.

The unit cost of routing through satellite links can be estimated by (see Figure 1)

$$C^{SAT}_{A,B} = C^{TER}_{A,S(A)} + C^{TER}_{B,S(B)} + C_S, \tag{4}$$

where C_S is the unit satellite bandwidth cost.

To a first degree approximation, for the low priority requirement, the route with the lower cost is preferred (see Figure 1), and thus, the unit cost for routing from A to B, $C_{A,B}$ is estimated by

$$C_{A,B} = \min \{C^{SAT}_{A,B}, C^{TER}_{A,B}\} \tag{5}$$

3.2 An Approach for Locating Backbone Switches

With Eqs. (3)-(5), we can extend the MODULARIZED ADD algorithm to select the satellite and terrestrial switch locations. At each iteration of the ADD algorithm, we not only can select one more backbone switch location, we can also determine whether it is profitable to place an earth station at that location. An approach to select the switch is as follows: First, compare the (remaining) candidates assuming that they are equipped with earth stations. Let the best candidate chosen be Q_1 and corresponding saving S_1. Then, compare the same set of candidates assuming that they are not equipped with earth stations. Let the best candidate chosen be Q_2 and corresponding saving S_2. The Q_i with the larger saving is selected as the next switch location (with or without an earth station at the location depending on whether i is 1 or 2).

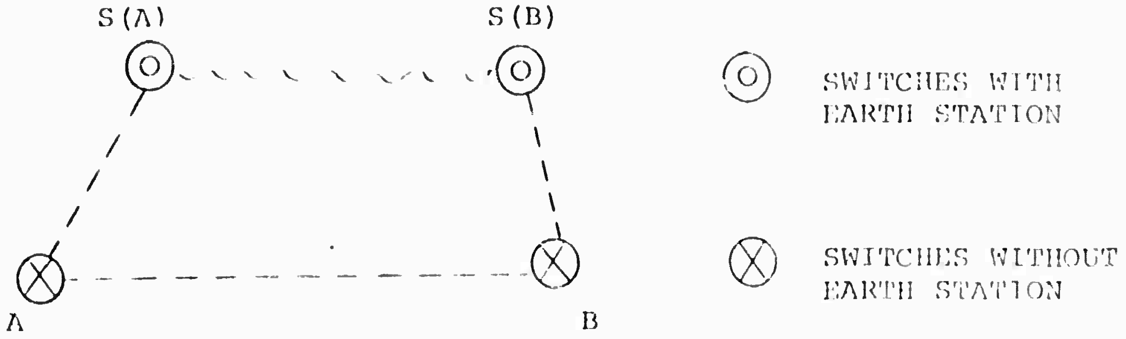


FIGURE 1: ROUTING ALTERNATIVES IN A MIXED NETWORK

There are some inherent weaknesses with this approach, aside from computational complexity considerations. For example, during some iteration, a switch location Q without an earth station may be selected on the basis of the saving criterion. However, when more switches are selected later, it may become more profitable to have an earth station at Q. Yet, this information was not available during the selection of Q.

Alternatively, one can use some very simple, but fast estimate for the backbone line cost. The following is an outline of one such approach. Steps 1 and 2 are used to estimate the incremental backbone line cost.

1. Based on the given user traffic matrix and design constraints, generate a number of reasonably "good" mixed terrestrial-satellite backbone network designs.

2. Based on the designs obtained in Step 1, construct a function which approximates the backbone line cost relative to the number of backbone nodes. Let this function be BC(N). Also, let

$$\Delta BC(N) = BC(N+1) - BC(N)$$

be the incremental cost function.

3. Use the ADD algorithm to select the backbone switch locations with the following modification: Suppose K switches have already been selected. Then the saving that can be achieved by selecting a candidate Q as the (K+1)-th location is

$$S_K(Q) = \begin{pmatrix}\text{local access saving of selecting } Q \text{ as} \\ \text{the } (K+1)\text{-th switch location}\end{pmatrix} - \begin{pmatrix}\text{fixed cost of installing a switch at} \\ Q\end{pmatrix} - ABC(K).$$

4. Suppose switch locations $C_1, \ldots, C_N$ are selected by the modified ADD algorithm. Determine which of the C_i's are to be equipped with earth stations so as to optimize the backbone network cost.

Based on experimental results presented in [NAC, 1976] and in Section 5.2, we found that for terrestrial networks:

- The number and location of backbone switches are rather insensitive to variations in unit backbone cost.

- The best selection made by the basic ADD algorithm given in Section 2.2 is only slightly better than the best selection made by the simplified scheme described in Steps 1-3.

We thus conclude that for a general backbone network (terrestrial, mixed terrestrial-satellite, hybrid packet-circuit, etc.), the procedure based on Steps 1-3 is sufficient to obtain a good selection of backbone switches.

Step 4, the optimization of number and location of satellite switches, is in itself a rather difficult problem. It is discussed in the next section.

4. SATELLITE SWITCH SELECTION

In this section, we consider the problem of selecting the satellite switch (and thus the associated ground station) locations from among the backbone switch locations. More formally, the problem is as follows:

Given:

- Switch locations.
- Switch-to-switch traffic requirement.
- Ground station cost (including the cost of connection to the terrestrial network).
- Satellite bandwidth cost.
- Satellite access technique.
- Terrestrial bandwidth cost.

Optimize:

Total communication cost D:

D = ground station cost + terrestrial trunk cost + satellite bandwidth cost.

Over the Variables:

- Number and location of satellite switches (ground stations).
- Terrestrial network topology.

Such that:

- Traffic requirements are met.
- Appropriate constraints (delay, reliability, etc.) are satisfied.
- Switch capacity constraints are met.

To simplify the solution procedure, we use the following assumptions; they can be relaxed by modifying the basic procedure appropriately:

- Switch capacity is unlimited. This is quite realistic if we use modularized switches such as the Pluribus IMP [HEART, 1973].
- Switch cost varies linearly with capacity, and the fixed cost portion dominates. Again, this is a reasonable cost model for the modular type switch.
- The traffic is assumed to consist of two priorities; the average delay for the high priority traffic is smaller than the propagation time over the satellite channel and hence must be accommodated by the terrestrial subnet. The low priority traffic can tolerate satellite propagation delay.

In the solution procedure, the delay of the low priority traffic is not explicity taken into account; it is implicitly used in the average link utilization in deriving the cost of routing a unit flow over a channel.

4.1 Basic Solution Approach

From the transmission cost estimates developed in the last section (Eqs.(3)-(5)), we can approximate the backbone cost for any switch set configuration. This forms the basis of our exhaustive search algorithm.

First, notice that $C^{TER}_{A,B}$, the line cost of sending a unit flow from A to B, can be regarded as a constant independent of the satellite switches selected.

Now let Φ be a subset of the switches. Suppose the nodes in Φ are used as satellite switches. Then, the satellite transmission cost for a unit flow between A and B, $C^{SAT}_{A,B}(\Phi)$, can be calculated from Eq.(4). The unit transmission cost, $C_{A,B}(\Phi)$, can then be calculated from Eq.(5). It follows that the total backbone communication cost (excluding the switch cost) for the low priority traffic is:

$$C(\Phi) = |\Phi| \times C_{GRD} + \sum_{A,B} t_{A,B} \times C_{A,B}(\Phi), \qquad (6)$$

where

C_{GRD} = Ground station cost,

$t_{A,B}$ = Low priority requirement from A to B.

The optimum satellite switch set can thus be determined by selecting the collection Φ with minimum cost $C(\Phi)$.

For a given set of N switches, the total number of possible subsets is 2^N. Also, for each subset Φ, the computation of $C(\Phi)$ requires on the order of N^2 operations. Thus, the complexity of the above procedure is on the order of $N^2 \times 2^N$. Since for most practical problems, N is small (on the order of 10), thus the computation of the optimum subset Φ is feasible. Moreover, as will be seen in Sections 4.2 and 4.3, there are various ways to reduce the computation.

4.2 Cyclic Subset Ordering

Suppose Φ and Φ' are two sets of switches such that Φ' is obtained from Φ by deleting a switch. We investigate how to compute $C(\Phi)$ $\{C_{A,B}(\Phi)\}_{A,B}$, and $\{C^{SAT}_{A,B}(\Phi)\}_{A,B}$, given that the corresponding items for Φ are in storage.

First, we assume that for any two switches A and B in a network, the number of links (and hence the line hardware cost) between A and B is proportional to the distance between A and B. It follows that for any four switches A, B, C, and D

$$d(A,B) \leq d(C,D) \text{ implies } C^{TER}_{A,B} \leq C^{TER}_{C,D} \qquad (7)$$

Moreover, the triangular inequality holds for C^{TER}, i.e., for any three switches A, B, and C,

$$C^{TER}_{A,B} + C^{TER}_{B,C} \geq C^{TER}_{A,C} \qquad (8)$$

Let $\Pi(Q)$ be the set of switches that has Q as the nearest satellite switch in Φ. Given two switches A and B, note that $C_{A,B}(\Phi')$ could be different from $C_{A,B}(\Phi)$ only if either A or B is in $\Pi(Q)$. Suppose A is in $\Pi(Q)$. Let S'(A) be the satellite switch in Φ' nearest to A. Note that

$$d(A,S'(A)) \geq d(A,Q)$$

Hence, it follows from Eqs. (7) and (4),

$$C^{SAT}_{A,B}(\Phi') \geq C^{SAT}_{A,B}(\Phi)$$

We conclude that;

1. If $C_{A,B}(\Phi) = C^{TER}_{A,B}$, then

   $$C_{A,B}(\Phi') = C^{TER}_{A,B} = C_{A,B}(\Phi)$$

   (Note that if $A,B \in \Pi(Q)$, then $C_{A,B}(\Phi) = C^{TER}_{A,B}$);

2. If $S'(A) = S'(B)$, then

   $$C_{A,B}(\Phi') = C^{TER}_{A,B}$$

   where $S'(B)$ is the nearest satellite switch to B in Φ';

3. Otherwise,

   $$C^{SAT}_{A,B}(\Phi') = C^{TER}_{A,S'(A)} + C^{TER}_{B,S'(B)} + C_S$$

   and

   $$C_{A,B}(\Phi') = \min\{C^{TER}_{A,B}, C^{SAT}_{A,B}(\Phi')\}$$

From the discussion in the last paragraph, $C(\Phi')$ can be computed from $C(\Phi)$ as follows:

1. Set $C(\Phi') = C(\Phi)$.

2. For each $A \in \Pi(Q)$, calculate $S'(A)$.

3. For each $A \in \Pi(Q)$ and each B, compute $C^{SAT}_{A,B}(\Phi')$.

4. For each $A \in \Pi(Q)$ and each switch $B \notin \Pi(Q)$, if

   $C_{A,B}(\Phi) \neq C^{TER}_{A,B}$, then

   a. Compute $C_{A,B}(\Phi')$

   b. $C(\Phi') = C(\Phi') + C_{A,B}(\Phi') - C_{A,B}(\Phi)$.

5. $C(\Phi') = C(\Phi') - C_{GRD}$.

Next, we consider the situation when Φ' is obtained from Φ by adding a switch Q not in Φ. Let $\Pi(Q)$ be the set of switches that has Q as the nearest satellite switch in Φ'. Similar to before, $C_{A,B}(\Phi')$ would be different from $C_{A,B}(\Phi)$ only if either A or B is in $\Pi(Q)$. For each switch B, let $S(B)$ be the satellite switch nearest to B in Φ. Let A be a switch in $\Pi(Q)$. Then we have

$$d(A,Q) \leq d(A,S(A)),$$

and hence by Eqs. (7) and (4),

$$C^{SAT}_{A,B}(\Phi') \leq C^{SAT}_{A,B}(\Phi),$$

for any switch B. Note that if B is also in $\Pi(Q)$, then by Eq.(8),

$$C^{TER}_{A,B} \leq C^{TER}_{A,Q} + C^{TER}_{B,Q} \leq C^{TER}_{A,S(A)} + C^{TER}_{B,S(B)}$$

Hence,

$$C_{A,B}(\Phi') = C_{A,B}(\Phi) = C^{TER}_{A,B}.$$

We thus conclude that

1. $C_{A,B}(\Phi') \neq C_{A,B}(\Phi)$ only if $B \notin \Pi(Q)$.

2. If $C_{A,B}(\Phi) = C^{SAT}_{A,B}(\Phi)$, then

   $C_{A,B}(\Phi') = C^{SAT}_{A,B}(\Phi')$, and

   $C_{A,B}(\Phi') - C_{A,B}(\Phi) = C^{TER}_{A,Q} - C^{TER}_{A,S(A)}$.

3. Otherwise,

   $C^{SAT}_{A,B}(\Phi') = C^{TER}_{A,Q} + C^{TER}_{C,S(B)} + C_S$,

   and

   $C_{A,B}(\Phi') = \min\{C^{TER}_{A,B}(\Phi'), C^{SAT}_{A,B}(\Phi')\}$.

From the discussion in the last paragraph, $C(\Phi')$ can be computed from $C(\Phi)$ as follows:

1. Set $C(\Phi') = C(\Phi)$.

2. Find the set $\Pi(Q)$ of switches that have Q as the nearest satellite switch in Φ'.

3. For each $A \in \Pi(Q)$ and each B, compute $C^{SAT}_{A,B}(\Phi')$.

4. For each $A \in \Pi(Q)$ and $B \notin \Pi(Q)$,

   a. If $C_{A,B}(\Phi) = C^{SAT}_{A,B}(\Phi)$, then

1. $C_{A,B}(\Phi') = C^{SAT}_{A,B}(\Phi')$,

2. $C(\Phi') = C(\Phi') + C^{TER}_{A,Q} - C^{TER}_{A,S(A)}$

b. Otherwise,

1. $C_{A,B}(\Phi') = \min\{C^{TER}_{A,B}(\Phi'), C^{SAT}_{A,B}(\Phi')\}$,

2. $C(\Phi') = C(\Phi') + C_{A,B}(\Phi') - C_{A,B}(\Phi)$.

5. $C(\Phi') = C(\Phi') + C_{GRD}$.

From above discussions, it follows that substantial savings in cost computation can be obtained if each successive set of satellite switches is obtained from the previous set by either adding a switch or deleting a switch. Suppose there are N switches in the backbone network. Then each subset S of switches corresponds to a unique N-bit binary word $g_N g_{N-1}, \ldots, g_1$, where g_i is 1 if and only if switch i is in S. Hence, what we desire is an ordering of the N-bit words such that any two successive words differ in one and only one bit. A binary code with such a property is called the Gray code or the cyclic code [BOOTHROYD, 1964], [GSCHWIND, 1975]. A discussion on the generation procedure of one type of such code words, the reflected binary cyclic code, is presented in Appendix A.

4.3 Complexity of the Cyclic Subset Ordering Algorithm

Below, we estimate the time complexity of the procedure presented in Section 4.2. Suppose there are k satellite switches. Then, on the average, each satellite switch is the nearest satellite switch for N/k switches, where N is the total number of switches. Hence, the total time required over all the subsets of N switches, t_{Total}, is

$$t_{Total} = \text{(Time required to generate all subsets)}$$

$$+ \sum_{k=0}^{N} [\text{(number of k-element subsets)}$$

$$\times \text{(average time required for each k subset)}]$$

$$= \text{(time required to generate all subsets)}$$

$$+ (N^2 + \sum_{k=1}^{N} \binom{N}{k} \times \frac{N}{k} \times N).$$

From Appendix A, the generation of all subsets such that each two successive subsets differ in one and only one element can be done in time $O(2^{N+1})$. Hence,

$$t_{Total} = O(2^{N+1}) + N^2 + \sum_{k=1}^{N} \binom{N}{k} \times \frac{N}{k} \times N.$$

$$= O(2^{N+1}) + N^2 + N \sum_{k=1}^{N} \binom{N}{k} \times \frac{N}{k} .$$

We will show that

$$\sum_{k=1}^{N} \binom{N}{k} \times \frac{N}{k} \leq 3 \times 2^N,$$

and consequently,

$$t_{Total} = O(2^{N+1}) + N^2 + 3N \times 2^N$$

$$= O(N \times 2^N)$$

This is an order N improvement over the straight-forward approach presented in Section 4.1.

ASSERTION

$$\sum_{k=1}^{N} \binom{N}{k} \times \frac{N}{k} \leq 3 \times 2^N.$$

PROOF

$$\sum_{k=1}^{N} \binom{N}{k} \times \frac{N}{k} = \sum_{k=1}^{N} \binom{N+1}{k+1} \times \frac{k+1}{N+1} \times \frac{N}{k}$$

$$= \sum_{k=1}^{N} \binom{N+1}{k+1} \times \frac{N}{N+1} \times \frac{k+1}{k}$$

$$= \sum_{k=1}^{N} \binom{N+1}{k+1} \times \frac{N}{N+1} \times (1+\frac{1}{k})$$

$$= \sum_{k=1}^{[N/2]} \binom{N+1}{k+1} \times \frac{N}{N+1} \times [(1+\frac{1}{k}) + (1+\frac{1}{N-k})]$$

$$= \sum_{k=1}^{[N/2]} \binom{N+1}{k+1} \times \frac{N}{N+1} \times (2+\frac{1}{k}+\frac{1}{N-k}).$$

Now, notice that for $1 \leq k \leq [N/2]$, the maximum value for $\frac{1}{k} + \frac{1}{N-k}$ occurs at $k=1$. Thus

$$\frac{N}{N+1} \times (2+\frac{1}{k}+\frac{1}{N-k}) \leq \frac{N}{N+1} \times (2+1+\frac{1}{N-1})$$

$$= \frac{N}{N+1} \times \frac{3N-2}{N-1}$$

$$< 3.$$

But, then we obtain

$$\sum_{k=1}^{N} \binom{N}{k} \times \frac{N}{k} = \sum_{k=1}^{[N/2]} \binom{N+1}{k+1} \times \frac{N}{N+1} \times \left(2 + \frac{1}{k} + \frac{1}{N-k}\right)$$

$$< \sum_{k=1}^{[N/2]} \binom{N+1}{k+1} \times 3$$

$$< 3 \times \frac{1}{2} \times 2^{N+1}$$

$$= 3 \times 2^{N}.$$

Q.E.D.

4.4 Other Problem Reduction Techniques

The procedure described in Section 4.2, although is an improvement over the straightforward procedure, still takes time $N \times 2^N$. Hence, for switch sets of moderate to large size (say 20 to 50 switches), other problem reduction techniques are needed to obtain a solution. In the following, we describe several such possibilities.

First, with the given earth station cost, terrestrial transmission cost, traffic matrix, etc., we can obtain various upper and lower bounds on the number of satellite switches that can be deployed in a region. For example, the following result is derived in Appendix B.

ASSERTION

Suppose A is selected as a satellite switch. Then, for any switch B with

$$t_B \times C^{TER}_{A,B} \leq C_{GRD},$$

it is not cost-effective to implement a satellite switch at B, (where t_B is the total low priority traffic requirement at B).

These types of bounds are especially useful when the number of switch sites is large.

Second, during experiments, we found that a number of the locations consistently get picked as satellite switch sites, independent of the total number of backbone switches, and within a wide range of cost and traffic variations (e.g., TINKER, McCELLAN, ALBANY). These are usually nodes with high traffic requirements. Hence, during the satellite switch selection process, we can include these nodes in the satellite switch list, and exclude some of their neighboring nodes from consideration.

Moreover, from the experimental results reported in Section 5.6.1, we observe that for a given Host and terminal data base, the number of satellite switches selected is almost independent of the size of the backbone switch set. It follows that for a design involving a large number of switches, we can first estimate the number of satellite switches by applying the procedure to a problem with much fewer switches, and then limit the search to only switch sets with size close to the estimated value.

5. EXPERIMENTAL RESULTS

5.1 System Model and Base Parameter Values

We have conducted extensive experiments with both the satellite switch selection algorithm developed in Section 4, and the MODULARIZED ADD algorithm, on the AUTODIN II system with the AUTOVON sites as candidate backbone switch locations. The AUTODIN II and AUTOVON system size is as follows:

AUTODIN II SYSTEM

Total Number of User Locations = 300
    Number of Host Computers = 86
    Number of Concentrators = 26
    Number of TDMX's = 85
    Number of TCU's = 11
    Number of Isolated Terminals = 101
Total Traffic = 1.26 Megabits/Sec.

AUTOVON SYSTEM

Total Number of AUTOVON Sites = 60

Figures 2 and 3 are the AUTODIN II and AUTOVON locations, respectively.

The assumptions used in the mixed media backbone network design are as follows:

Proportion of Low Delay Traffic = 10%
Average End-to-End Delay For Low Delay Traffic $\leq$ 100 msec.
Protocol Overhead = 45%

FIGURE 2: USER SITES IN THE AUTODIN II SYSTEMS

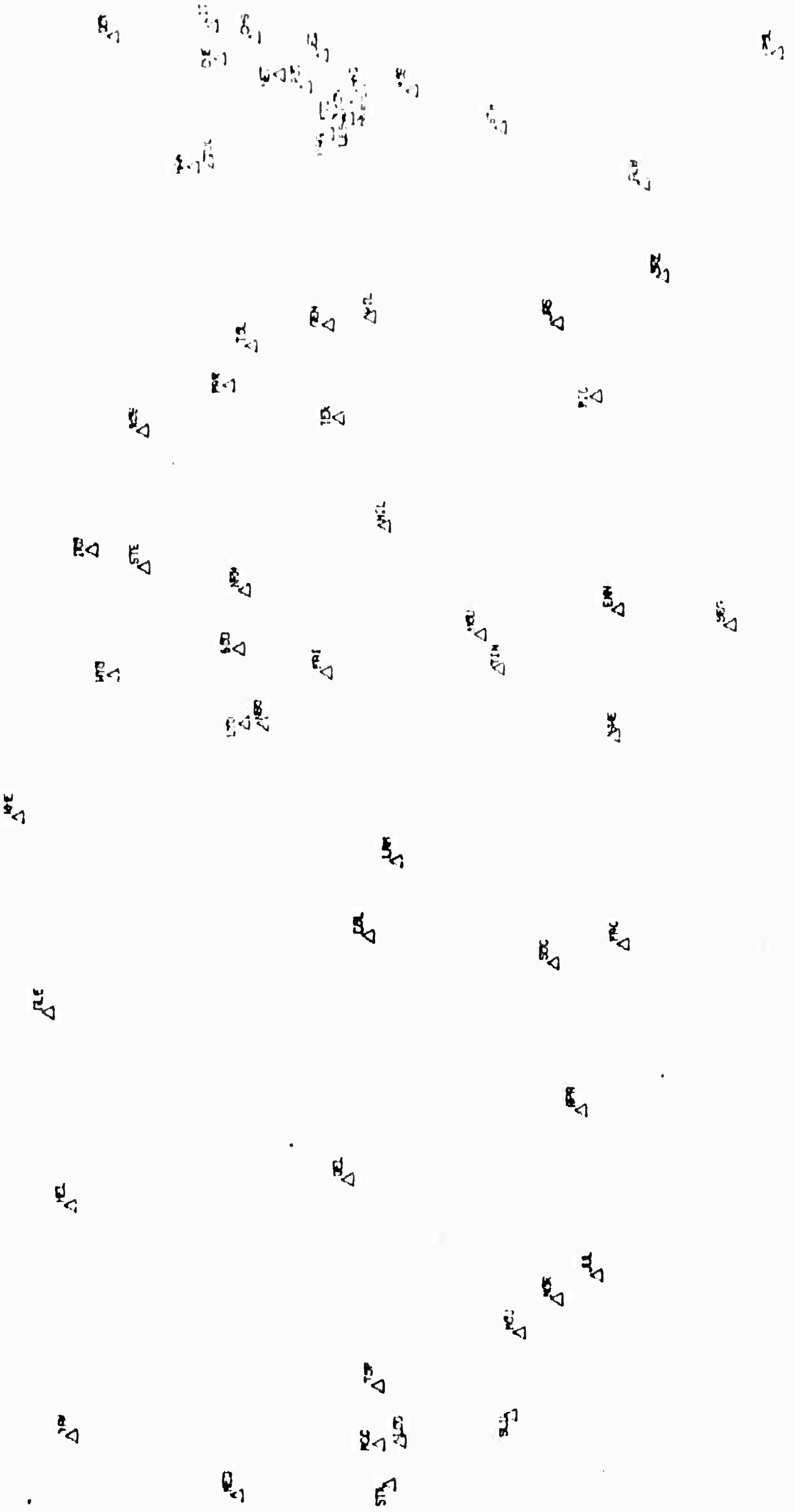


FIGURE 3: AUTOVON LOCATIONS

Routing Overhead = 7%
Average Packet Length = 500 bits
Routing Method = Optimal Multipath
Satellite Channel Access Technique = Dedicated Up-Link/ Broadcast Down-Link

Tables 1 and 2 list cost and characteristics of local access and backbone facilities used in the design.

5.2 Sensitivity Studies on the Backbone Location of Terrestrial Networks

Here, we present some experimental results on the backbone switch location problem for terrestrial networks. The main purpose of this section is to provide, in some sense, experimental justifications for the switch location approach given in Section 3.2.

We compare the following two different modifications of the ADD algorithms on terrestrial network backbone designs:

1. The procedure presented in Section 2.2 (and [NAC, 1976]) which uses detailed backbone line cost tradeoff evaluations. We call this the DB (Detailed Backbone Estimation) approach.

2. The procedure presented in Section 3.2 which uses only rough incremental backbone line cost estimate, $\Delta BC(K)$, dependent on the number of backbone switches used. We call this the RB (Rough Backbone Estimation) approach.

For simplicity, in this study we make $\Delta BC(K)$ the same for all values of K. Thus, we use only ΔBC to denote $\Delta BC(K)$. From experience with the AUTODIN II system design, the backbone line cost varies approximately linearly with the number of backbone switches within the range of 5-20 switches, [NAC, 1976], thus the above approximation is justified.

FACILITY TYPE		COSTS ($/MO).
LINE MILEAGE COST	2.4 KB/S Line	2.0/mile
	4.8 KB/S Line	2.0/mile
	9.6 KB/S Line	2.0/mile
	50. KB/S Line	12.0/mile
LINE FIXED COST (both ends)	2.4 KB/S Line	100.
	4.8 KB/S Line	240.
	9.6 KB/S Line	570.
	50. KB/S Line	850.
FIXED HARDWARE COST	Isolated Terminal	0
	TDMX	67.
	TCU	1170.
	Concentrator	2724.
PER TERMINAL CONNECTION COST	TDMX	13.
	TCU	0
	Concentrator	0

TABLE 1: LOCAL ACCESS FACILITY UNIT COST

		FACILITY TYPE	COST ($/MO.)
		Software Development One-Time Fixed Cost	60,000
		Satellite Earth Station Cost	10,000
		Satellite Channel (50 KB/S-Simplex) Space Segment Cost	500
		Satellite Channel (50 KB/S-Simplex) Access Cost	300
		Terrestrial Channel (50 KB/S) Mileage Cost	5/mile
		Terrestrial Channel (50 KB/S) Termination Cost	425/end
		Switch Node (Pluribus) Fixed Cost	10,600
SWITCH NODE COST	INTERFACE COST	Trunk Line	87
		Host or Concentrator Line	87
		TCU Line	87
		MUX Line	12
		2.4 KB/S Terminal Line	12
		4.8 KB/S Terminal Line	62
		9.6 KB/S Terminal Line	75
		50. KB/S Terminal Line	87

TABLE 2: BACKBONE FACILITY UNIT COST

Table 3 lists the cost estimates of the designs generated by the two procedures. (Although each set of switch locations is generated by using a particular ΔBC value, in order to have an adequate comparison, the backbone costs listed in Table 3 are estimated by the same formula, Eq. (2), with unit costs given by Table 2.

Comparing the design costs in Table 3; we can conclude that

1. The cost of the best (lowest cost) designs for different values of ΔBC's differ very little from one another. This implies that the exact determination of ΔBC is not important. A rough estimate for ΔBC can usually give us a reasonably good selection of backbone switches.

2. The best designs obtained with the RB approach are slightly worse than the best design obtained with the DB approach (the difference is less than 1.1%). This suggests that a coarser approach such as RB suffices for most applications. The RB approach is inherently much faster than the DB approach.

In Table 4, we compare the designs listed in Table 3 that have the same number of backbone switches but with different total cost. We can observe the following:

1. For the designs giving the same number of backbone nodes, the design with lower local access cost almost always also has lower total cost.

2. For the designs giving the same number of backbone nodes, the design with lower local access cost usually has higher backbone line cost.

DESIGN APPROACH	RB			DB
Number of Backbone Nodes \ ABC (K$/mo)	0.0	6.6	12.0	--
7	757	757	757	757
8	-	-	*745	753
9	*749	751	771	750
10	-	-	-	-
11	-	*750	750	*742
12	757	757	761	756
13	-	-	-	-
14	784	-	-	791
15	-	-	830	-
16	-	844	-	797
17	-	-	-	-
18	858	-	-	828

TABLE 3: ESTIMATED TOTAL COST OF DESIGNS GENERATED

BACKBONE NETWORK COST ESTIMATED BY EQUATION (2)

UNIT = K$/mo

---

*Lowest cost designs

NUMBER OF BACKBONE NODES	SOLUTION PROCEDURE	ABC (K$/MO)	LOCAL ACCESS COST (K$/MO,	BACKBONE LINE COST (K$/MO)	TOTAL COST (K$/MO)
8	DB	-	429	164	753
	RB	12.0	416	169	745
9	DB	-	406	173	750
	RB	0.0	383	184	749
	RB	6.6	406	173	751
	RB	12.0	417	182	771
11	DB	-	360	190	742
	RB	6.6	365	192	750
	RB	12.0	365	192	750
12	DB	-	357	196	756
	RB	0.0	356	198	757
	RB	6.6	356	198	757
	RB	12.0	360	197	761
14	DB	-	362	205	791
	RB	0.0	348	211	784
15	DB	-	329	223	797
	RB	12.0	379	218	830
18	DB	-	326	235	828
	RB	0.0	354	237	858

TABLE 4: COMPARISON OF DESIGNS WITH SAME NUMBER OF SWITCHES
(BASED ON DESIGNS LISTED IN TABLE 3)

BACKBONE NETWORK COST ESTIMATED BY EQUATION (2)

From the above observations, we conclude that for a given number of backbone switches, a good heuristic criterion in selecting low cost designs is to minimize the local access cost. This agrees with the intuition that, because the local access cost is the largest component of total cost, it has the greatest influence on the total design cost.

5.3 Accuracy of Backbone Network Cost Estimates

The ADD algorithm, given in Section 3.2, for selecting the backbone switches (for mixed networks) estimates the backbone incremental cost of one additional switch, given that there are already K switches selected, by a single value: $\Delta BC(K)$. The algorithm given in Section 4 for selecting the satellite switches estimates the backbone network cost by Eq. (6):

$$C(\Phi) = |\Phi| \times C_{GRD} + t_{A,B} \times C_{A,B}(\Phi),$$

where

Φ = the set of satellite switches,

C_{GRD} = ground station cost,

$t_{A,B}$ = low priority requirement from A to B,

and

$C_{A,B}(\Phi)$ is given by Eqs. (3) - (5).

Here we examine the accuracy and the consistency of these backbone network cost estimates.

We have manually designed several mixed-media networks with varying numbers of backbone nodes. Based on these designs, we estimate the backbone network incremental cost (excluding the switch cost) to be

$$\Delta BC(N) \approx \$4{,}500/\text{month}, \tag{9}$$

and the backbone network cost (excluding the switch cost) to be

$$BC(N) \approx \$88{,}100/\text{month} + \Delta BC(N) \times N, \tag{10}$$

for $5 \leq N \leq 20$. Several sets of backbone switch locations were then selected using the ADD algorithm based on these values over a range of alternatives. For each set of the backbone switches selected, the optimum satellite switch locations were then selected using the satellite switch selection algorithm based on Eq. (6). Actual topological network designs were then generated using these satellite and terrestrial switch locations. In Figure 4, we compare the backbone network cost of the actual mixed network designs to the estimates obtained from Eq. (10). As can be seen, the estimates and the actual costs are uniformly close (always within 10%). The fact that the optimum number of satellite switches is fairly independent of the switch set size (see Section 5.6.1) may contribute partially to the close fit of the estimates. It is somewhat surprising that the backbone network cost (as a function of the number of switches) can be approximated so closely by a straight line.

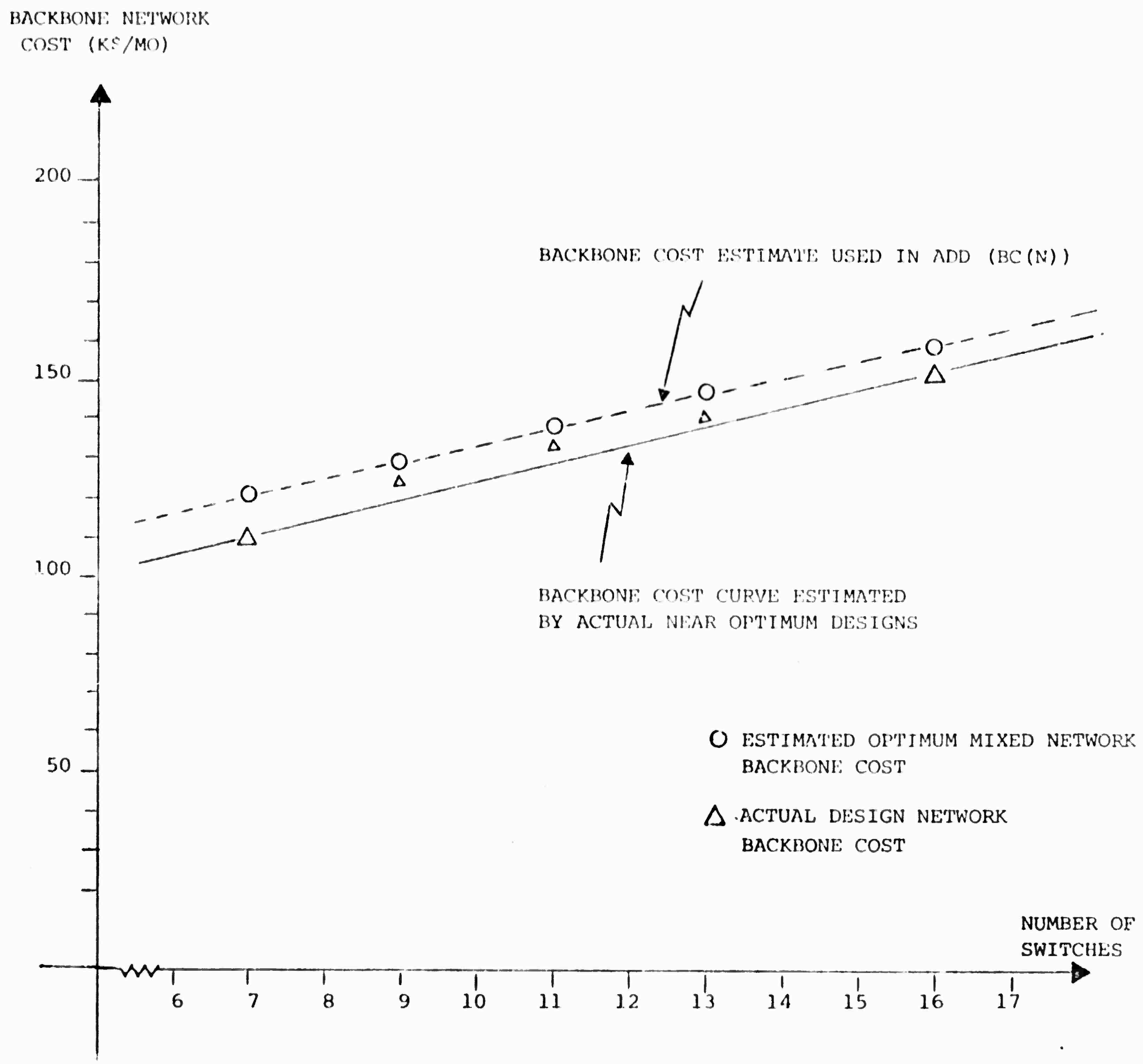


FIGURE 4: COMPARISON OF ACTUAL BACKBONE NETWORK COSTS (EXCLUDING SWITCH COST) VS. ESTIMATES BY EQUATION (10) USED IN ADD ALGORITHM

In order to assess the accuracy of Eq. (6) and the satellite switch selection algorithm, we consider an 11-switch set selected by the ADD algorithm. For a given k, $0 \leq k \leq 11$, we use the algorithm to select the k optimum satellite switches. A mixed network topological design is then generated based on the 11 switches and these selected satellite switches. In Figure 5 we compare the backbone network cost of the actual design with a given set of satellite switches to the estimates by Eq. (6) on the same set of satellite switches. It can be seen that there is a gap between the actual costs and the estimates (about 10-15% difference). However, the difference is quite consistent, and the general pattern of the two cost curves are similar. We thus conclude that Eq. (6) is adequate for the purpose of satellite switch site selection.

5.4 Total Network Cost Versus Number of Backbone Nodes

Figure 6 plots the total network cost estimates (in which backbone costs are estimated by Eq. (10)) as a function of the number of backbone switches. Observe that, similar to terrestrial network designs, these points represent very closely a convex (U-shaped) curve, and this curve is quite flat near the optimum point. Moreover, the mixed networks usually offer considerable savings over the terrestrial networks.

For illustrative purpose, the following designs are plotted in Figures 7a-7f:

a. A terrestrial network design for 8 prespecified backbone switches, with a total system cost of $764K/mo.

b. A mixed network design for the same 8 prespecified backbone switches, with a total system cost of $722K/mo.

c. A terrestrial network design for 11 ADD algorithm selected switch locations with a total system cost of $734K/mo.

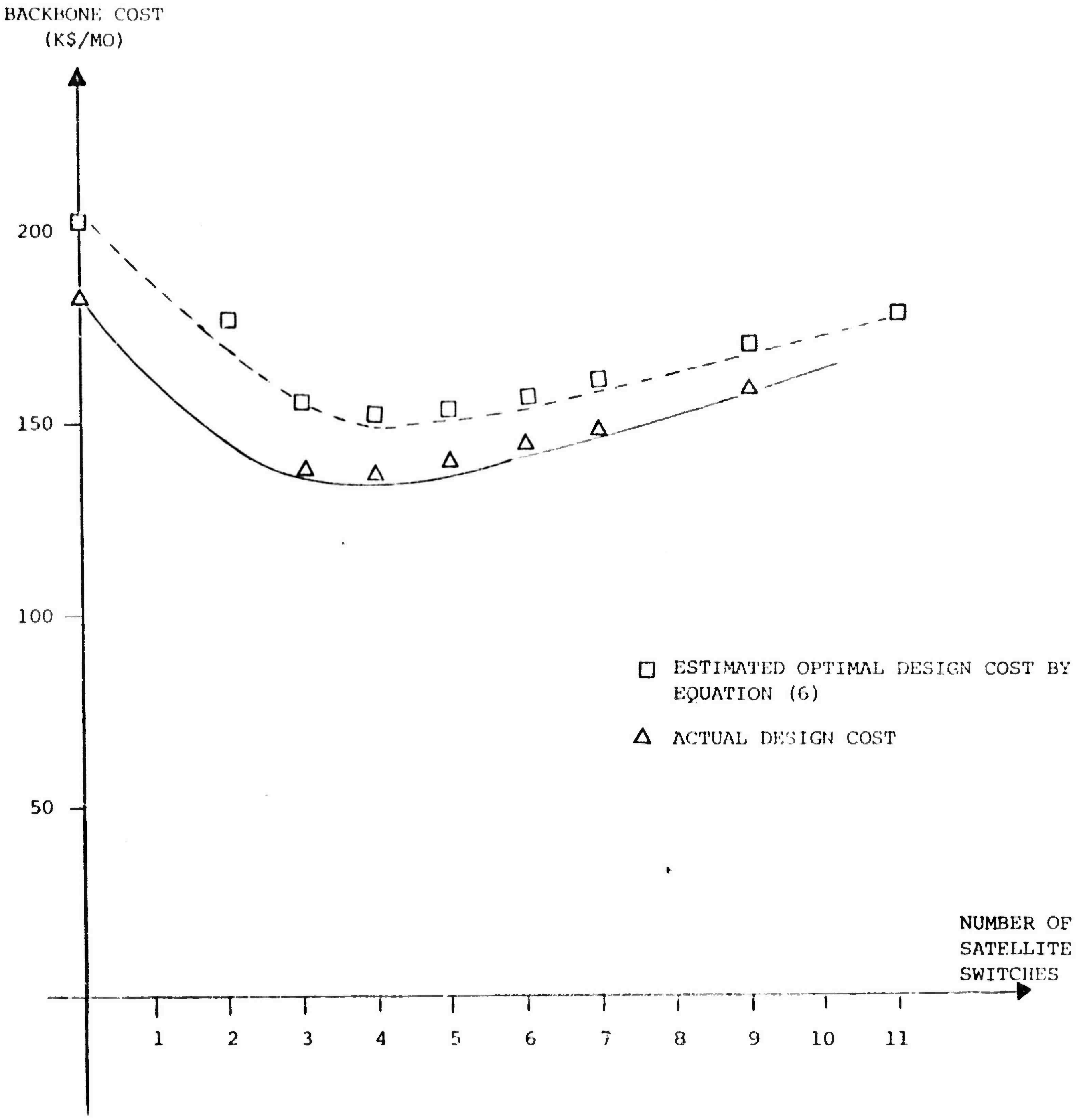


FIGURE 5: COMPARISON OF BACKBONE NETWORK COSTS (EXCLUDING SWITCH COST) VS. ESTIMATES BY EQUATION (6) USED IN SATELLITE SWITCH SELECTION ALGORITHM

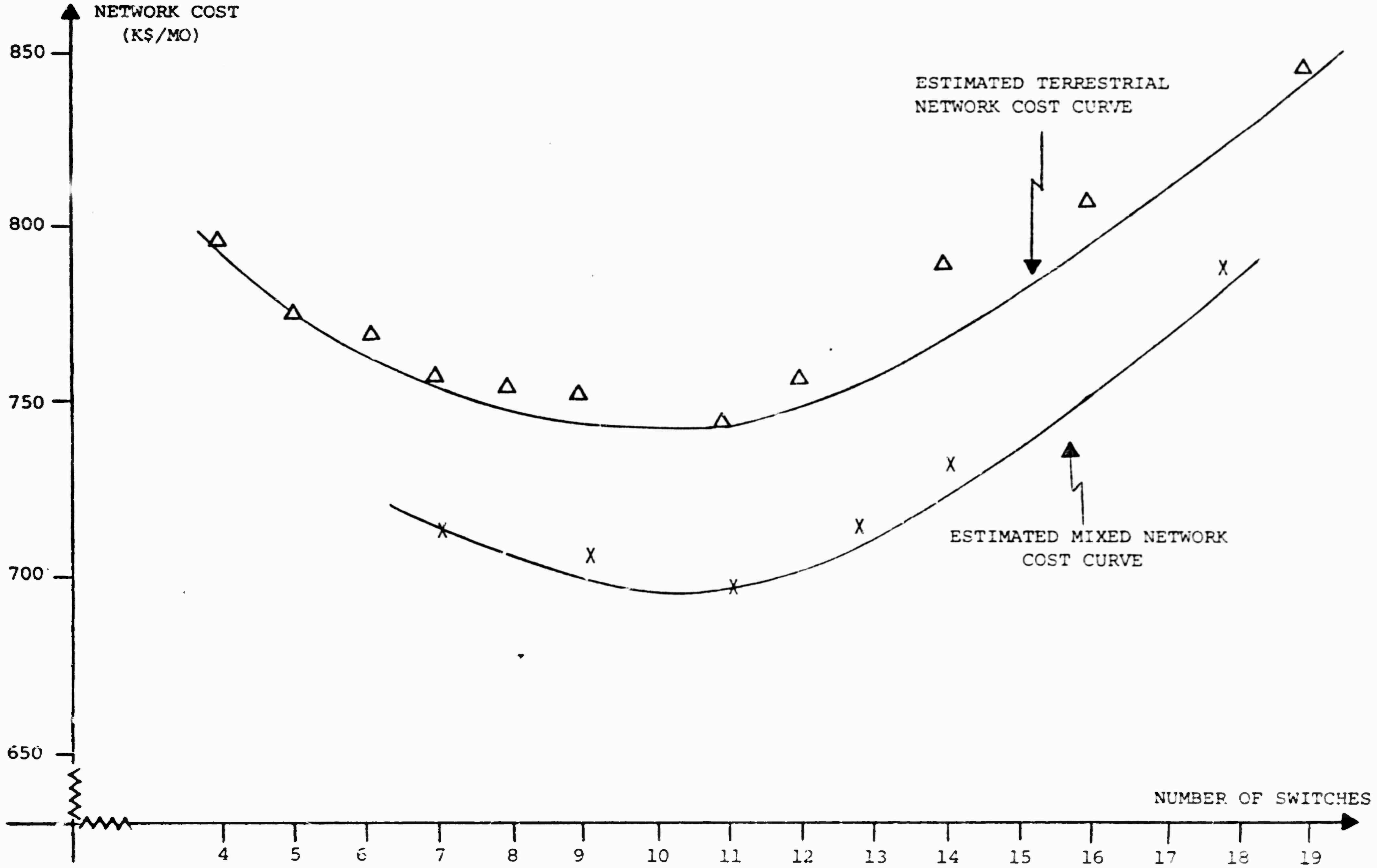
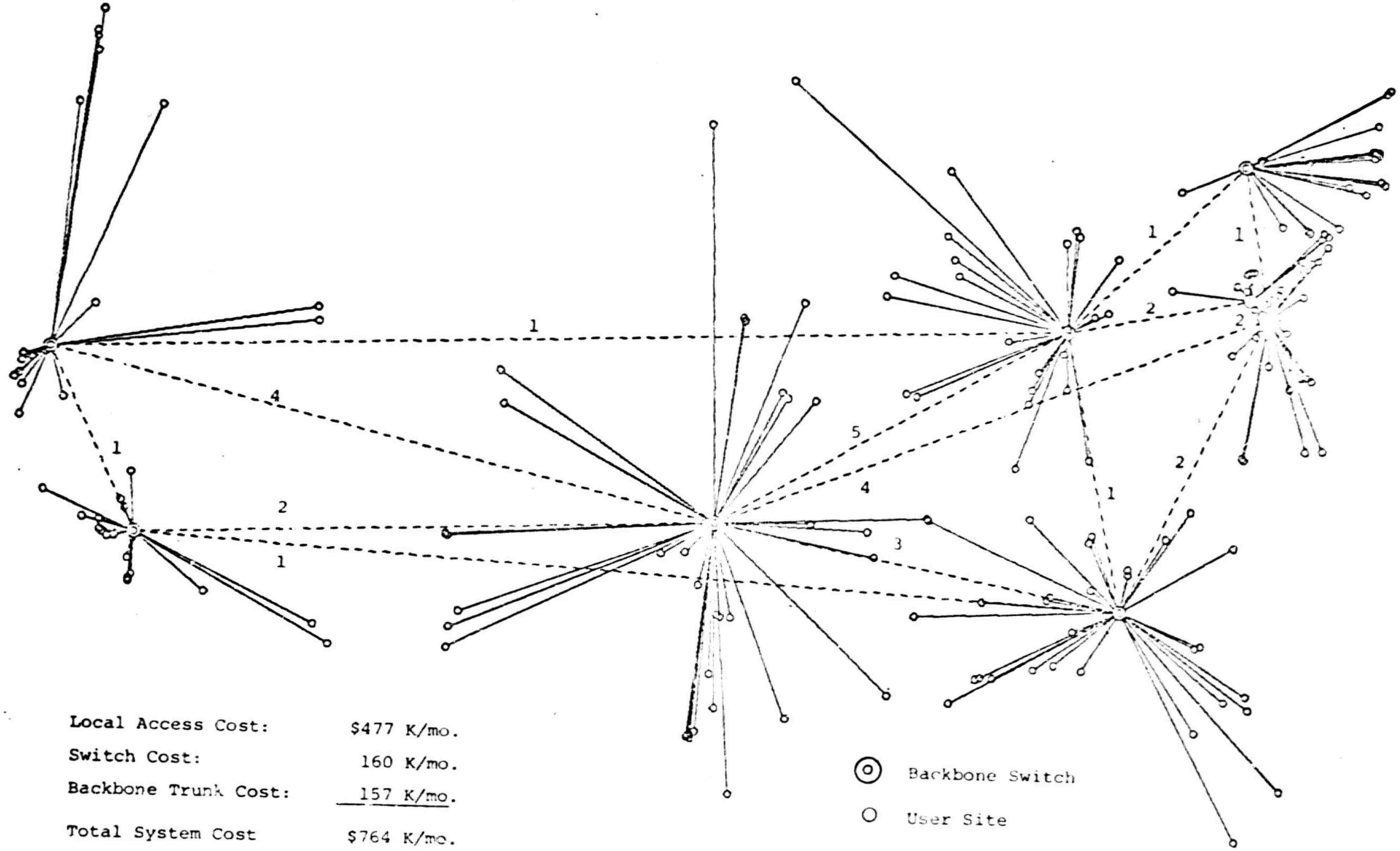


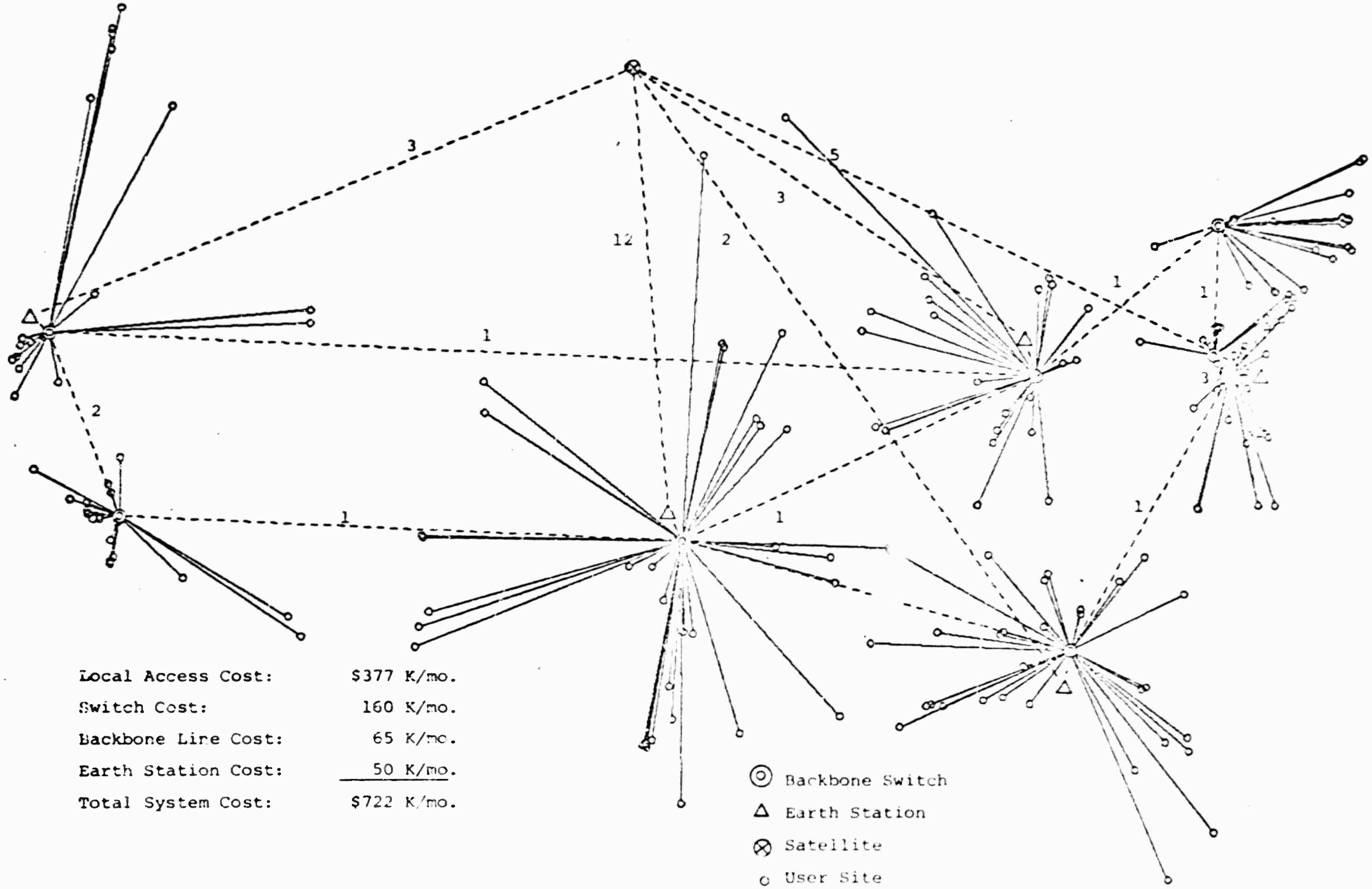
FIGURE 6: NETWORK COST AS A FUNCTION OF NUMBER OF BACKBONE SWITCHES



Local Access Cost:	$477 K/mo.
Switch Cost:	160 K/mo.
Backbone Trunk Cost:	157 K/mo.
Total System Cost	$764 K/mo.

FIGURE 7(a): NETWORK WITH ALL TERRESTRIAL BACKBONE LINKS: 8 PRE-GIVEN SWITCHES

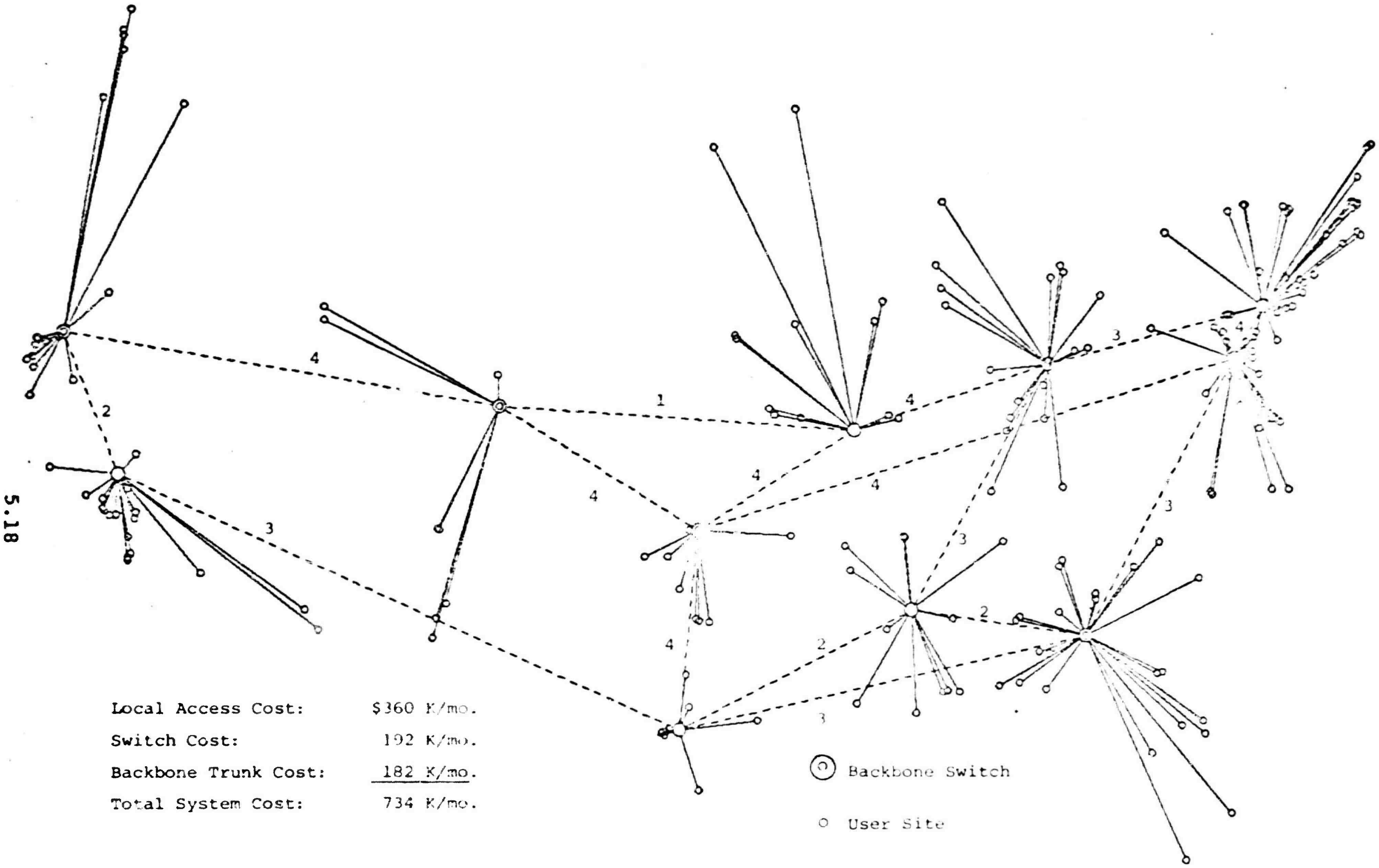
NOTE: THE NUMERALS ALONG THE LINKS INDICATE THE NUMBER OF 50KB/s CHANNELS USED



Local Access Cost:	$377 K/mo.
Switch Cost:	160 K/mo.
Backbone Lire Cost:	65 K/mo.
Earth Station Cost:	50 K/mo.
Total System Cost:	$722 K/mo.

FIGURE 7(b): NETWORK WITH MIXED MEDIA BACKBONE LINKS: 8 PRE-GIVEN SWITCHES

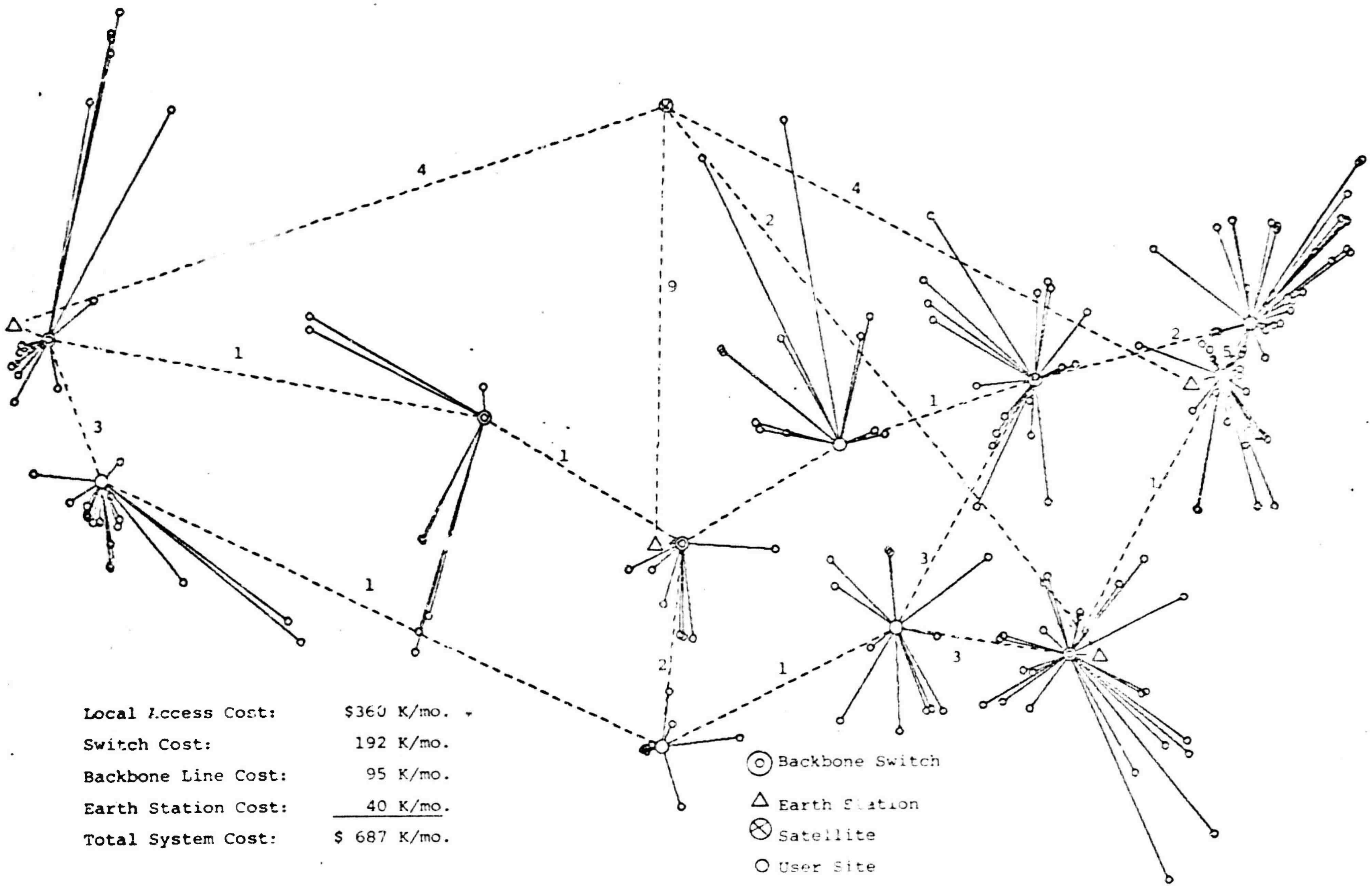
NOTE: THE NUMERALS ALONG THE LINKS INDICATE THE NUMBER OF 50KB/s CHANNELS USED



Local Access Cost:	$360 K/mo.
Switch Cost:	192 K/mo.
Backbone Trunk Cost:	182 K/mo.
Total System Cost:	734 K/mo.

FIGURE 7(c): NETWORK WITH ALL TERRESTRIAL BACKBONE LINKS: 11 ADD-GENERTED SWITCHES

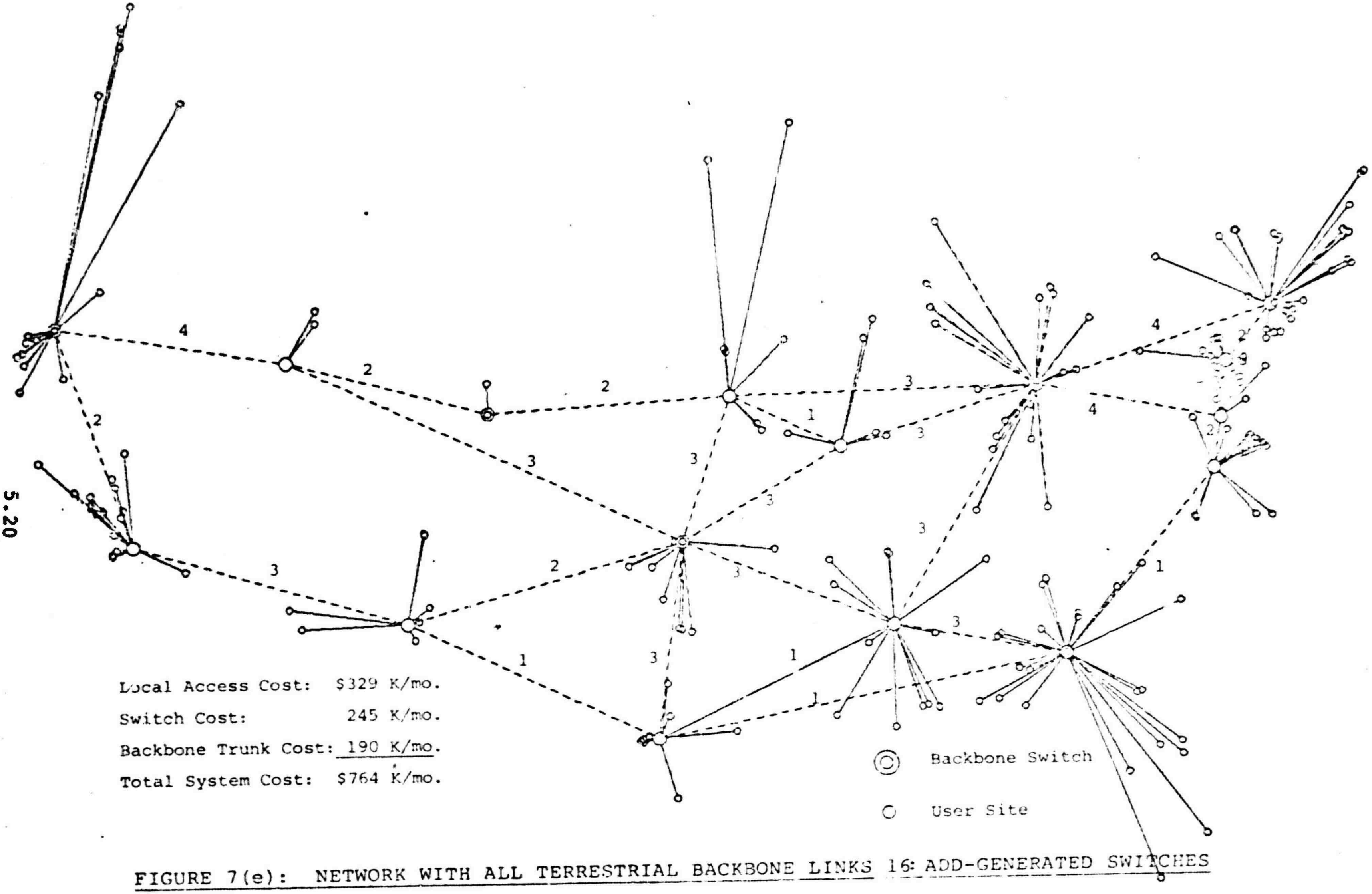
NOTE: THE NUMERALS ALONG THE LINKS INDICATE THE NUMBER OF 50KB/s CHANNELS USED



Local Access Cost:	$360 K/mo.
Switch Cost:	192 K/mo.
Backbone Line Cost:	95 K/mo.
Earth Station Cost:	40 K/mo.
Total System Cost:	$ 687 K/mo.

FIGURE 7(d): NETWORK WITH MIXED MEDIA BACKBONE LINKS: 11-ADD-GENERATED SWITCHES

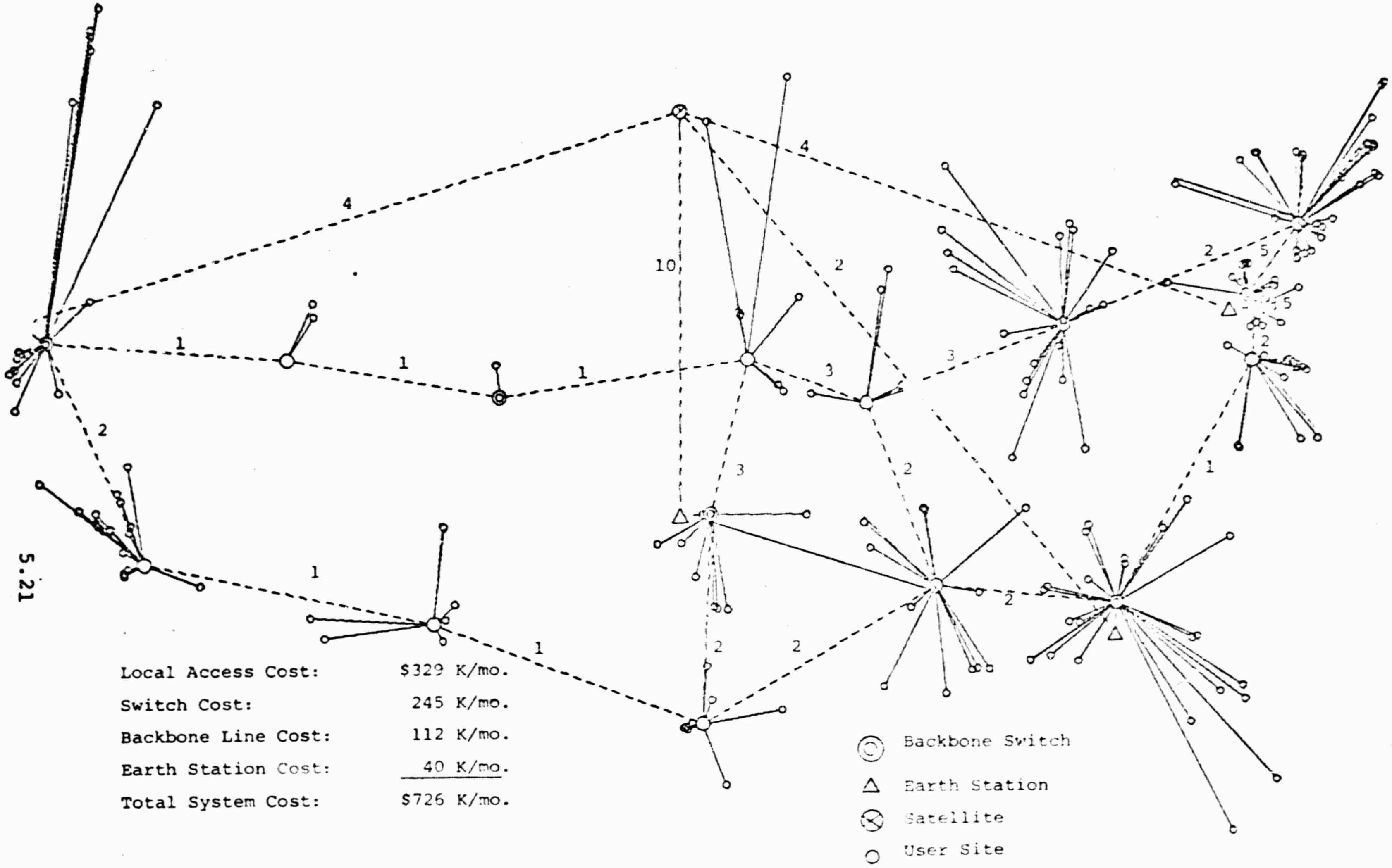
NOTE: THE NUMERALS ALONG THE LINKS INDICATE THE NUMBER OF 50KB/s CHANNELS USED



Local Access Cost: $329 K/mo.
Switch Cost: 245 K/mo.
Backbone Trunk Cost: 190 K/mo.
Total System Cost: $764 K/mo.

FIGURE 7(e): NETWORK WITH ALL TERRESTRIAL BACKBONE LINKS 16 ADD-GENERATED SWITCHES

NOTE: THE NUMERALS ALONG THE LINKS INDICATE THE NUMBER OF 50KB/s CHANNELS USED



Local Access Cost: $329 K/mo.
Switch Cost: 245 K/mo.
Backbone Line Cost: 112 K/mo.
Earth Station Cost: 40 K/mo.
Total System Cost: $726 K/mo.

FIGURE 7(f): NETWORK WITH MIXED MEDIA BACKBONE LINKS 16: ADD-GENERATED SWITCHES

NOTE: THE NUMERALS ALONG THE LINKS INDICATE THE NUMBER OF 50KB/s CHANNELS USED

d. A mixed network design for the same 11 ADD algorithm selected switch locations with a total system cost of $687K/mo.

e. A terrestrial network design for 16 ADD algorithm selected switch locations with a total system cost of $764K/mo.

f. A mixed network design for the same 16 ADD algorithm selected switch locations with a total system cost of $726K/mo.

Observe that the mixed network topologies is usually much simpler than the corresponding terrestrial network topologies. The terrestrial portion of a mixed network usually consists of a low capacity loop through all the switches plus a few more cross links.

5.5 Backbone Network Cost Versus Number of Satellite Switches

Figure 8 plots the estimated backbone network cost (cost estimate based on Eq.(7)) of the best designs with various given number of satellite switches, with different backbone switch sets and throughput levels. As can be seen, the total cost curve is generally very flat. This fact is of great help in making near-optimal designs.

In Figure 9, the five backbone component costs (earth station cost, satellite space segment cost, satellite access cost, terrestrial mileage cost and terrestrial hardware cost) are plotted as functions of the number of satellite switches for the designs generated with the 11 backbone switch set. It can be seen that as more satellite switches are used, the cost of the terrestrial components decreases, while the cost of the satellite components increases. This is because aside from the earth station cost, satellite transmission is usually much cheaper than terrestrial transmission. Hence, as more satellite switches become available, more and more terrestrial channels are replaced by satellite channels.

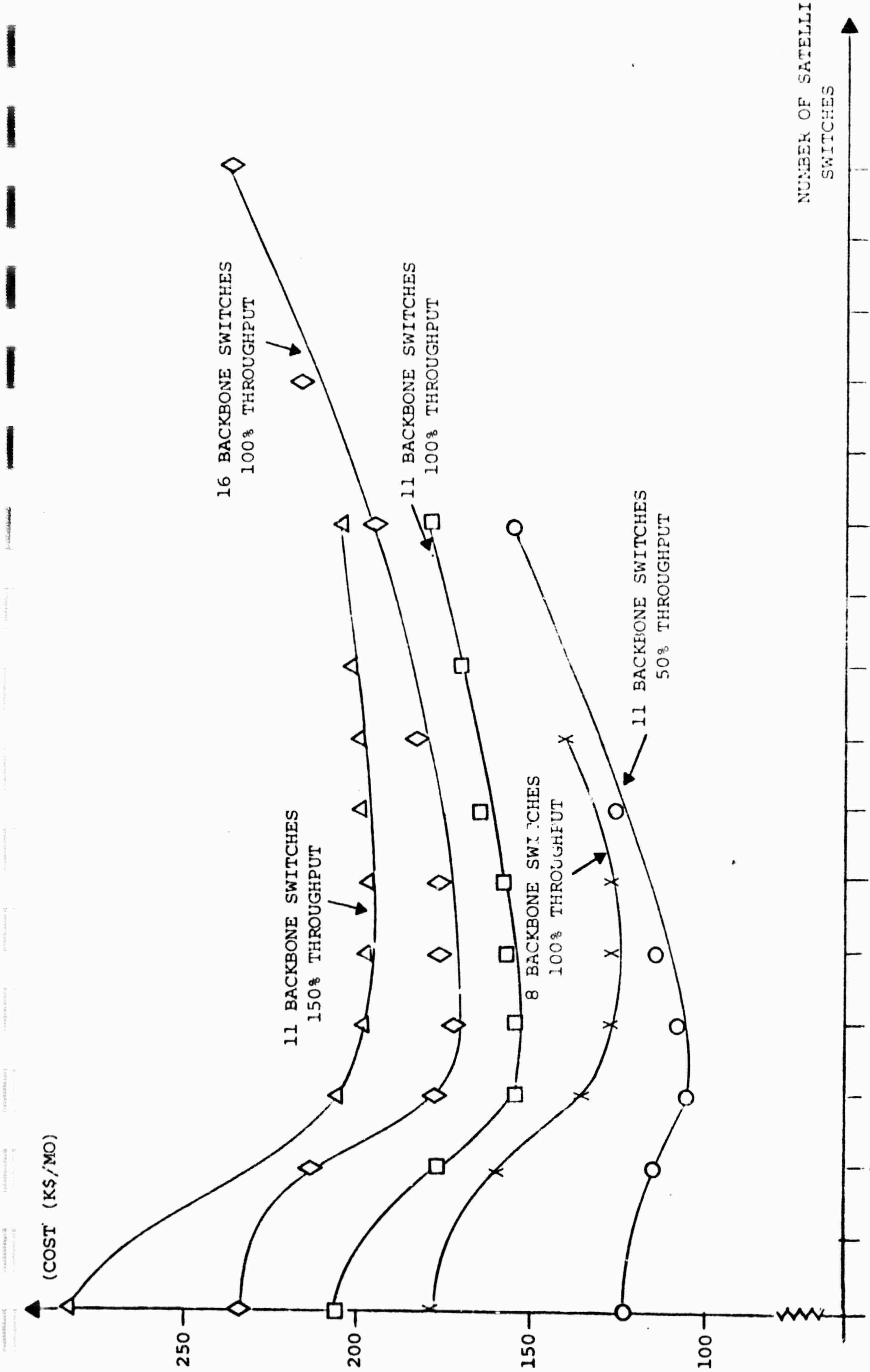


FIGURE 8: ESTIMATED BACKBONE NETWORK COST VS. NUMBER OF SATELLITE SWITCHES COST ESTIMATED BY EQ. (6)

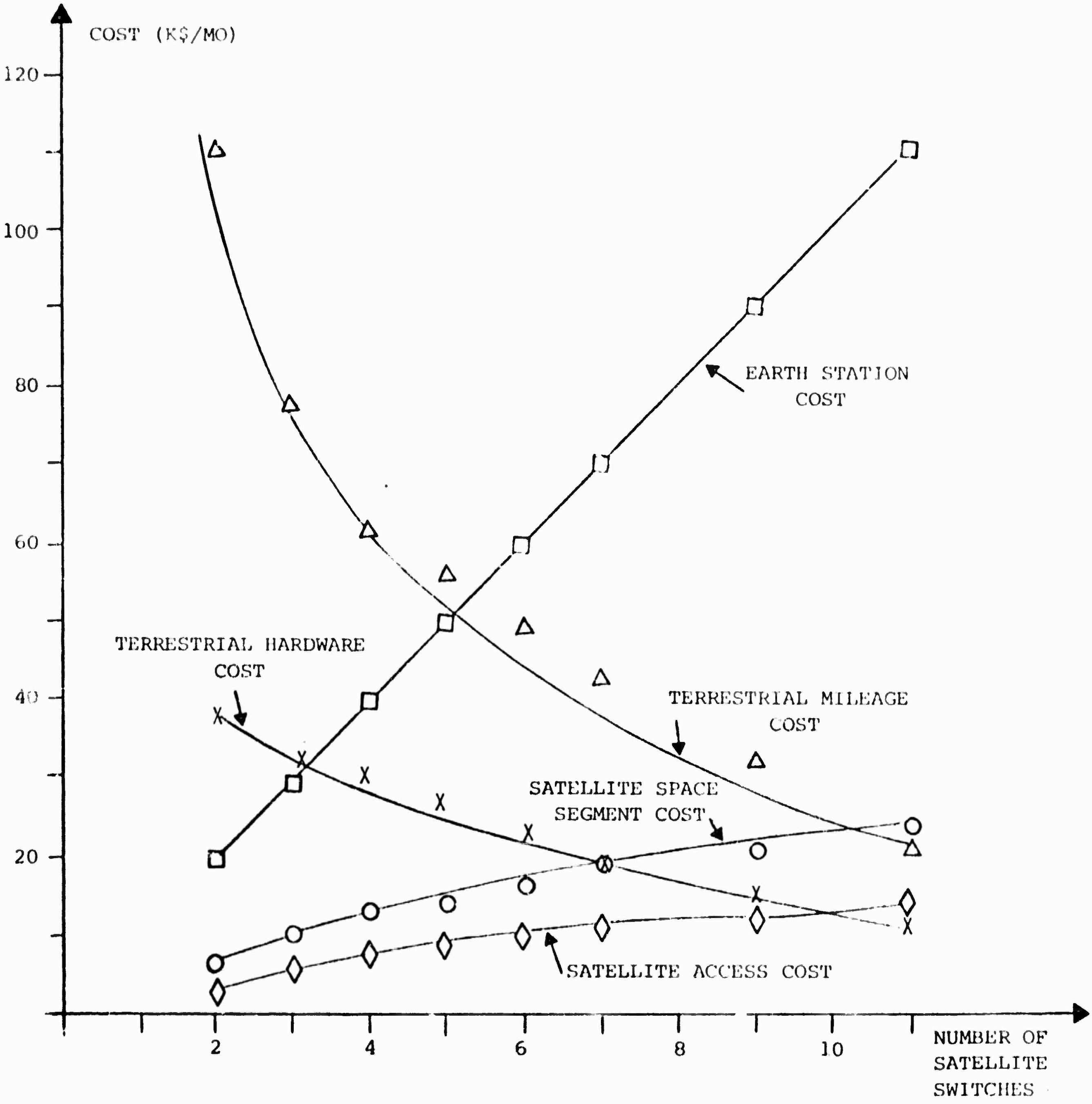


FIGURE 9: BACKBONE COMPONENT COST VS. NUMBER OF SATELLITE SWITCHES, 11 BACKBONE SWITCH SET, 100% THROUGHPUT

NOTE: Network Cost Estimated by Eq. (6)

Moreover, as is obvious from Figure 9, the earth station cost and the terrestrial mileage cost are the two main cost components, and compared with the other cost components, they are much more sensitive to the variation in the number of satellite switches. Thus, the main tradeoff in determining the number of satellite switches is between the earth station cost and the terrestrial mileage cost.

5.6 Sensitivity Study on the Optimum Number of Satellite Switches

5.6.1 Effect of Backbone Switch Set Size

On the switch sets we have investigated (all except the 8-node switch sets are obtained by the ADD algorithm), the optimal number of satellite switches are as follows:

Backbone Switch Set Size	Satellite Switch Set Size
7	4
8	5
9	4
11	4
13	4
16	4

It thus appears that the number of satellite switches required for a good design is independent of the total number of switches, given that the switch sites are chosen optimally, e.g., by the ADD algorithm. This, together with the fact that the total backbone network cost curve is very flat (with respect to the number of satellite switches), suggests that in selecting the satellite switches, one needs only consider switch sets with size within a certain limited range (see also Section 4.3).

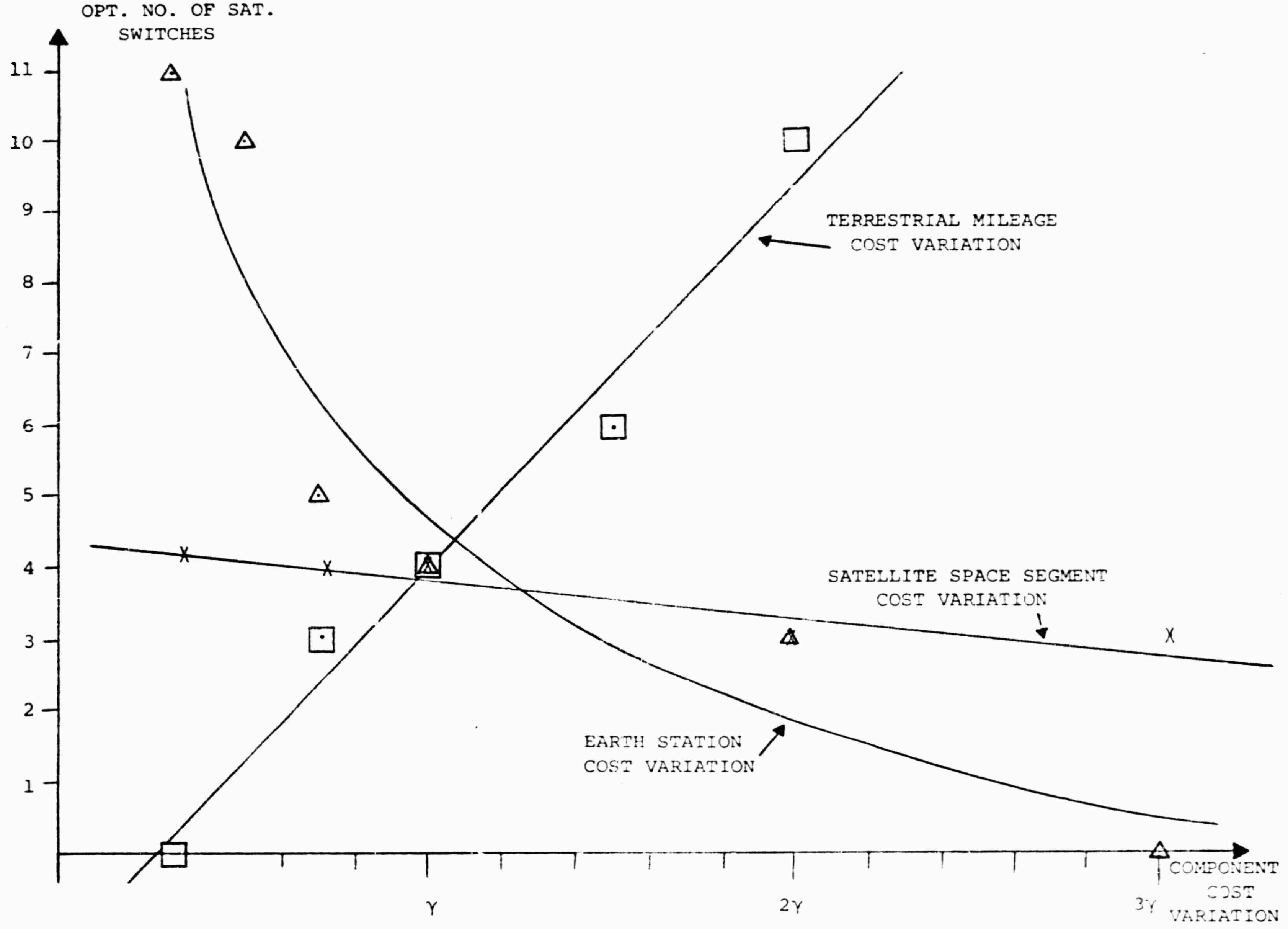


FIGURE 10: OPTIMAL NUMBER OF SATELLITE SWITCHES AS A FUNCTION OF INDIVIDUAL COMPONENT COST VARIATIONS, 11 BACKBONE SWITCH SET

NOTE: For each component considered, γ denotes the ratio between actual unit cost and basic unit cost as listed in Table 2.

5.6.2 Effect of Component Cost Variations

We examine the following three cost components: earth station cost, space segment cost and terrestrial mileage cost. Figure 10 shows that the effect of unit component cost changes on the optimal satellite switch number. We observe that changes of either terrestrial mileage cost or earth station cost has a more pronounced impact on optimal number of satellite switches than changes in the unit space segment cost. Similar to results obtained in [NAC, 1976] we would expect that the effect of a unit component cost variation on the optimum number of satellite switches is roughly proportional to the rate of change of that component with respect to the number of satellite switches at the optimum point. A comparison between Figures 9 and 10 also supports this claim.

5.6.3 Effect of Traffic Level

In Figure 11, we plot the curve of optimum number of satellite switches as a function of the traffic level, for various values of unit earth station cost, for the 11 ADD-algorithm-generated backbone switch set.

As expected, for a given unit earth station cost, the optimal number of satellite switches increases as the traffic requirement level increases. Moreover, the satellite switch selection problem is only meaningful over a limited traffic range. When the traffic level is below this range, satellite communication is not cost-effective (i.e., no earth stations need be used), and when traffic level is above this range, earth stations should be used at all backbone switch sites.

It can also be seen from Figure 11 that, for a given traffic requirement level, the number of satellite switches decreases as the unit earth station cost increases.

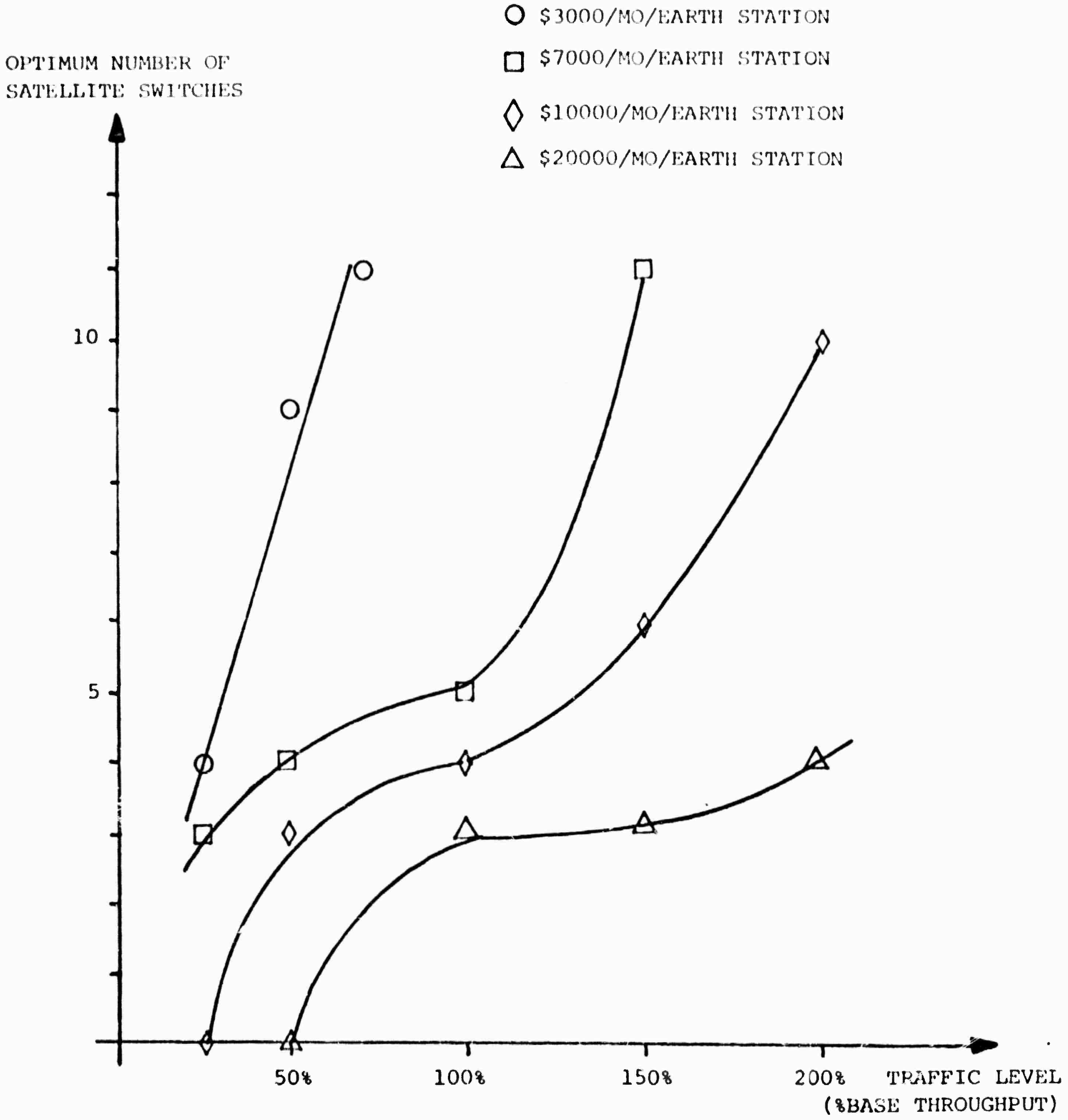


FIGURE 11: TRAFFIC REQUIREMENT LEVEL VS. OPTIMUM NUMBER OF SATELLITE SWITCHES, 11-SWITCH SET

The optimal satellite switch number versus traffic requirement curves in Figure 11 exhibit an interesting property: For each specific earth station unit cost, there seems to be a threshold traffic level below which the number of feasible satellite switches increases slowly as the traffic level grows, but beyond which the number of feasible satellite switches increases rapidly, until all switches are used as satellite switches. This can be explained intuitively as follows. For each system, there are usually some natural traffic concentration clusters. For example, in the AUTODIN II system, there is a cluster in the West coast area, a cluster in the Central area, a cluster in the Northeastern area, and a cluster in the Southeastern area. Nodes in the same cluster are much closer than nodes in different clusters. Thus, as traffic level increases, one node from each cluster quickly gets selected as the satellite switch site. Then for a large range of traffic values, very few additional nodes can be profitably used as satellite switch locations. Finally when the traffic level passes a certain threshold value, even nodes in the same cluster can communicate to each other effectively through satellite, thus many nodes become attractive for earth station sites almost simultaneously.

5.7 Procedure Run Time

The run time for the MODULARIZED ADD algorithm has been reported in [NAC, 1976]. For the present application, since we are not doing detailed backbone cost tradeoff evaluations (see Section 3), the run time should actually be several times faster.

The run times for the satellite switch selection algorithm are recorded in Table 5. As expected, the algorithm is quite slow. Thus, for switch sets with more than 15 switches, some of the satellite switches have to be preselected and some of the switches have to be excluded in order to have a reasonable run time.

SWITCH SET SIZE	RUN TIME*
8	0'5"
11	0'22"
16	15'44"

TABLE 5: CPU TIMES VS. NUMBER OF SWITCHES
SATELLITE SWITCH SELECTION ALGORITHM

Computer Used: PDP-10

*0'5" means 0 minutes and 5 seconds.

6. CONCLUSIONS

In this chapter, the problem of optimally locating the terrestrial and satellite switches in a large scale network environment is concisered. This problem is generally recognized to be an important factor in the overall optimization of packet switched network designs [GERLA, 1974b].

Two procedures are presented The first procedure selects the backbone switches from a set of candidate switches for a mixed media network. This procedure is a simple modification of the MODULARIZED ADD algorithm presented in [NAC, 1976] and is very efficient. The second procedure selects the satellite switches from the backbone switches. This procedure is basically exponential.

Mixed network designs based on the switch locations selected by the above two procedures were generated. It is found that mixed network designs generally offer considerable savings over network designs using only terrestrial technologies. Moreover, it was observed that the number of satellite switches selected is generally quite independent of the backbone switch set size. Hence, for a large switch set, the selection can be restricted to switch sets of size within a certain limited range.

Parametric studies were carried out to determine the sensitivity of the solutions to changes in some of the network parameters. It was found that, for a given backbone switch set, the optimum number of satellite switches is more sensitive to the variations in the unit earth station cost and terrestrial mileage cost, and less sensitive to variations in the unit satellite space segment cost. This is attributed to the fact that the rate of change of the earth station cost and the terrestrial mileage cost as a function of the number of satellite switches is higher than the rate of change of the space segment cost.

Some possible areas for further investigation are the following:

1. Development of more efficient algorithms for satellite switch selection (possible using clustering techniques). This is especially important for a large backbone switch set.

2. Extension of the algorithms to handle more general switch models (e.g., monlinear switch cost).

3. Development of lower bounds to assess the accuracy of heuristics.

REFERENCES

[ABRAMSON, 1973]	Abramson, N., "Packet Switching with Satellites," *AFIPS National Computer Conference Proceedings*, New York, Vol. 42, June 1973, pp. 695-702.
[BOOTHROYD, 1964]	Boothryod, J., "Algorithm 246 Gray Code [Z]," *Communications of the ACM*, Vol. 7, No. 12, December 1964, p. 701.
[CHOU, 1973]	Chou, W. and A Kershenbaum, "A Unified Algorithm for Designing Multidrop Teleprocessing Networks," *Data Networks: Analysis and Design Third Data Communications Symposium*, St. Petersburg, Florida, November 13-15, 1973.
[ESAU, 1966]	Esau, L.R. and K.C. Williams, "On Teleprocessing System Design, Part II," *IBM Systems J.*, Vol. 5, No. 2, 1966, pp. 142-147.
[FRANK, 1972]	Frank, H. and W. Chou, "Topological Optimization of Computer Networks," *Proceedings of the IEEE*, Vol. 60, No. 11, November 1972, pp. 1385-1397.
[GERLA, 1974a]	Gerla, M., H. Frank, W. Chou and J. Eckl, "A Cut Saturation Algorithm for Topological Design of Packet-Switched Communications Networks," *National Telecommunications Conference Proceedings*, San Diego, California, December 2-4, 1974.

[GERLA, 1974b] Gerla, N., W. Chou and H. Frank, "Cost Throughput Trends in Computer Networks Using Satellite Communications," National Telecommunications Conference Proceedings, San Diego, California, December 2-4, 1974.

[GERLA, 1975] Gerla, M., H. Frank, W. Chou and J. Eckl, "Design Alternatives for Large Distributed Networks," New Orleans, Louisiana, National Telecommunications Conference Proceedings, December 1975.

[GSCHWIND, 1975] Gschwind, H. and E.J. McCluskey, Design of Digital Computers, 2nd Ed., Springer-Verlag, Inc., New York, 1975.

[HEART, 1973] Heart, F., et al "A New Minicomputer/Multiprocessor for the ARPA Network," AFIPS National Computer Conference Proceedings, Vol. 42, 1973, pp. 529-537.

[HUYNH, 1976] Huynh, D., H. Kobayashi, and F.F. Kuo, "Design Issues for Mixed Media Packet Switching Networks," AFIPS National Computer Conference Proceedings, Vol. 45, June 1976, pp. 541-549.

[KLEINROCK, 1973] Kelinrock, L. and S.S. Lam, "Packet-Switching in a Slotted Satellite Channel," AFIPS National Computer Conference Proceedings, Vol. 42, June 1973, pp. 703-710.

[LAM, 1974] Lam, S.S., "Packet-Switching in a Multi-Access Broadcast Channel With Application to Satellite Communication in a Computer Network," Ph.d. Thesis, UCLA; also avail. as UCLA-ENG-7429, UCLA Computer Science Department, April 1974.

[LAVIA, 1975] Lavia, A. and E. Manning, "Perturbation Techniques for Topological Optimization of Computer Networks," Proceedings of the Fourth ACM Data Communications Symposium, Quebec, Canada, October 7-9, 1975.

[MCGREGOR, 1975] McGregor, P. and S , "Locating Concentration Points in Data Commu .ions Networking," Proceedings of the Fourth ACM Data Communications Symposium, Quebec, Canada, October 7-9, 1975.

[NAC, 1976] Network Analysis Corporation, Recent Advances in Large Scale Integrated Networks, Sixth Semiannual Technical Report, Vol. 3, Chapter 2, January 1976.

[ROBERTS, 1973] Roberts, L.G., "Dynamic Allocation of Satellite Capacity Through Packet Reservation," AFIPS National Computer Conference Proceedings, New York, Vol. 42, June 1973, pp. 711-716.

[WOO, 1973] Woo, L.S. and D.T. Tang, "Optimization of Teleprocessing Networks with Concentrators," National Telecommunications Conference Proceedings Atlanta, Georgia, 1973.

APPENDIX A

GENERATION OF REFLECTED BINARY CYCLIC CODES (GRAY CODES)

In this Appendix we derive a procedure for generating the successive reflected binary cyclic codes (Gray codes). This procedure is essentially the same as the one given in [BOOTHROYD, 1964].

The reflected binary code may be considered as an unusual notation for binary counts. Its structure can be seen in Table A.1.

Counting in the reflected binary code, we start in the same manner as in the true binary number system: 0,1. Then forced to introduce a second position, we reverse the count in the first position: 11, 10. Installing the third position, we repeat all previous counts in the first two positions in inverted sequence: 110, 111, 101, 100. Following the same principle, we obtain the first four-digit number 1100 and count up to 1000, the highest four-digit number. Points at which we install a new position and reverse the count are marked by a horizontal line in Table A.1. It can be seen that the reflected binary code is cyclic, i.e., when counting from one number to the next, only one digit is changed.

Let $(g_n, g_{n-1}, \ldots, g_1)$ be the k^{th} Gray code word (where g_n is the most significant digit and g_1 is the least significant digit) and $(b_n, \ldots, b_1)$ is the k^{th} binary code word. It can be shown that [GSCHWIND, 1975];

$$\begin{aligned} g_i &= b_i \oplus b_{i+1}, \text{ for } i = 1, \ldots, n-1 \\ g_n &= b_n, \end{aligned} \tag{A.1}$$

where $\oplus$ denotes mod 2 addition.

Let $(g_n, g_{n-1}, \ldots, g_1)$ and $(\bar{g}_n, \bar{g}_{n-1}, \ldots, \bar{g}_1)$ be two consecutive Gray codes and $(b_n, b_{n-1}, \ldots, b_1)$ and $(\bar{b}_n, \bar{b}_{n-1}, \ldots, \bar{b}_1)$

REFLECTED BINARY CODE $g_4g_3g_2g_1$	BINARY EQUIVALENT $b_4b_3b_2b_1$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
$0\ 0\ \overline{1\ 1}$	0 0 1 0
0 0 1 0	0 0 1 1
$0\ 1\ \overline{1\ 0}$	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
$1\ \overline{1\ 0\ 0}$	1 0 0 0
etc.	etc.

TABLE A.1: REFLECTED VERSUS TRUE BINARY NUMBERS be the two corresponding consecutive binary codes. Let i be the smallest k such that $b_k = 0$ (i.e., $b_i = 0$ and $b_{i-1} = \ldots = b_1 = 1$). Then it can be easily shown that

1. $\bar{b}_k = b_k$, for $k = i + 1, \ldots, n$
2. $\bar{b}_i = 1$

and

3. $\bar{b}_k = 0$, for $k = 1, \ldots, i - 1$.

Combining the above with Formula (A.1), we obtain the following relationship between $(\bar{g})$ and (g):

1. $\bar{g}_k = g_k$, for $k = i + 1, \ldots, n$;
2. $\bar{g}_i = \bar{b}_i \oplus \bar{b}_{i+1} = b_i \oplus 1 \oplus b_{i+1} = g_i \oplus 1$;

and for $i \geq 2$,

3. $\bar{g}_{i-1} = g_{i-1} = 1$;
4. $\bar{g}_k = g_k = 0$, for $k = 1, \ldots, i - 2$.

Consequently, we have the following two cases:

<u>CASE 1</u> $i = 1$.

In this case, the pattern is;

$$\begin{tikzcd}
b_n \cdots b_2 0 \arrow[r, leftrightarrow] \arrow[d] & g_n \cdots g_2 g_1 \arrow[d] \\
\bar{b}_n \cdots b_2 1 \arrow[r, leftrightarrow] & g_n \cdots g_2 \bar{g}_1
\end{tikzcd}$$

It follows that $g_1 = b_2$ and $\bar{g}_1 = b_2 \oplus 1 = g_1 \oplus 1$.

CASE 2 $\quad i \geq 2$.

In this case, the pattern is;

$$\begin{tikzcd} b_n \cdots b_{i+1}\; 0\; 1 \cdots 1 \arrow[r, leftrightarrow] \arrow[d] & g_n \cdots g_{i+1}\; g_i\; 1\; 0 \cdots 0 \arrow[d] \\ b_n \cdots b_{i+1}\; 1\; 0 \cdots 0 \arrow[r, leftrightarrow] & g_n \cdots g_{i+1}\; \bar{g}_i\; 1\; 0 \cdots 0 \end{tikzcd}$$

It follows that $g_i = b_{i+1}$ and $\bar{g}_i = b_{i+1} \oplus 1 = g_i \oplus 1$.

Notice that Case 1 occurs for every other code in the sequence (every even number has $b_1 = 0$). Also, in Case 2, $i-1$ corresponds to the smallest k such that $g_k = 1$. These observations lead to the following Gray code generation procedure. This procedure is essentially the same as the one given in [BOOTHROYD, 1964].

Procedure for Gray Code Generation

1. Initialization:

   $g_i \leftarrow 0$ for $i = 1, \ldots, n+1$.

   $E \leftarrow 1$.

2. If $E = 1$, then $i \leftarrow 1$;
   otherwise, let $i-1$ be the smallest k such that $g_k = 1$.

3. $g_i \leftarrow g_i \oplus 1$

   $E \leftarrow E \oplus 1$.

4. If $g_{n+1} = 1$, then Done;
   otherwise go to 2.

Notice that in the above procedure, the stopping condition is effected when $g_{n+1} = 1$ (which signifies that all n-bit codes have been enumerated).

Next, we consider the time complexity of the above procedure. Notice that for the n-bit Gray code sequences, there are 2^{n-1} codes with $E = 1$, 2^{n-2} codes with $g_1 = 1$, $E = 0$, 2^{n-3} codes with $g_2 = 1$, $g_1 = 0$, $E = 0$, and in general 2^{n-k} codes with $g_{k-1} = 1$, $g_{k-2} = \ldots = g_1 = 0$, $E = 0$. Now, for a code word with $g_{k-1} = 1$, $g_{k-2} = \ldots = g_1 = 0$, $E = 0$, the amount of work involved in getting the next code is: k-1 steps to find that $g_{k-1} = 1$ and $g_{k-2} = \ldots = g_1 = 0$, 1 step in setting g_k to $g_k \oplus 1$, and 1 step in changing E to $E \oplus 1$. Hence, the total amount of computation involved is $\sum_{i=1}^{n} (i+1)\, 2^{n-i}$.

In the following assertion, we show that $\sum_{i=0}^{n} i\, 2^{n-i} = 2^{n+1}-n-2$.

Hence, the total computation is essentially $2^{n+1} + 2^{n}$.

ASSERTION

$$\sum_{i=0}^{n} i\, 2^{n-i} = 2^{n+1} - n-2.$$

PROOF

$$\sum_{i=0}^{n} i\, 2^{n-i} = \sum_{i=0}^{n} n 2^{n-i} - \sum_{i=0}^{n} (n-i)\, 2^{n-i}$$

$$= n \sum_{i=0}^{n} 2^{n-i} - \sum_{i=0}^{n} i 2^{i}$$

$$= n \overset{n}{\underset{i=0}{}} 2^{i} - \overset{n}{\underset{i=0}{}} i 2^{i}. \qquad \text{(I)}$$

Now, $$\sum_{i=o}^{n} x^i = \frac{x^{n+1}-1}{x-1}. \qquad (II)$$

Thus, we have

$$\sum_{i=o}^{n} 2^i = 2^{n+1} - 1. \qquad (III)$$

Moreover, differentiating (II), we obtain

$$\sum_{i=1}^{n} ix^{i-1} = \frac{nx^{n+1}-(n+1)x^n+1}{(x-1)^2}.$$

Consequently,

$$\sum_{i=1}^{n} i2^{i-1} = n2^{n+1} - (n+1)\ 2^n + 1,$$

i.e.,

$$\sum_{i=o}^{n} i2^i = n2^{n+2} - (n+1)\ 2^{n+1} + 2. \qquad (IV)$$

Combining (I), (III) and (IV), we obtain

$$\sum_{i=o}^{n} i2^{n-i} = n(2^{n+1} - 1) - [n\ 2^{n+2} - (n+1)\ 2^{n+1} + 2]$$

$$= 2^{n+1} - n-2. \qquad \text{Q.E.D}$$

Attached is a copy of the Gray code generation procedure written in RATFOR, and a sample output with $n = 5$.

```
TY GRAY.RAT

;   <NAC>GRAY.RAT;8    WED 19-MAY-76 4:26PM                          PAGE 1
#
#       INVERTED BINARY CYCLIC(GRAY)CODE
#
#
          DIMENSION J(20),JJ(20)
DEFINE TTY 5
#
#    N IS THE WORD LENGTH
#    E CHANGES FROM 1 TO 0 AND VICE VERSA ON EACH CYCLE
#    I1 DENOTES THE BIT TO BE CHANGED
#
16600   FORMAT(1X,20I1)
#
#
        WRITE(TTY,1)
1       FORMAT('    ENTER VALUE FOR N   '$)
        CALL FFREAD
        CALL GET(1,N,' N ')
#
#    INITIALIZATION
#
        E=1
        M=N+1
        DO I=1,M;  [J(I)=0]
#
#    START ITERATION
#
        WHILE(J(M)!=1)
#
#    INVERT CODE SEQ. AND PRINT GRAY CODE
#
        [  DO I=1,N
           [    II=N-I+1
                JJ(I)=J(II)
           ]
           WRITE(5,16600)(JJ(I),I=1,N)
#
#    SEARCH FOR I1
#
           I1=1
           IF(E!=1)
           [  K=1
              WHILE(J(K)!=1)
              [    K=K+1   ]
              I1=K+1
           ]
#
#    INVERT BIT I1
#
           IF(J(I1)==1)  J(I1)=0
              ELSE       J(I1)=1
```

```
↑L
;   <NAC>GRAY.RAT;3    WED 19-MAY-76 4:26PM                         PAGE 1:1
#
#    INVERT E
#
                IF(E==1)        E=0
                   ELSE         E=1
           }
           STOP
           END
↑L
@



GRAY.SAV

  ENTER VALUE FOR N  <5

00000
00001
00011
00010
00110
00111
00101
00100
01100
01101
01111
01110
01010
01011
01001
01000
11000
11001
11011
11010
11110
11111
11101
11100
10100
10101
10111
10110
10010
10011
10001
10000

CPU TIME: 0.59  ELAPSED TIME: 5.46
NO EXECUTION ERRORS DETECTED

EXIT.
↑C
@
```

APPENDIX B

A BOUND ON THE SATELLITE SWITCH SELECTION

Here we discuss several other possibilities in reducing the problem size of the satellite switch selection problem presented in Section 4.

Given two switch locations, A and A'. Suppose that a satellite switch has been implemented at location A'. Intuitively, if location A is too close to A', it will not be cost-effective to implement a satellite switch at A. In the following, we derive a bound on this "closeness".

Take any switch location B. Let S(B) be the satellite switch for B (S(B) may be B itself).

We consider the possible (terrestrial and/or satellite) line saving of adding a satellite switch at A.

Let

$C_{A,B}$ = Cost of transmitting a unit of flow from A to B, assuming no satellite switch at A,

S(A) = Nearest satellite switch to A,

$\overline{C}_{A,B}$ = Cost of transmitting a unit of flow from A to B, assuming a satellite switch at A.

Then,

$$C^{TER}_{A,S(A)} \leq C^{TER}_{A,A'},$$

$$C_{A,B} = \min \{C^{TER}_{A,S(A)} + C^{TER}_{B,S(B)} + C_S, C^{TER}_{A,B}\},$$

and

$$\bar{C}_{A,B} = \min\{C^{TER}_{B,S(B)} + C_S,\ C^{TER}_{A,B}\}.$$

If

$$C^{TER}_{A,S(A)} + C^{TER}_{B,S(B)} + C_S \leq C^{TER}_{A,B},$$

then

$$C_{A,B} = C^{TER}_{A,S(A)} + C^{TER}_{B,S(B)} + C_S$$

$$\leq C^{TER}_{A,A'} + C^{TER}_{B,\text{[illegible]}} + C_S,$$

and

$$\bar{C}_{A,B} = C^{TER}_{B,S(B)} + C_S.$$

Hence, it follows that

$$C_{A,B} - \bar{C}_{A,B} \leq C^{TER}_{A,A'}.$$

On the other hand, if

$$C^{TER}_{A,S(A)} + C^{TER}_{B,S(B)} + C_S > C^{TER}_{A,B},$$

then

$$C_{A,B} = C^{TER}_{A,B}.$$

In this case, if

$$\bar{C}_{A,B} = C^{TER}_{A,B},$$

then

$$C_{A,B} - \bar{C}_{A,B} = 0 \leq C^{TER}_{A,A'};$$

and if

$$\bar{C}_{A,B} = C^{TER}_{B,S(B)} + C_S,$$

then

$$\begin{aligned} C_{A,B} - \bar{C}_{A,B} &= C^{TER}_{A,B} - (C^{TER}_{B,S(B)} + C_S) \\ &< (C^{TER}_{A,S(A)} + C^{TER}_{B,S(B)} + C_S) - (C^{TER}_{B,S(B)} + C_S) \\ &= C^{TER}_{A,S(A)} \\ &\leq C^{TER}_{A,A'}. \end{aligned}$$

We thus conclude that in all cases,

$$C_{A,B} - \bar{C}_{A,B} \leq C^{TER}_{A,A'}.$$

It follows that the total saving of implementing a satellite switch at A, SAVE, is bounded by;

$$\begin{aligned} SAVE &= (\text{line saving}) - (\text{ground station cost}) \\ &= [\sum_B t_{A,B}(C_{A,B} - \bar{C}_{A,B})] - C_{GRD} \\ &\leq [\sum_B t_{A,B} \times C^{TER}_{A,A'}] - C_{GRD} \\ &= t_A \times C^{TER}_{A,A'} - C_{GRD}' \end{aligned}$$

where

t_A = Total low priority traffic requirement at switch A,

$t_{A,B}$ = Low priority traffic requirement between A and B.

We thus have the following result:

ASSERTION

Give two switches A and A', suppose A' has a satellite switch. If

$$t_A \times C^{TER}_{A,A'} \leq C_{GRD} \tag{B.1}$$

(where t_A is the total low priority traffic requirement at A), then it is not cost-effective to implement a satellite switch at A.

Other problem size reduction bounds are possible. For example, suppose a switch is of some distance away from all other switches. Then most likely it would be cost-effective to implement a satellite switch at that location. However, the bound in this case appears to be not as good as that given in Eq.(B.1). Reduction consideration for a cluster of switches (e.g., at at most two satellite switches in a group of four switches) can also be developed along the same line as that for a single switch.

CHAPTER 2

AN ALGORITHM FOR DESIGN OF NONHIERARCHICAL CIRCUIT-SWITCHED NETWORKS

CHAPTER 2

TABLE OF CONTENTS

CHAPTER 2

TABLE OF CONTENTS: FIGURES

CHAPTER 2

TABLE OF CONTENTS: TABLES

CHAPTER 2

AN ALGORITHM FOR DESIGN OF NONHIERARCHICAL CIRCUIT-SWITCHED NETWORKS

1. INTRODUCTION

Over the last few months, attention has been directed at developing and programming an algorithm for accomplishing the design of a circuit switched network. In particular, effort has been focused at developing a program for designing a circuit switched network which is non-hierarchical and has a topology based upon some "minimum cost" criterion. It is desired that the network produced be capable of adequately handling integrated (mixed voice and data) traffic. Circuit switched network design for integrated traffic is, of course, of interest by itself. However, in addition, when the work in this area is coupled with our work on packet switched network design for mixed voice and data, insight can be gained for considering the network design problem when integrated traffic is to be handled by hybrid switching techniques.

This chapter reports on the development, to date, of a circuit switched network design program. The work carried out is summarized in the sequel as follows. Section 2 describes the "ideal" structure of a program for carrying out circuit switched network design. Section 3 deals with the actual programmed algorithm which has been developed. It describes the structure of the design program and discusses the philosophy of the design approach. Experiments that can be performed with the programmed algorithm in the near term are suggested in Section 4. The future development of the program is discussed in Section 5.

2. "IDEAL" PROGRAM STRUCTURE

It is worthwhile to describe the "ideal" structure of a program for designing a circuit switched network. Such a description serves two purposes:

1. It provides a standard against which to judge the structure of the actual programmed algorithm which has been developed,
2. It provides a model for use in developing design algorithms in the future.

Any procedure for carrying out a network design must begin with certain inputs. In the context of the circuit switched network design under consideration these inputs are taken to be:

1. The set of nodes, including their locations,
2. The traffic requirements; the node-to-node average traffic to be handled by the network,
3. A set of performance requirements to be satisfied by the design; this will almost always include some measure of "loss probability."

Ideally, the goal of a network design procedure is to specify a minimum cost network of the input node set which satisfies the traffic and performance requirements. Specification of a network of nodes is taken to mean:

1. Defining the connectivity; listing the node pairs which are connected by (directed or undirected) links,

2. Specifying the routing for the connectivity; listing the sequence of links traversed by traffic from a given source node to a given destination node,

3. Sizing the links - for each link listing the number of trunks required in the bundle - "the trunk engineering."

Cost, traffic and performance are related in a very complicated and extremely non-linear manner. Except in the simplest cases, it is all but impossible to obtain an optimum procedure for designing a minimum cost network . Instead, design procedures are heuristic with the goal of producing a network which satisfies traffic and performance requirements in an "economic" fashion - a network which is close to minimum in cost. Design procedures tend also to be iterative. Given inputs, an initial pass at design is made. This first pass design is then sequentially altered until the "designer" is satisfied with the resulting network cost and is convinced that traffic and performance requirements have been met.

The "ideal" structure of a program implementation of a general "heuristic" circuit switched network design algorithm should be similar to that illustrated in Figure 1. Here, "connectivity," "routing," "trunk engineering," "costing," and "performance analysis," are kept as separate and distinct modules in the program. Thus, the functional operation of each of these elements of the design procedure can be altered without affecting the other elements. This allows experimentation with the overall design procedure to be performed easily. For example, by keeping all modules the same, but by changing the function code in the costing module, the variation of final network design with different tariff structures can be observed. By keeping all modules the same, but changing the routing scheme, the effect on overall performance can be judged. The boxes labeled "instructed adjustments," are the variations ordered in the connectivity, routing and trunk engineering modules in order to allow cost to be reduced and/or performance requirements to be met. These adjustments might either be automated as program code (modules themselves) or might represent the effect of a "human designer" interacting with the programmed design procedure.

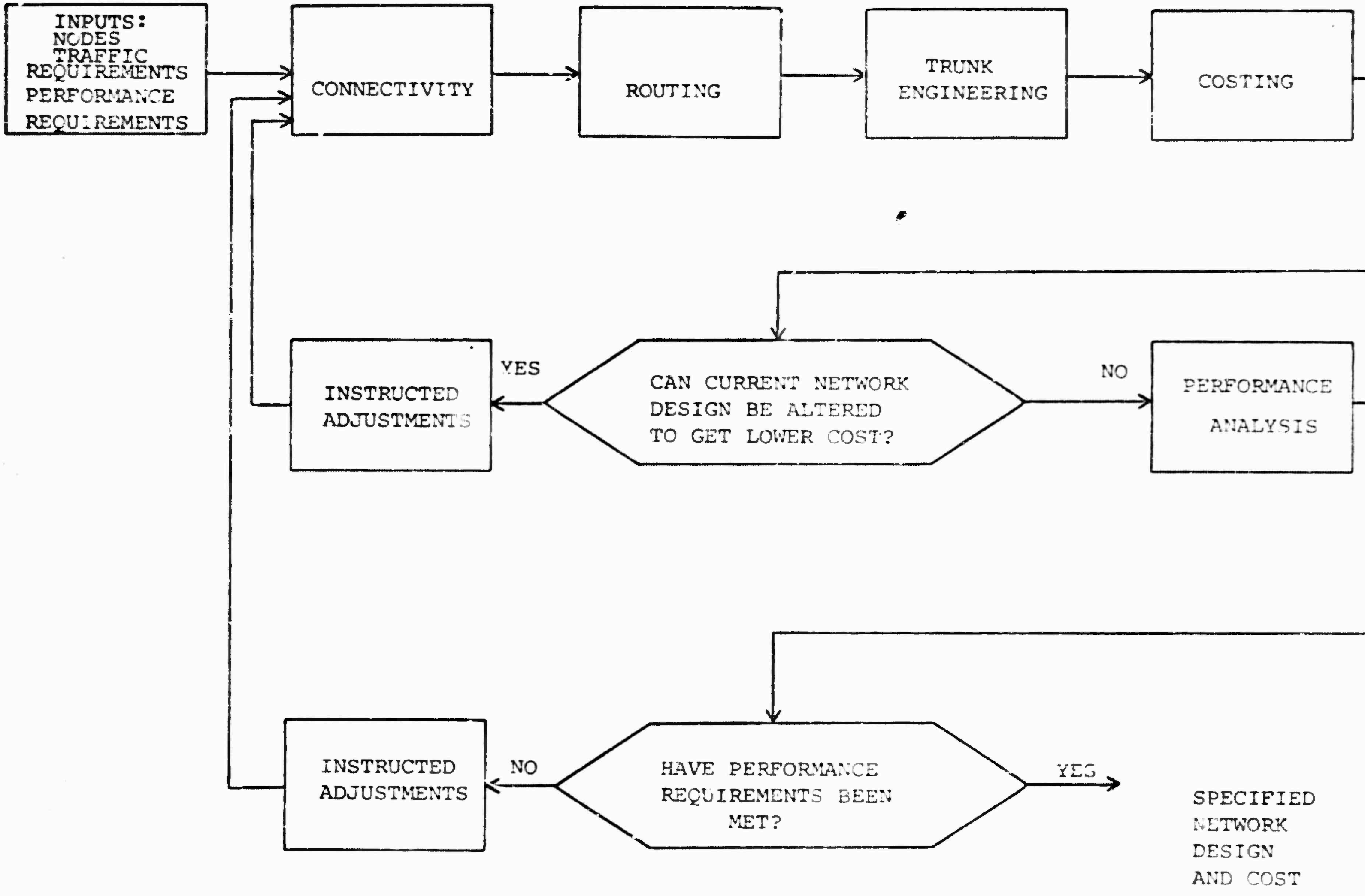


FIGURE 1: "IDEAL" STRUCTURE OF A CIRCUIT SWITCHED NETWORK DESIGN PROGRAM

3. THE CURRENT CIRCUIT SWITCHED NETWORK DESIGN PROGRAM

The circuit switched network design program which has been developed is based upon an algorithm of J.E. Knepley (KNEPLEY, 1973]. There are other approaches to the problem of circuit switched network design, e.g., [COVO, 1973], [KATZ, 1973]. However, Knepley's procedure provides an attractive starting point for a number of reasons. It is very general, dealing with a wide variety of design issues ranging from routing to trunk engineering. It has the flexibility to deal with many interesting design parameters such as tariff structures and switch cost. Finally, the design problem that the procedure addresses is that dealt with in the consideration of such existing networks as AT&T and AUTOVON.

A block diagram of the developed network design program is illustrated in Figure 2. Henceforth, this program will be referred to as "the baseline program." Although our ultimate goal is the handling of integrated traffic, to date the function code of individual blocks in the program has been generated considering only voice (or statistically equivalent) voice users. In the figure, the correspondence has been noted between blocks (or groups of blocks) of the baseline algorithm and blocks in the ideal structure of Figure 1. Observing Figure 2, the baseline algorithm consists, mainly, of four segments. The first is an "initialization segment." Here variables are defined, data read in and parameters are initialized. It ends with the "initialization box." The second is a "design segment." This extends from the box labeled iteration control to the "cost decision box." This segment is the "heart" of the design procedure. Connectivity, primary routing and trunk sizing are performed here. The third segment is the alternate routing module. At this point an alternate routing plan is specified for "the partial design" which has just been obtained. The last segment is the

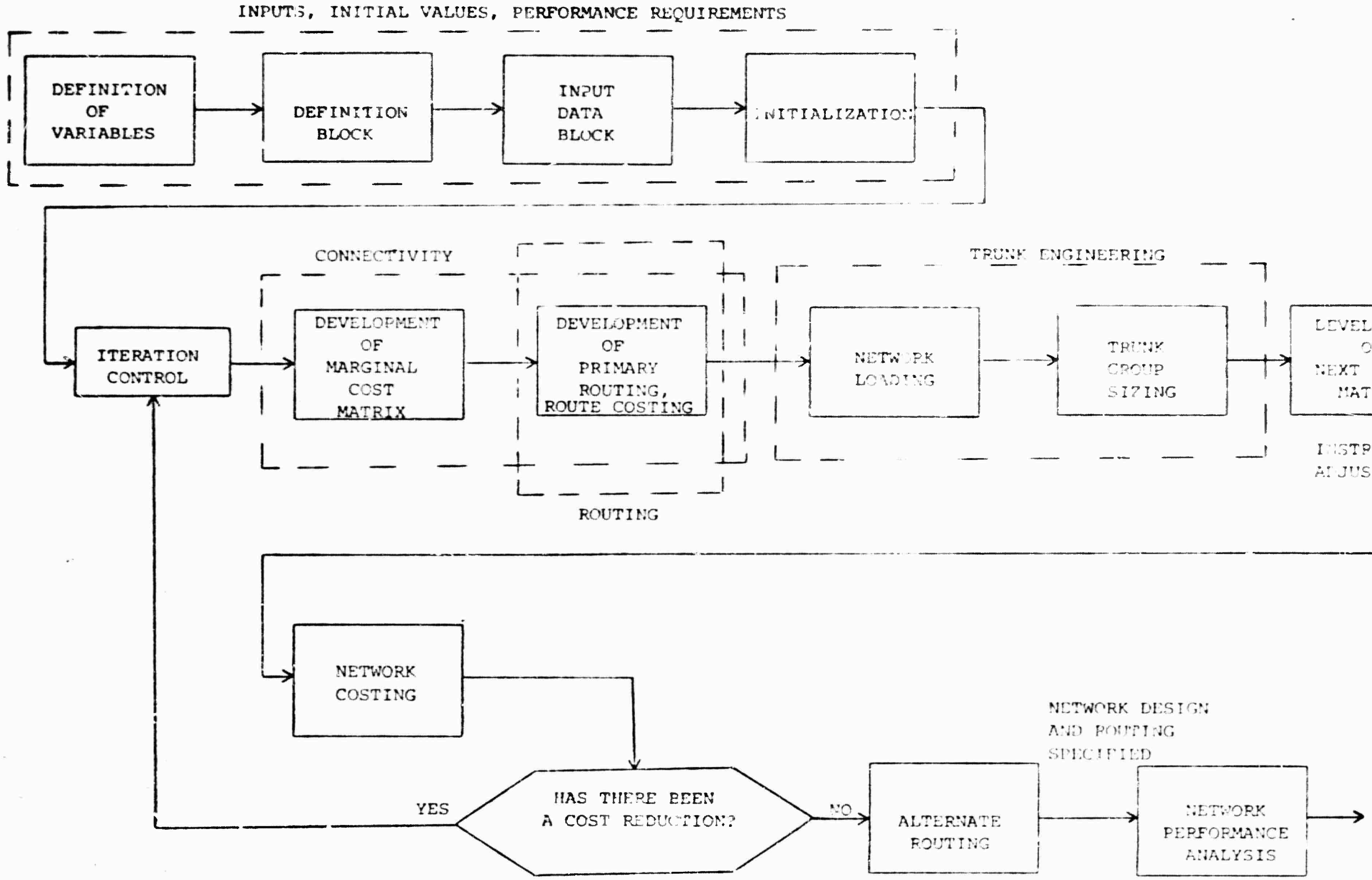


FIGURE 2: BLOCK DIAGRAM OF CIRCUIT SWITCHED NETWORK DESIGN PROGRAM IN CURRENT STATE

network performance analysis module. In this element the performance of the complete network design (connectivity, primary and alternate routing plans, trunk sizing) is determined, the average node-to-node loss probability is estimated. At present there is no automatic comparison in order to judge whether performance specifications have been met or have been met with too much margin. There is no feedback from this module to the design procedure.

A description of the operations carried out in each of the four segments of the program will now be given.

3.1 Initialization Segment

In this segment required constants (such as N, the number of nodes) are stored. Input data consisting of 1) the node labels with the latitude and longitude of each node and 2) the node-to-node traffic requirements are supplied in matrix form. From this input data the internode distances are computed and stored. To date, although this will change, the internode distance has been considered to be "true" link cost per trunk between two nodes. The internode distance matrix is equivalently a "true" cost matrix. Henceforth, it is called C(I,J).

At the end of this segment parameters are initialized for use in the subsequent "design segment." The "current" total cost of the network is set initially at infinity (10^{20} in the program). The procedure in the design segment requires an a priori value of link blocking probability. This is "E" and it is initialized here at some appropriate value, for example, at 0.005. In later program development it may be more convenient to read E from an input data file. The design procedure finds minimum cost routes based upon a marginal link cost per trunk rather than the true link cost per trunk. The marginal link cost per trunk, for a link between I and J is obtained using W(I,J), a link weighting. In the initialization block W(I,J) is set equal to 1 for all I,J. These "weights" are stored in a weight matrix.

3.2 Design Segment

It is worthwhile first, to describe, "in words," the design philosophy and total operation of this segment of the program. For the interested reader more detailed discussion of each of the program modules in this segment will then follow.

3.2.1 Overview

The design segment uses an iterative procedure in order to obtain a "minimum cost" network connectivity, primary routing and trunk sizing for the input node set and traffic requirements. The first iteration begins by connecting nodes together by paths composed of directed links. Specifically, each node pair, (I,J), is connected by the set of directed links which constitute the "least cost" path between node I and node J. Cost in this first iteration being the "true" link cost per trunk supplied in the previous segment. This "least cost" path is temporarily designated as the primary route between I and J. Thus, the design algorithm in this segment merges together the "connectivity" and "routing" elements of the network design procedure. Using the input node-to-node traffic requirements and the temporarily designated primary routing, each link of the "current network" (the network of this iteration) is loaded with traffic. The Erlang B formula in (1) is then used to determine the number of trunks required by each link in order to satisfy the a priori designated link blocking probability, B with this loaded traffic.

$$B = \frac{\frac{G^N}{N!}}{1 + G + \frac{G^2}{2!} + \frac{G^3}{3!} + \ldots + \frac{G^N}{N!}} \tag{1}$$

Here G is the traffic in Erlangs offered to a group of N trunks. "B" is the probability that an incoming call to such a trunk group will be blocked. The total "true cost" of the resulting network (connectivity, routing and trunk sizes have at this point been specified) is then computed and stored. In the last part of the iteration a new "marginal cost" is computed for each link (i.e., each node pair). This marginal cost is equal to the "true cost" times a weighting factor. The weighting factor, W_{ij}, for node pair (i,j) is given by

$$W_{ij} = \left. \frac{-\partial E_{s,a}/\partial a}{\partial E_{s,a}/\partial s} \right|_{E_{s,a}=E,\ s=s_{ij},\ a=a_{ij},\ i\neq j}$$

$$W_{ij} = \infty, \quad i=j$$

Here, $E_{s,a}$ is the Erlang B formula, E is the a priori designated value for the link blocking probability, s_{ij} is the number of trunks computed in this iteration for link (i,j) and a_{ij} is the traffic loading on link (i,j) computed in this iteration. With this weighting the marginal cost effectively considers links with large trunk group sizes to be more economical than links with small trunk group sizes. The second iteration is then begun. The procedure is the same as with the first iteration but with the "least cost" paths based upon the new marginal costs rather than the "true costs." Subsequent iterations continue in the same manner. At the end of each, a network design (connectivity, primary routing and trunk sizes) is produced, the true cost of the network design is computed and the link cost weighting factors to be used in the next iteration are determined. The iteration loop terminates when the computed network cost begins to change only by an insubstantial amount. The final (partial) network design produced by the design segment is that corresponding to the last iteration.

3.2.2 Detailed Description

A detailed description of each of the modules which implement the design segment will now be given.

3.2.2.1 Iteration Control

This block keeps track of the number of iterations that the design segment has gone through. It is entered at the beginning from the initialization block and subsequently by looping from the "Cost decision box."

3.2.2.2 Development of Marginal Cost Matrix Block

Here the marginal cost matrix for each particular iteration is constructed. The construction is carried out in two steps. In the first step the matrix entries are filled using the following formula

$$MC_{IJ} = C_{IJ}W_{IJ}$$

C_{IJ} is the "true" cost per trunk of link (I,J) and W_{IJ} is the current weighting of link (I,J). C_{IJ} and the first iteration value of W_{IJ} are obtained in the initialization segment. In the second step MC_{IJ} is set equal to infinity for all I=J. This prevents development of minimum cost routes having self loops at nodes.

3.2.2.3 Development of Primary Routing and Route Costing Block

In this block the "primary routing" is accomplished. For each node pair I,J where I≠J, the least marginal cost path between I and J is determined. That is, the least cost path between I and J is obtained where the link costs per trunk used are the marginal costs computed in the previous block. Floyd's algorithm for shortest paths is currently used for determining the "least marginal cost path." This may be replaced in the future by a more efficient algorithm. This routing establishes a connectivity of the nodes (i.e., tells where links should be placed) and specifies the primary routes or first routes which an incoming call will attempt to reserve in establishing an originating node - destination node connection. The current (for the iteration) marginal link costs per trunk corresponding to these primary routes are also stored.

3.2.2.4 Network Loading Block

At this point each of the links in the node connectivity defined by the previous block is considered. Using the input end-to-end traffic requirements, the total traffic offered to each link by all the primary routes which utilize the link is determined and stored.

3.2.2.5 Trunk Group Sizing Block

Here a computation is made of the minimum number of trunks required by each of the links (in the current node connectivity) in order to satisfy the a priori designated value of the link blocking probability, E. For a given link having an offered load of A erlangs as determined by the previous NETWORK LOADING BLOCK, the minimum number of trunks required, S, is determined recursively from the following relation

$$E_{s,A} = \frac{A \cdot E_{(s-1),A}}{s + A \cdot E_{(s-1),A}}$$

This relation is derived from the Erlang B Loss formula. Here, $E_{0,A}=1$. The desired minimum value, s, is the smallest value of s such that $E_{s,A} \le E$. The trunk size outputs are stored in a matrix.

3.2.2.6 Development of Next Weight Matrix Block

In this block the link weighting factors, $\{W_{ij}\}$, to be used in the next iteration of the design segment are determined. These are computed from the link loading and trunk sizing of the "design" established in the current iteration. These new link weighting factors allow the computation of new marginal link costs in the next iteration. The weighting factor, W_{ij}, corresponding to link (i,j) is computed using the following formula which is equivalent to $\frac{\partial E_{s,a}/\partial s}{\partial E_{s,a}/\partial a}$

$$W_{ij} = \left.\frac{a(1-E)-s}{a[\ln(a)-E-\psi(s'')]}\right|_{s=s_{ij},\ a=a_{ij}} .$$

Here,

1. s_{ij} is the number of trunks currently required by link (i,j) as determined by the previous trunk sizing block,

2. a_{ij} is the traffic offered to link (i,j) as determined by the previous network loading block,

3. s" is the integer portion of s_{ij} and $\psi(s'')$ is defined as follows

$$\psi(s'') = -\gamma \text{ when } s'' = 1$$

$$= -\gamma + \sum_{m-1}^{s''-1} m^{-1} \text{ when } s'' \geq 2$$

$\gamma = 0.5772156 \ldots$ (Euler's constant)

The outputs of this block are stored in the weight matrix, W(I,J), replacing the current entries of this matrix. These weights make the marginal cost smaller for efficient links utilizing many trunks and larger for inefficient links where few trunks are employed. Effectively, link replacement will ultimately be carried out by this change of the weight matrix. It is automated in this design program whereas in other design procedures it might be the task of a human designer interacting with the program.

3.2.2.7 Network Costing Block

Here the cost of the network design obtained in the current iteration is computed. The following formula is used to compute cost.

$$\text{Cost} = \sum_{\substack{i,j \\ i \neq j}} C_{ij} T_{ij}$$

T_{ij} is the number of trunks employed by link (i,j) in the current design.

3.2.2.8 Cost Decision Block

At this point the total true cost of the network produced in the current iteration (computed in the previous block) is compared to the true cost of the network produced in the last iteration. If the cost of the current network is lower (i.e., there has been a cost improvement) then the program loops back to the iteration control. It proceeds with the next iteration using the weighting matrix that has been most recently computed. If the cost of the current network is greater than or equal to the cost of the network in the previous iteration (i.e., no cost improvement) the program exists from the design loop. (The true cost, C, is initialized at ∞, hence there will be at least two iterations in the loop.) Upon exiting from the loop, the (partial) network design produced in the last iteration is printed out, that is, the matrix of primary routes and the trunk size matrix are printed. The true cost of this (partial) network produced by the design segment is also printed.

3.3 Alternate Routing Module

The design segment establishes the primary routing. For an incoming call at node "i" intended for node "j", the network routing control first attempts to reserve the links on the primary route between i and j in establishing the end-to-end connectivity. If this cannot be done (i.e., if one of the links is blocked), the call will be lost at this point unless there is a method of alternate routing. This module provides an alternate routing plan. For each end-to-end node pair it establishes a hierarchical order of alternate routes to handle overflow traffic from the primary route.

The alternate routing module uses as inputs:

1. The node connectivity established by the output of the design segment (the partial network design produced in the last iteration of the design segment),

2. The marginal link costs per trunk over the links of the primary routes of the design segment output.

The first procedure in the module is to determine which of the nodes are pendent or semi-pendent relative to the topology established by the output of the design segment.* The development of alternate routes for traffic from pendent and semi-pendent source nodes is handled differently from the development of alternate routes for traffic from "normal" (all other) source nodes.

The module handles the problem of alternate routing from the normal source nodes first. Consider the source node - destination node pair, i,j, where i is a normal node. The program determines the set of nodes {k} which are adjacent to node i (in the sense that there is a link from node i to each of the nodes in this set). The node in {k} which is adjacent to i, but also on the primary route from i to j is purged from this node set. For each of the remaining nodes the parameter d_{ikj} given by

$$d_{ikj} = \ell_{ik} + \ell_{kj}$$

is computed. Here ℓ_{ik} is the marginal cost per trunk of the link from i to k, ℓ_{kj} is the sum of the marginal costs per trunk of the links of the primary route from k to j. Each d_{ikj} is in correspondence with a possible alternate route from i to j. Specifically, d_{ik^*j} corresponds to the alternate route composed of the link (i,k*) followed by the primary route from k* to j. The set of d_{ikj}'s are

---

* Pendent nodes have only one node adjacent to them and can only be entered or exited using the link from that node. As used here, semi-pendent nodes may have more than one node adjacent to them, but can only be left through one link.

ordered from lowest to highest. The alternate route corresponding to the lowest d_{ikj} is considered and is checked to see if it contains any loops. If it does not it is accepted as the first in a hierarchy of alternate i-to-j routes. The procedure continues with the next lowest d_{ikj}, etc., until either the required number of alternate routes is obtained or until there are no more entries on the list of d_{ikj}'s. If a route corresponding to a particular d_{ikj} is found to loop it is not considered as a viable alternate route candidate and is discarded. There may be situations where for a particular i,j pair all alternate route candidates considered by the module are found to loop. The module notes this and the program prints this out as information to the user.

With the alternate routing from the normal source nodes completed the module handles the alternate routing from the pendent and semi-pendent nodes in the following manner. Consider the source node - destination node pair i,j where i is either a pendent or semi-pendent node. Let k be the (only) outward adjacent node to i. The module designates the first alternate route from i to j as being composed of the link (i,k) followed by the first alternate route from k to j. Other alternate routes are similarly designated.

3.4 Network Performance Analysis Module

3.4.1 Overview

In this module the network design (primary and alternate routing plans and trunk size specification) has its performance analyzed. This module has two outputs. The first is a matrix of estimates of the node-to-node loss probabilities. The second is an estimate of the "average" node-to-node loss probability. This is a weighted average of the matrix entries with the weights being the fraction of total traffic corresponding to each node pair.

The module requires as inputs,

1. The primary routing plan and trunk sizes specified by the design segment.

2. The alternate routing plan.

3. The node-to-node traffic requirements.

4. n, the number of primary and alternate routes for each end-to-end pair (assumed the same for each pair).

The actual computation of the loss probability output is a complicated, iterative procedure, and is best explained using the block diagram illustrated in Figure 3. The "inputs and initialization" box contains the inputs to the module as described above. The module also uses two matrices, OVRFLO(I,J) and CAP(I,J). Each has dimensions of number of nodes x number of nodes. OVRFLOW(I,J) is initially set equal to the traffic requirements matrix. CAP(I,J) is initially set equal to the trunk size matrix. The loop in the block diagram goes through n iterations. In the first iteration the end-to-end traffic is loaded onto the primary routes, overflow traffic is determined and the residual capacity of the network is also determined. In subsequent iterations, overflow traffic is systematically loaded onto the alternate routes until the hierarchy of routes is exhausted. The end-to-end loss probabilities are estimated from the remaining overflow traffic. The detailed operation of the first iteration is described below; a description of subsequent iterations follows.

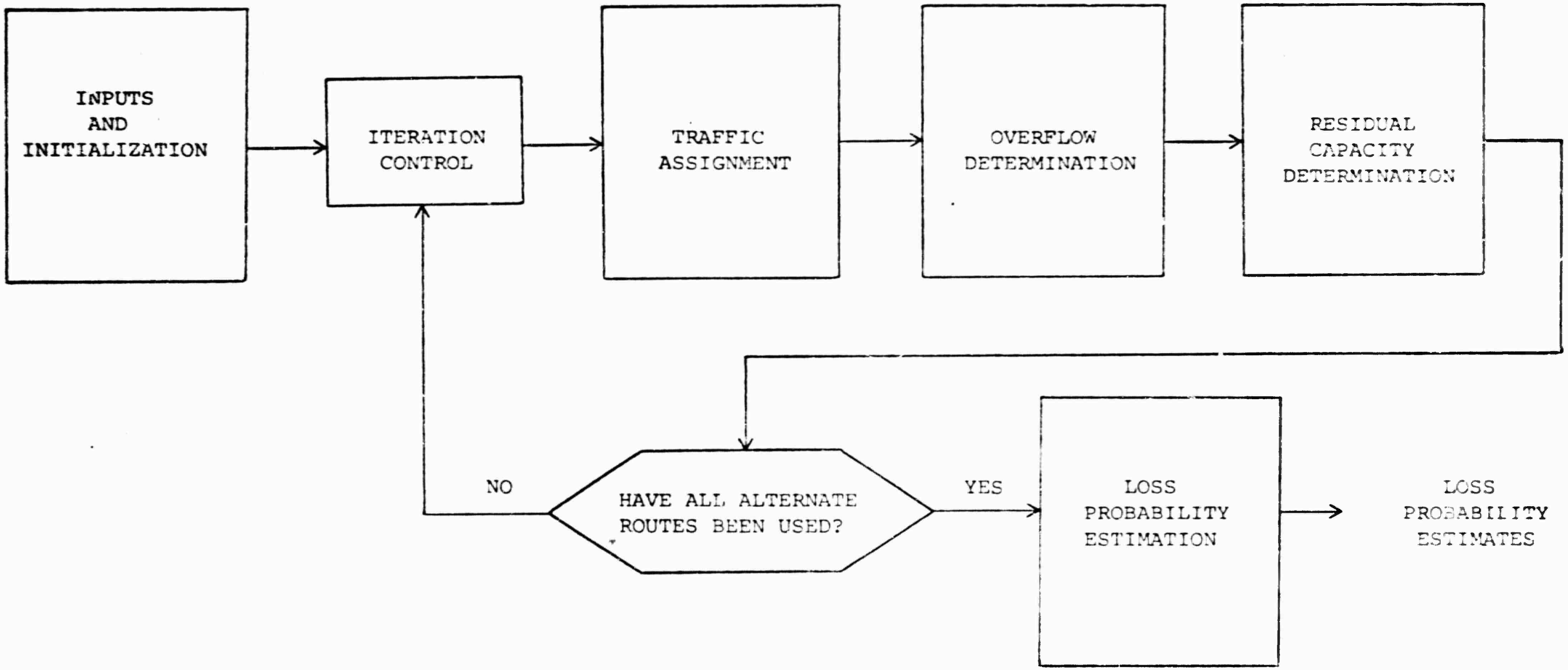


FIGURE 3: BLOCK DIAGRAM OF COMPONENTS OF NETWORK PERFORMANCE ANALYSIS MODULE

3.4.2 Traffic Assignment Block

Each link in the network design is dealt with separately. Consider link (i,j), the set of primary paths which uses this link is determined. The total traffic over these primary paths is then computed using OVERFLO(I,J) as the traffic requirements matrix (at this point it equals the traffic requirements matrix anyway). Let this total traffic be denoted by SUM(i,j). The primary paths which use (i,j) can be designated by the corresponding source node - destination node, (I,J). For link (i,j) and each of the corresponding (I,J)'s the following parameter is computed

$$ID_{(i,j)}(I,J) = \frac{\text{OVERFLO}(I,J)}{\text{SUM}(i,j)}\ \text{CAP}(i,j)$$

This is the amount of capacity (number of trunks) in link (i,j) that this network analysis procedure considers as being allocated to I-J traffic. This may be non-integral. However, the diaphontine problem is ignored at this point. Finally, in this block, for each end-to-end node pair I,J the parameter X(I,J) is computed. Here

$$X(I,J) = \min_{(i,j)} ID_{(i,j)}(I,J)$$

the minimum being taken over all links (i,j) in the primary route between source node I and destination node J. X(I,J) is the effective capacity of the primary route between I and J since the I-J traffic will be limited by that link in its primary path where the smallest number of trunks have been allocated to it. The analysis procedure considers X(I,J) trunks on each link in the primary path of (I,J) as being available for the initial end-to-end traffic requirements.

3.4.3 Overflow Determination Block

In this block, for each end-to-end node pair I,J, the amount of traffic which is turned away from the primary route (overflows it) is computed. The following formula is used to compute this overflow traffic:

$$\text{Overflow traffic from I-J primary path} = \text{OVERFLO}(I,J)\ E_{\lfloor X(I,J)+1 \rfloor,\ \text{OVERFLOW}(I,J)}$$

Here, $E_{\lfloor X(I,J)+1 \rfloor,\ \text{OVERFLOW}(I,J)}$ is the Erlang B formula evaluated with $\lfloor X(I,J)+1 \rfloor$ trunks and an offered load of OVERFLO(I,J). The symbol $\lfloor Y \rfloor$ represents the greatest integer contained in Y. The computed quantity "overflow traffic from I-J primary path" then replaces the I-J entry of the matrix OVERFLOW(,).

3.4.4 Residual Capacity Determination Block

In this block a determination is made of the network capacity remaining after the initial end-to-end traffic is loaded onto the primary routes by the method described above. This residual capacity is that available for overflow traffic being loaded onto the alternate routes. The operation of this block is quite simple. For each i,j, CAP(i,j), the i-j entry of the capacity matrix, CAP, is replaced by

$\text{CAP}(i,j) - \sum_{(I,J)} \text{ID}_{(i,j)}(I,J)$. The sum is taken over all end-to-end pairs which have primary routes using link (i,j).

3.4.5 The Decision Block

If there are only primary routes (i.e., no alternate routing) the loop is left at this point. Otherwise iterations will continue as described later.

3.4.6 Loss Probability Estimation Block

Here each entry in the last matrix OVRFLO(I,J) is divided by the corresponding entry in the input traffic requirements matrix. The resulting values are estimates of the end-to-end loss probability for each node pair This is then printed as the output of the entire module.

3.4.7 Subsequent Iterations

The second iteration uses the same procedure as the first with the exception that traffic is loaded onto the first alternate routes for each end-to-end node pair rather than the primary routes. Due to this, definition of the various parameters used must be appropriately changed. The other iterations follow in like manner, with the third iteration corresponding to the second alternate route, etc. When the entire hierarchy of alternate routes is exhausted the loop is exited and the loss probabilities estimated as previously described. The average node-to-node loss probability is then computed.

4. NEAR TERM EXPERIMENTS

There are a variety of experiments that can be performed in the near future (next one-to-two months) with the circuit switched network design program in its current state of development. These experiments can be divided into two categories,

1. Preliminary tests,

2. Circuit switching strategies.

They are described in the sequel. In the following paragraphs E, as before is the "link blocking probability." "P_L" is the (program provided) estimate of the average node-to-node loss probability.

4.1 Preliminary Tests

4.1.1 Sample Networks

In this experiment the existing program will be run with several different traffic matrices. The point of the experiment is to determine how global traffic patterns affect the topology of the network design output. One example of the results of this experiment is illustrated by Figure 4. The network topology illustrated by this figure was obtained when the design program was run with the traffic requirement matrix given in Table 1.

4.1.2 Variation of Performance With E

The current program requires E as an input. In one test E has been set equal to (0.5) 10^{-2} in hopes of obtaining a network design with performance specified by a P_L of 10^{-2}. Observations

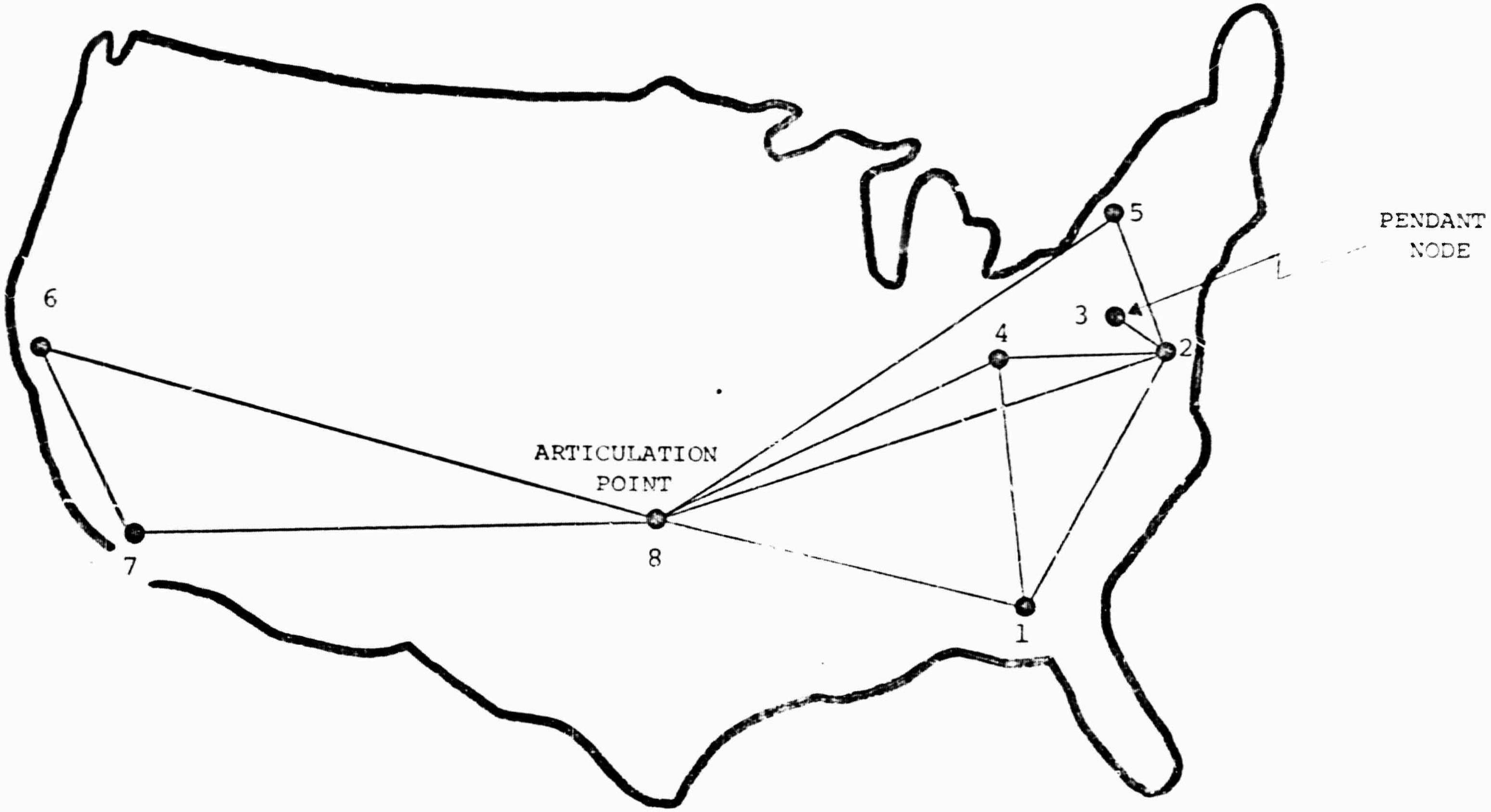


FIGURE 4: EXAMPLE OF NETWORK TOPOLOGY DESIGN PRODUCED BY CURRENT CIRCUIT SWITCHED NETWORK DESIGN PROGRAM

TABLE 1: Sample Traffic Req. Matrix (Values in Erlangs)

Nodes Nodes	1	2	3	4	5	6	7	8
1-	0.00000	3.86600	0.26900	1.128600	1.12800	0.74500	1.58400	6.05700
2-	5.9940	0.00000	2.16000	3.50600	5.94800	2.90000	2.88900	5.3700
3-	0.26200	2.29100	0.00000	0.43700	0.35000	0.75900	2.02000	1.20200
4-	2.14500	4.12600	0.48100	0.00000	1.35300	2.62100	1.69200	3.73400
5-	1.18600	7.13300	0.60700	1.30600	0.00000	1.09000	1.07300	2.92400
6-	1.05000	2.24200	0.68100	0.96100	0.84600	0.00000	1.47200	3.52600
7-	2.16600	3.65900	0.01700	1.02100	1.62700	2.38700	0.00000	1.95900
8-	7.69100	7.35800	1.22400	7.92000	4.78700	7.63100	3.78700	0.00000

to date, based upon some initial work, indicate that this has not always been met. In this experiment program structure will not be altered. However, final network performance (as represented by P_L) and network cost will be examined as E is varied through the following set, $(0.5)10^{-3}$, 10^{-3}, $(0.5)10^{-2}$. An example output of this experiment is illustrated in Figure 5. The curves shown illustrate P_L as a function of E for the network of Figure 4. Curve "a" corresponds to P_L being taken as the average node-to-node loss probability over all node pairs in the network. Curve 'b" corresponds to the average being taken over all those node pairs where the source node is not pendent or semi-pendent (i.e., excluding nodes 4 and 5. Node 5 is semi-pendent although not evident in The figure.) Node 8, in Figure 4, is an (non pendent node type) articulation point. Curve "c" is the P_L obtained by averaging over those nodes in the left hand partition of the network given by this articulation point. Curve "d" is the P_L obtained by averaging over those nodes in the right hand partition of the network.

4.2 Circuit Switching Strategies

4.2.1 Variation of P_L With the Degree of Alternate Routing

In this experiment the variation of P_L is examined as the number of alternate routes, provided for each node pair communication, is increased.

4.2.2 Performance Penalty for Alternate Routing

As Aaron Kershenbaum has pointed out alternate routing may provide a mixed blessing [KERSHENBAUM, 1976]. Alternate routed traffic may steal capacity from primary routed traffic. In this experiment

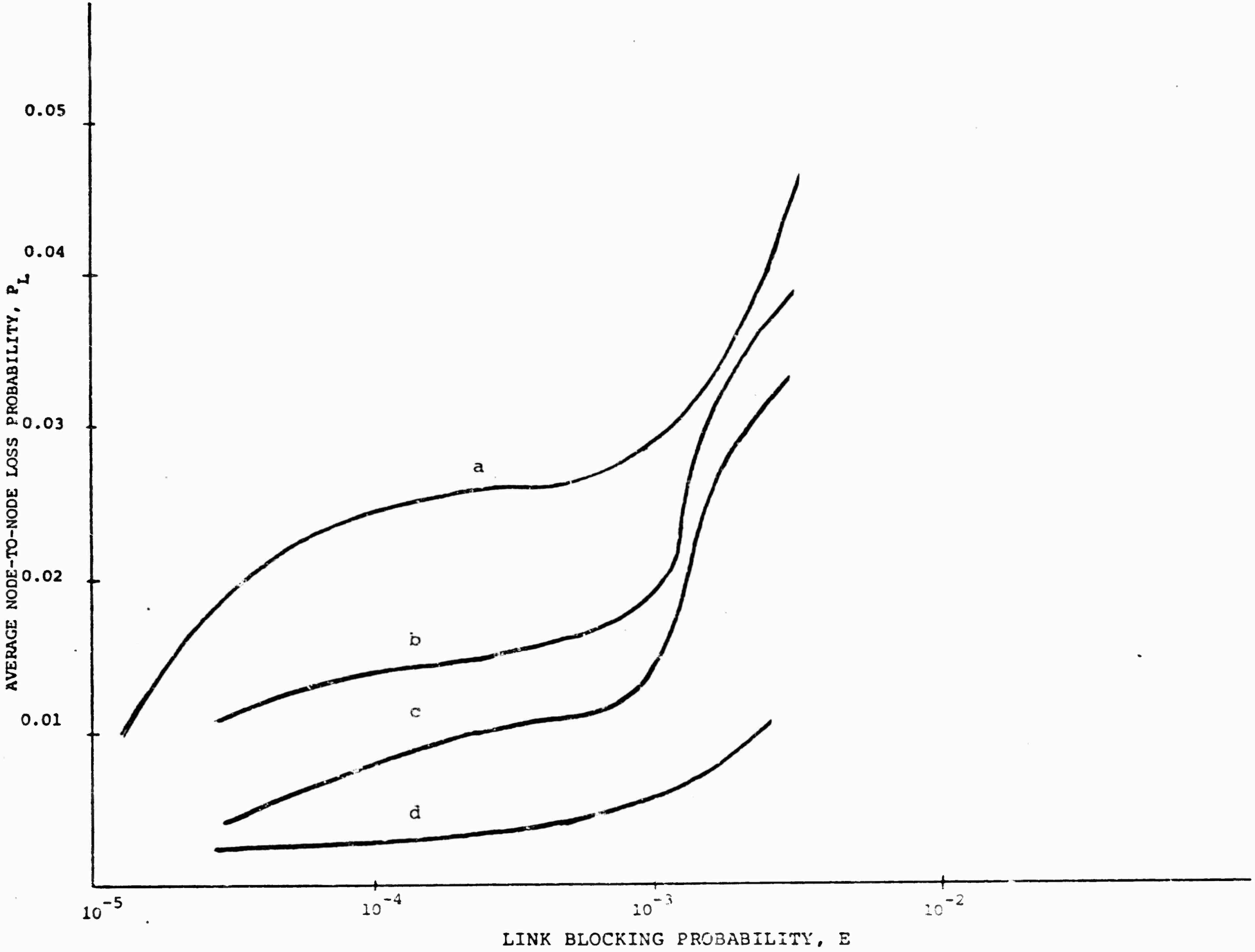


FIGURE 5: P_L VS. E FOR NETWORK OF FIGURE 4

the network performance analysis module of the existing program will be changed by penalizing alternate routed traffic. With each traffic loading iteration of this module the residual capacity will be decreased by some factor so as to account for the interaction of alternate and primary routed traffic in a pessimistic way. The variation of P_L with the capacity reduction factor will be examined. If P_L is very sensitive to the traffic interaction a new module will have to be substituted for the current method of performance analysis.

4.2.3 Effect of Making Links Bilateral

In this experiment the trunk group sizing block of the program is altered so that the output of the design segment is a network having bilateral (equal capacity in both directions) links. Each link will now have capacity equal to the maximum of what it was previously. The effect of this on network (cost) and P_L will be explored.

5. FUTURE DEVELOPMENT

In the months ahead the following topics will be addressed in the further development of the circuit switched network design program:

5.1 Reliability

A program module will be created which analyzes the reliability of the network design output.

5.2 Tariff Structure

The method of dealing with link cost per trunk will be changed so as to account for switch complexity.

5.3 Alternate Routing

A wide variety of other alternate routing schemes will be considered for use in the program. These will include "no alternate routing," "progressive routing," and alternate routing with total path diversity. Toward this end an exhaustive search and classification of the various types of routing strategies that can be used in telecommunication networking is being performed.

5.4 Mixed Voice-Data Traffic

Elements of the program will be suitably altered to account for the fact that traffic may be the desired integrated voice and data rather than pure voice (or statistically equivalent voice) which has been considered so far. In particular, the formula for computing link blocking probability will have to be changed. In making these alterations a variety of cases will be considered in order to test the robustness of the network design. Specifically, we will consider the following traffic situations:

1. Pure voice.
2. Heavy voice - light data.
3. Equal voice - equal data.
4. Light voice - heavy data.
5. Pure data.

5.5 Network Performance Analysis Module

This will be improved in order to obtain sharper estimates of the end-to-end loss probabilities.

5.6 Incorporation of the Node Model

The baseline program requires link blocking probability as an input. This is related to link loading and trunk sizing which are currently accounted for in the baseline program. However, it is also inherently related to the properties and capabilities of the network nodes that the assumed circuit switch models. In addition, link cost per trunk should include the cost of termination which is related to switch complexity. To date these items have been neglected and have never been taken into account in great detail in any investigation of circuit switched network design techniqes. However, future development of the baseline algorithm will take switch complexity into account. This will be a major study effort. A considerable amount of study of switch models has already been done. Specifically, an extensive survey and modeling effort is currently underway to investigate the cost performance tradeoffs associated with representative circuit switch architectures. Of necessity, this has entailed an assessment of most existing technology including:

Electromechanical systems,

Electronic systems,

Computer controlled systems,

and a variety of circuit switching techniques:

Space division,

Time division,

Hybrid space-time division.

The short term goal is to incorporate the relevant characteristics of many switches into a single generic switch model. This will, of course, subsume all ongoing work directed at specific models. The long term objectives are to investigate the impact of a given circuit switch architecture on network performance. The merits of specific architectures can, therefore, be more meaningfully determined in the presence of network constraints.

Two areas of particular importance in which the circuit switch can influence overall performance are internal call blocking and setup delay. Although more economical, circuit switch architectures that permit internal blocking can lead to intolerable end-to-end blocking, cost-benefit tradeoffs will be investigated. The additional constraint of switch setup delay will be studied in detail. Interrelationships between routing plans (fixed, alternate vs. adaptive), signalling techniques (inband vs. common channel), and the operating speed of the switch will be examined relative to acceptable values of cross-network (setup) delay. In addition to existing switch hardware, a hypothetical new generation of circuit switches will be used as the referrent in order to design circuit switched networks which can satisfy AUTOVON requirements. Appreciable cost reductions over the current system are expected to emerge. A rudimentary effort is als envisioned that will address the integration of a second traffic type (i.e., data) on existing and future circuit switch hardware and networks.

5.7 An Alternative Circuit Switch Design Approach

Whenever a new design program is developed it must be validated in terms of alternative models. NAC has a very effective program to design packet switched networks. The program employs the "flow deviation" method for assigning traffic to routes [GERLA, 1973]. "Flow Deviation" is an approach to assigning costs in a network by using convex cost functions to measure link saturation. It is proposed to use the same program to design circuit switched networks by deriving link cost functions which are measures of blocking probability [VAN SLYKE, 1976].

REFERENCES

[COVO, 1973] Covo, A. A., "Sizing of Military Circuit-Switched Communication Systems by Computer Aided Analytic Methods," *ICCC Conference Record*, Vol. II, 1973.

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

[KATZ, 1973] Katz, S., "Long Range Planning of Hierarchical Trunk Networks With Rearrangement Costs," *Proceedings of the International Teletraffic Congress*, 1973.

[KATZ, 1967] Katz, S., "Statistical Performance Analysis of a Swtiched Communication Network," *Proceedings of the International Teletraffic Congress*, 1967.

[KERSHENBAUM, 1967] Kershenbaum, A., Personnel Communication, 1967.

[KNEPLEY, 1973] Knepley, J. E., "Minimum Cost Design for Circuit Switched Networks," *Technical Note No. 36-73*, Defense Communication Agency System Engineering Facility, Reston, Virginia, July 1973.

[VAN SLYKE, 1976] Van Slyke, R., Personal Communication, 1976.

CHAPTER 3

ON CONNECTIVITY IN MOBILE PACKET RADIO NETWORKS

CHAPTER 3

TABLE OF CONTENTS

CHAPTER 3

TABLE OF CONTENTS: FIGURES

CHAPTER 3

TABLE OF CONTENTS: TABLES

CHAPTER 3

ON CONNECTIVITY IN MOBILE PACKET RADIO NETWORKS

1. INTRODUCTION

The Packet Radio Network (PRNET) is a store-and-forward packet switching radio system [KAHN, 1975; FRANK, 1975]. Functionally, it includes three types of devices: Terminal, Repeater, and Station. The repeater provides area coverage for mobile terminals and also acts as a relay node. The station provides global control functions, gateway functions for interfacing with other networks, and initialization functions. A centralized hierarchical routing algorithm described in [GITMAN, 1976] and implemented in the experimental PRNET is assumed in this chapter. According to this algorithm, packet transportation between any nodes in the network is via the station. Furthermore, the station initializes and periodically updates repeater parameters for routing.

One of the objectives of the system is to enable communication between mobile devices. Up to the present, investigations have been limited to the study of Stationary Packet Radio networks, which are partially characterized by the following two conditions:

1. Terminals, repeaters and stations are stationary.

2. The area in which terminals originate is fixed and is covered by appropriately placing the hardware at some initial time t_o.

On these assumptions, routing algorithms and initialization algorithms have been developed. In particular, the routing algorithms use information on the network configuration to derive an optimal communication path. In the case of general deployment, in which the repeaters and the station are not initially aware of the network configuration, some system capacity must be spent to acquire such information and initialize the devices with valid parameters.

The Mobile Packet Radio system generalizes the stationary system by allowing every element of the hardware to be independently in motion in the 3-dimensional space. Conditions 1 and 2 above must be dropped and substituted with appropriate requirements, to be described below. There is a clear distinction between the two systems, not only in the definition and analysis, but also the problems one must address. For example, in a Packet Radio Network (PRNET) one is trying to determine the location of the repeaters to achieve optimal coverage of a certain area. A repeater may be permanently placed in a precalculated location. In the Mobile Packet Radio Network (MPRNET) repeaters are assumed in motion.

In PRNET, a particular and predetermined zone is covered.

In MPRNET, the covered zone varies with time. Requirements as to the coverage of a particular area, may, however, be imposed.

In PRNET, once the system is initialized, it will remain so; only one initialization and labeling is necessary (except if there is a high failure rate of repeaters).

In MPRNET, the system may become inoperative due to motion of the devices. Hence, one must address himself to the questions of how long the system is operational, or how long one must wait before one gets coverage at a specified location. Consequently, a frequent MONITORING OPERATION is required by the station.

The dynamic character of this system accentuates the need to acquire, process and proliferate network configuration information. Specifically, the connectivity matrix will be altered within a short interval, depending on the mobility parameters of the system. Thus, the initialization algorithms developed for the stationary packet radio must be activated and more frequently, in addition to the more frequent acquisition of data, if the system is to be operational. As a consequence, the initialization capacity becomes considerable, decreasing, in conjunction, the useful information capacity.

It can be inferred from the statements above that the available algorithms may no longer be optimal in a dymanic environment, and hence, new or modified algorithms are sought.

A predictable mobile PRNET (station knows the trajectories of repeaters) is addressed in this chapter. The efficiency of a predictable system can be improved by changing some protocols in the network, as follows:

(i) Connectivity monitoring via ROP's can be reduced or eliminated.

(ii) The station can predict connectivity, and hence labels, and send new labels to repeaters at the proper times.

(iii) Alternatively, the station may send to a repeater a set of labels and associated time units for using each.

It is noted that the criterion for determining labels in a mobile network should not be based on "shortest path" only but take into account the interval of time during which a label will be valid.

A method for computing connectivity as a function of time and its computational complexity are presented. Approximate methods and their complexity are also given. A program which implements the exact method was developed (not presented in the chapter).

2. MATHEMATICAL FRAMEWORK

In this section, we introduce the mathematical framework for the MPRENT.

Definition: [Initial Operating Region]

Let $O(t_o)$ be an open 3-dimensional region, which we assume to be the inside of a cube.

Definition: [Hardware]

Let $S = \{S_1, S_2, \ldots, S_n\}$, $R = \{r_1, r_2, \ldots, r_m\}$,

$\tau = \{\tau_1, \tau_2, \ldots, \tau_L\}$ be sets satisfying:

a. $S \cap R = \phi$, $S \cap \tau = \phi$, $\tau \cap R = \phi$

b. $S \subset O(t_o)$

$S \subset O(t_o)$

$\tau \subset O(t_o)$

The sets S, R, and τ correspond to the set of stations, repeaters and terminals of a packet radio system, respectively. The requirement in a implies that each device fulfills only one function.

Definition: [Frame of Reference]

Let $X_{S_1, O(t_o)}$, $Y_{S_1, O(t_o)}$, $Z_{S_1, O(t_o)}$ be a cartesian coordinate system, with axes parallel to the sides of $O(t_o)$ and with origin at S_1.

Note that the frame of reference can be arbitrarily selected without loss of generality. However, it is convenient to select a coordinate system with the origin at station S_1 (packet radio networks under consideration are assumed to have at least one station). The set of free nodes is implied by the selection of the coordinate system, and is defined as:

Definition: [Free Nodes]

Let $\hat{S} = S - \{S_1\}$ and $F = \hat{S} \cup R \cup \tau$. The elements of F are called free nodes.

Definition: [Zone of Coverage]

Let $O(t)$ be the zone of desired coverage at time t. Possibly $O(t) = O(t_o) = O_o$ for all t.

Definition: [Area of Coverage]

Let P_{xy} be the x-y plane.

Then $P_{xy}(t) = O(t) \cap P_{xy}$ will be called the terrestrial area of desired coverage.

Definition: [Stationary Geosystem]

The system described by the definitions above is called a Stationary Geosystem, and will be denoted as:

$$\mathscr{S}_o = \{S, R, \tau, O_o, O\}$$

Definition: [Dynamic Geosystem]

Assign to each free node $f \in F$ a velocity vector with reference to S_1 as follows:

$$\vec{v}_f^{\,S_1}(t) = \vec{v}_f(t), \text{ with } |\vec{v}_t(t)| = v_f(t), \text{ and } \arg(\vec{v}_f(t)) = O_f(t)$$

A Dynamic Geosystem is the system described by the previous definitions and the above vector assignment. It will be denoted by $\mathscr{S}_V = \{S, R, \tau, O(t), V(t)\}$ where $V(t) = \{\vec{v}_f(t) \mid f \in F\}$.

Equivalently, specification of the displacement vector of motion, in terms of the x, y, and z components, could be given.

Note that the definition requires S_1 to be fixed in the selected frame of reference, but no generality is lost by this. In fact, if in relation to some more general frame of reference, we had associated v'_g for all $g \in S \cup R \cup \tau$, then in our specific frame of reference through S_1, we would have,

$\vec{v} = \vec{C}$ associated with S_1

$\vec{v}_g = \vec{v}'_g - \vec{v}'_{S_1}$ associated with $g \in \tilde{F}$

Assumption 1:

There exists a set $P_R = \{P_1, P_2, \ldots, F_z\}$ of transmission powers for repeaters and stations with a corresponding set $\mu_R = \{\mu_1, \mu_2, \ldots, \mu_z\}$ of transmission ranges (distances).

Assumption 2:

There exists a set $P_T = \{P_\nu\}$ of transmission power for terminals, with a corresponding set $\mu_T = \{\mu_\nu\}$ of transmission range. We assume that for equal levels of transmission power, equal transmission ranges are achieved. In other words, a flat terrain is considered.

The zone covered by transmission by an element $f \varepsilon F$ at level μ, is

$$B_f^{\mu}(t) = \{x \varepsilon O(t) \mid d_{f,x}(t) \le \mu\}$$

where d is the Eucledian distance in 3 dimensions and μ is a transmission range from the set μ_R or μ_T corresponding to whether f is a repeater or a terminal.

Definition:

$M^{\mu_k}(t) = (a_{i,j}^{\mu_k}(t))$, $1 \le i,j \le n + m + L$ is called the connectivity matrix when

$$a_{i,j}^{\mu_k}(t) = \begin{cases} 1 \text{ if } f_i \varepsilon B_j^{\mu_k}(t) \\ 0 \text{ otherwise} \end{cases} \qquad j \le m + n$$

$$a_{i,j}^{\mu_k}(t) = \begin{cases} 1 \text{ if } f_i \varepsilon B_j^{\nu}(t) \\ 0 \text{ otherwise} \end{cases} \qquad j > m + n$$

The top entries in the definition of $a_{i,j}$ read: "f_i is within the range of device j."

Definition: [Status]

By status of the geosystem at time t, $\sigma(t)$, we mean its connectivity matrix, and the covered zone at level μ.

Definition: [Potentially Operational System]

The system is potentially operational at time t if the following hold:

1. "Potentially operational W.R.T. terminals:"

   $O(t)$ is covered by balls of radius ν centered on r_i, and

2. "Potentially operational W.R.T. repeaters:"

   Consider the graph with vertices r_i and edges between r_i and r_j if $d_{r_i,r_j} \leq \mu_k$; then the graph is connected.

Let $G(t)$ be the zone for which the system is potentially operational at t. Obviously, we would like $O(t) \subseteq G(t)$.

Definition: [Actually Operational System]

A system is actually operational if:

1. It is potentially operational, and

2. If the algorithms and parameters used by devices for packet transportation are such that there is a communication path between any pair of nodes. (The communication paths need not be optimal.)

This definition is motivated by the fact that even though a system may be potentially operational (the hardware is in the right position), no relabeling has yet taken place - either because the connectivity matrix in the station is not up to date or because the station has not yet completely proliferated the information to the repeaters.

Definition: [Predictable System]

A system is predictable if:

$\sigma(t) = q(\sigma(t_o), v_{f_1}(t), \ldots, v_{f_\lambda}(t))$ with q known; that is, the status at time t can be determined by the station by computation alone.

Definition: [Unpredictable system]

A system is unpredictable if either q or $\vec{v}$'s are not known.

3. CONNECTIVITY PROBLEMS

A number of pertinent problems can be raised at this point. The remainder of this chapter is dedicated to the solution of various connectivity issues in the predictable case. The following questions are of interest:

1. The interval of time during which repeater r_i can communicate with the station S in 1-hop;

2. The interval during which repeater r_i can communicate with the station (in any number of hops);

3. The interval of time during which the entire network is connected;

4. The degradation history of the network (i.e., when at most, at least, or exactly k repeaters are unable to communicate with the station); and

5. The enhancement history of the network (i.e., when at most, etc. k repeaters return within the range of the station).

3.1 Definitions

1. a. A set $\Pi \subset \P$ ($\P$= real numbers) is said to be connected if for all $x,y \in \Pi$, $(1-\alpha)x + \alpha y \in \Pi$ for all $0 \leq \alpha \leq 1$.

   b. $\{\Pi_i\}_{n}^{i=1}$ are the connected components of Π if Π_i are the equivalence classes of the following equivalence relation:

$$x \sim y \text{ iff } \{(1-\alpha)x + \alpha y \mid 0 \leq \alpha \leq 1\} \subset \Pi$$

2. The L spectrum of Π with connected components $\{\Pi_i\}$, $^*t(\Pi)$, is the following collection:

$$^*t(\Pi) = \{t_i \mid t_i = \inf_{t\epsilon\Pi_i} t\} \qquad \text{if } \Pi \neq \phi$$

$$^*t(\Pi) = \{\infty\} \qquad \text{if } \Pi = \phi$$

The points of $^*t(\Pi)$ are called the death dates of Π.

3. The R spectrum of a set Π with components $\{\Pi_i\}$, $t_*(\Pi)$, is the following collection:

$$t_*(\Pi) = \{t_i \mid t_i = \sup_{t\epsilon\Pi_i} t\} \qquad \text{if } \Pi \neq \phi$$

$$t_*(\Pi) = \{-\infty\} \qquad \text{if } \Pi = \phi$$

The points of $t_*(\Pi)$ are called the resurection dates of Π. In this section, we are concerned only with systems having a single station.

4. Let $R = \{r_1, r_2, \ldots, r_j\}$. The Joint Life Status (JLS), $(r_1, r_2, \ldots, r_j)$ is said to be alive at time t if every repeater can communicate with the station.

5. The JLS is said to experience a k^{th} order failure, $k \leq j$, if at least k repeaters cannot communicate with the station.

6. Let $r_o = S$, and $r_1, r_2, \ldots, r_j$ be the repeaters of a geosystem. The set

$$\Xi_{ik} = \{t \mid d(r_i, r_k) > \mu\}, \ 0 \leq i, \ k \leq j\}$$

represents the time when there is no direct connection between r_i and r_k.

Figure 1 depicts the set Ξ_{10} for various mobility characteristics of a 1-repeater geosystem.

7. a. Let ${}^{i}\Pi$ be the time intervals such that repeater r_i cannot communicate with the station in any number of hops.

b. Let $\Pi^i, \Omega^i, \Lambda^i$ be the time intervals when at least, exactly, at most i repeaters, respectively, cannot communicate with the station in any number of hops.

c. Let α^i, β^i, γ^i be the time instance when exactly, at least, at most i repeaters come back into range of the station ("resurect").

We shall assume that the displacement vector of every repeater r_i is given; namely,

$$r_i : (x_i(t), \ y_i(t), \ z_i(t))$$

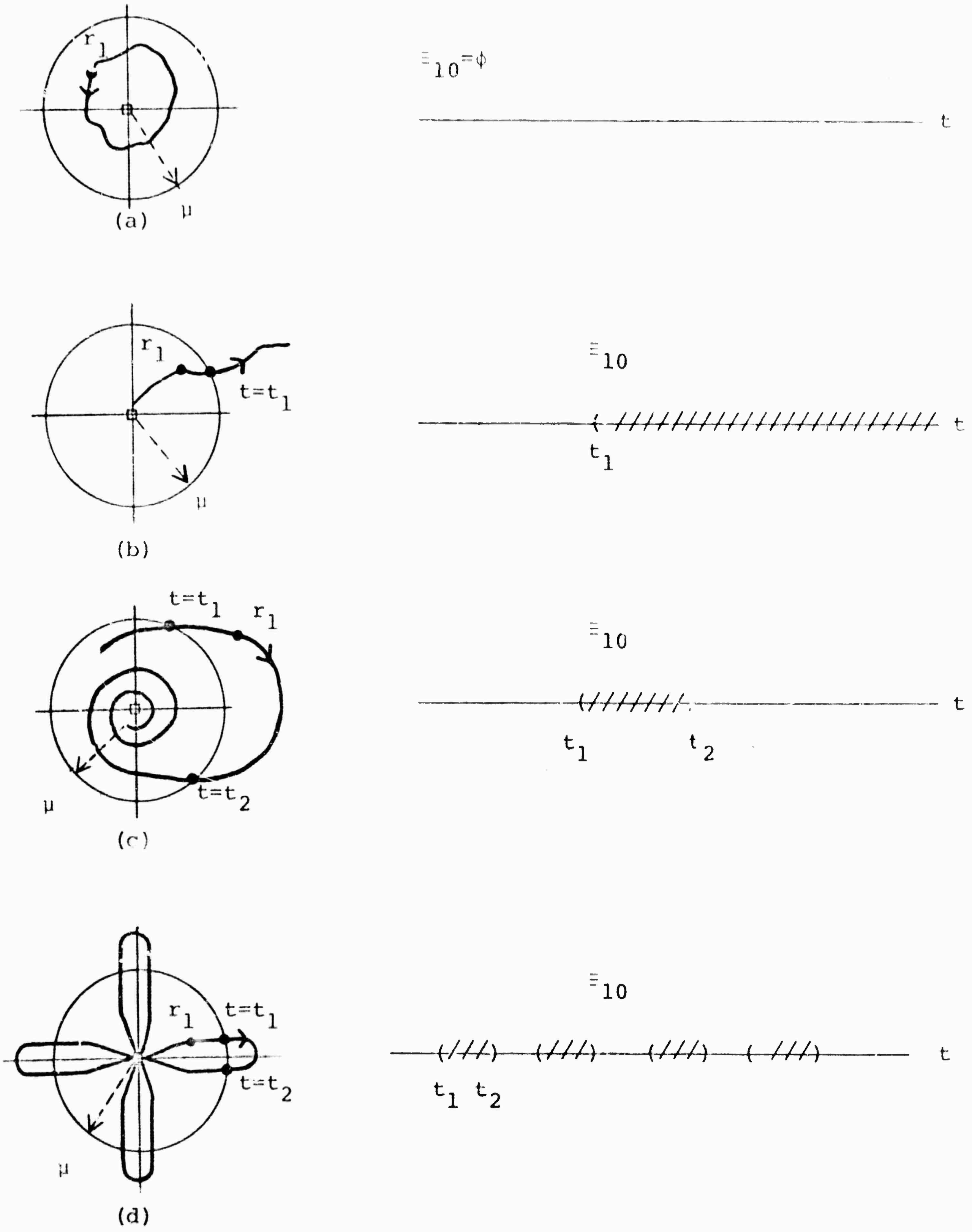


FIGURE 1: SETS Ξ_{10} FOR VARIOUS 1-REPEATER GEOSYSTEM

3.2 Motivational Example

Consider a 3 repeater system. We wish to derive ${}^1\Pi$, the time repeater r_1 is disconnected from the station S. Such an event would materialize when there are no 1-hop connections, there are no 2-hop connections, and no 3-hop connections. Each event (except the first one) can be further subdivided into subevents, relating to the particular path structure. A brief analysis shows that:

$${}^1\Pi = \Xi_{10} \cap (\Xi_{12} \cup \Xi_{20}) \cap (\Xi_{13} \cup \Xi_{30}) \cap (\Xi_{12} \cup \Xi_{23} \cup \Xi_{30}) \cap (\Xi_{13} \cup \Xi_{32} \cup \Xi_{20})$$

Similar expressions are obtained for ${}^2\Pi$ and ${}^3\Pi$. The JLS experiences a first order failure when either r_1 is outside the range of S, or r_2 is outside the range, or r_3 is outside the range. Thus,

$$\Pi^1 = {}^1\Pi \cup {}^2\Pi \cup {}^3\Pi$$

Similarly,

$$\Pi^2 = ({}^1\Pi \cap {}^2\Pi) \cup ({}^1\Pi \cap {}^3\Pi) \cup ({}^2\Pi \cap {}^3\Pi),$$

$$\Pi^3 = {}^1\Pi \cap {}^2\Pi \cap {}^3\Pi$$

It is clear that the quantities at hand can all be derived from knowing the $\frac{1}{2}\, j(j+1)$ distinct Ξ_{ik} sets. A general methodology is now presented.

4. GENERAL THEORY

The connectivity quantities described above can be obtained as follows:

Step 1:

Derive Ξ_{ik} for all $0 \le i \le k \le j$ by solving the cartesian equation:

$$\Xi_{ik} = \{t \mid (x_i(t) - x_k(t))^2 + (y_i(t) - y_k(t))^2 + (z_i(t) - z_k(t))^2 > \mu^2\}$$

Step 2:

Derive ${}^h\Pi$, $1 \le h \le j$ as follows:

$${}^h\Pi = \bigcap_{k=1}^{j} \bigcap_{\substack{1 \le i_1, i_2, \dots, i_k \le j \\ i_2 \ne i_1 \\ i_3 \ne i_2, i_1 \\ \vdots \\ i_k \ne i_{k-1}, i_{k-2}, \dots, i_1}} (\Xi_{h,i_1} \cup \Xi_{i_1,i_2} \cup \dots \cup \Xi_{i_k,0})$$

Namely,

$${}^h\Pi = \Xi_{ho} \cap \Big[\bigcap_{\substack{1 \le k \le j \\ k \ne h}} (\Xi_{hk} \cup \Xi_{ko})\Big] \cap \Big[\bigcap_{\substack{1 \le k, L \le j \\ k \ne h \\ i \ne k, h}} (\Xi_{hk} \cup \Xi_{kL} \cup \Xi_{io})\Big] \dots$$

Step 3:

Construct Π^h, $1\le h\le j$, as follows:

1. $\Pi^1 = \bigcup_i {}^i\Pi$.
2. $\Pi^2 = \bigcup_{1\le i\le k\le j} ({}^i\Pi \cap {}^k\Pi)$
3. $\Pi^3 = \bigcup_{1\le i\le k\le h\le j} ({}^i\Pi \cap {}^j\Pi \cap {}^h\Pi)$

et cetera.

Step 4:

Construct Ω^h, $1\le h\le j-1$, as follows:

$\Omega^h = \Pi^h - \Pi^{h+1}$

$\Omega^j = \Pi^j$

Step 5:

Construct Λ^h, $1\le h\le j$, as follows:

$$\Lambda^h = \bigcup_{i=1}^{h} \Omega^i$$

4.1 Properties

1. $\Pi^1 \supset \Pi^2 \supset \Pi^3 \supset \ldots \Pi^j$
For example, ${}^i\Pi \cap {}^k\Pi \subset {}^i\Pi$, so that $\Pi^2 = \bigcup({}^i\Pi \cap {}^k\Pi) \subset \bigcup {}^i\Pi = \Pi^1$

2. Π^h represents the time when h or more repeaters are outside the range of the station; Π^{h+1} represents the time when h+1 or more repeaters are outside the range of S; hence, indeed $\Omega^h = \Pi^h - \Pi^{h+1}$.

3. Ω^h's are disjoint sets.

4. Π^{o} = time when at least zero repeaters are outside the range, so $\Pi^{o} - (-\infty,\infty)$.

5. Ω^{o} = time exactly zero repeaters are dead; i.e., every repeater can communicate with the station; it follows that:

$$\Omega^{o} = \Pi^{o} - \Pi^{1} = \P - \Pi^{1} = C\,(\Pi^{1}).$$

where C(Λ) is the complement of set Λ.

6. $\Omega^{o} = \Lambda^{o}$

4.2 Complexity Considerations

We define complexity to be the number of computational steps required to carry out a given task. In this context, the term "step" does not refer to a specific operation; it could refer to a multiplication or to a union, etc.

Consider a system having j repeaters and one station. The complexity of Step 1 is $\frac{1}{2}j(j+1)$. (Step refers to the number of distinct sets to be obtained). The complexity of Step 2 is:

$$\chi = \sum_{k=1}^{j} \frac{j!}{(j-k)!}\, k$$

Since there are $\frac{j!}{(j-k)!}$ paths of k, each admitting k possible disconnections. The complexity of Step 3 is:

$$\sum_{k=1}^{j} \binom{j}{k} = 2^{j} - 1$$

(step refers to unions and intersections). Clearly, Step 2 is the most complex of the three. The following bounds are easily obtained.

$$(2j-1)j! \leq \chi \leq e.j.j!$$

4.3 Approximate Solutions

In view of the high computational complexity of the exact solutions, approximate solutions are sought. Two procedures are proposed.

Cumulative Approximating Procedures of Degree d (CAP:d)

In this approach, a repeater r_h is declared to be dead if it is unable to communicate with the station in d hops or less. It follows that:

$$ {}^{h}\Pi \approx \bigcap_{k=1}^{d} \bigcap_{\substack{1\le i_1,i_2,\dots,i_k\le j \\ i_2\ne i_1 \\ i_3\ne i_2,i_1 \\ \vdots \\ i_k\ne i_{k-1},i_{k-2},\dots,i_1}} (\Xi_{h,i_1}\cup\Xi_{i_1,i_2}\cup,\dots,\cup\,\Xi_{i_k,0}) $$

Note that the outer intersection encompasses d-or-less-hop connections to the station.

Sectional Approximation Procedure of Degree d (SAP:d)

In this approach, a repeater r_h is termed dead unless there is a d-hop connection to the station. Then,

$$ {}^{h}\Pi \approx \bigcap_{\substack{1\le i_1,i_2,\dots,i_d\le j \\ i_2\ne i_1 \\ i_3\ne i_2,i_1 \\ \vdots \\ i_d\ne i_{d-1},\dots,i_1}} (\Xi_{h,i_1}\cup\Xi_{i_1,i_2}\cup,\dots,\cup\,\Xi_{i_d,0}) $$

Under the CAP:d, the computational complexity is reduced from approximately $(2j-1)j!$ to approximately j^d. The CAP approach is superior to the SAP approach, and under mild conditions, the CAP:2 is an excellent approximation to the actual answer. With this procedure of complexity j^2, up to 30 repeater systems can be analyzed. Other approximating schemes are available.

4.4 Resurection Dates

The instance when an inactive repeater comes back into range is useful information, in particular, in a predictable system. This section addresses the issue.

It is clear that exactly one repeater resurects when the state changes from having i dead repeaters to i-1 dead. Similarly, exactly two repeaters resurect when the state changes, from having i dead repeaters to i-2 dead repeaters, etc, for $i>2$. Thus, it must follow that:

$$\alpha^1 = \bigcup_{i=0}^{j-1} (t_*(\Omega^{i+1}) \cap {}_*t(\Omega^i))$$

$$\cdots$$

$$\alpha^p = \bigcup_{i=0}^{j-p} (t_*(\Omega^{i+p}) \cap {}_*t(\Omega^i))$$

Then,

$$\beta^p = \bigcup_{k=1}^{p} \alpha^k$$

$$\gamma^p = \bigcup_{k=p}^{j} \alpha^k$$

5. EXAMPLES

Example 1:

Let

$$R_j : (\alpha_i t+\delta_i,\ \beta_i t+\gamma_i) \qquad i = 1,\ 2$$

let

$$\Lambda(a_1,b_1,c_1,d_1) = \Lambda(1) = 2a_1b_1c_1d_1 + a_1^2(\mu^2-c_1^2) + b_1^2(\mu^2 - d_1^2)$$

$$\nabla_{\pm}(a_1,b_1,c_1,d_1) = \nabla(1) = \frac{(-a_1d_1+b_1c_1) \pm \sqrt{\Lambda(a_1,b_1,c_1,d_1)}}{a_1^2 + b_1^2}$$

now

$$\Xi_{ij} = \{t \mid [(\alpha_i-\alpha_j)t + (\delta_i-\delta_j)]^2 + [(\beta_i-\beta_j)t + (\gamma_i-\gamma_j)]^2 \geq \mu^2\}$$

$$= [-\infty,\ \nabla_-(i;j)] \cup [\nabla_+(i;j),\ \infty]$$

where

$$\nabla_{\pm}(i;j) = \nabla_{\pm}(\alpha_i-\alpha_j, \beta_i-\beta_j, \gamma_i-\gamma_j, \delta_i, \delta_j)$$

We can now get the desired connectivity measures by the procedure of Section 4.

Example 2:

The mobile network under scrutiny consists of 5 repeaters moving on linear trajectories, within a two dimensional plane at low constant velocity. Such motion is completely characterized by specifying the four parameters of Table 1, where the units of time have been taken to be hours, and the units of velocity MPH. The motion is relative to the station.

Figure 2 describes graphically the trajectories followed, and Figure 3 depicts the status of the system at three sample points. The dashed circle represents the range of transmission of the devices.

The solution of the problem is furnished by a computer program which analyzes such dynamic systems. Topics of interest are:

1. The interval of time during which repeater r_i can communicate with the station S in 1-hop;

2. The interval during which repeater r_i can communicate with the station (in any number of hops);

3. The interval of time during which the entire network is connected;

4. The degradation history of the network (i.e., when at most, at least, or exactly k repeaters are unable to communicate with the station); and

5. The enhancement history of the network (i.e., when at most, etc. k repeaters return within the range of the station).

Table 2 shows the results for Items 1, 2, 3, and 4b above for the example at hand.

	X LOCATION AT T=0	Y LOCATION AT T=0	SLOPE	VELOCITY, MPH
REPEATER 1	-11.0000	-11.0000	-1.000	.3000
REPEATER 2	-5.0000	-5.0000	.5000	.1000
REPEATER 3	-5.0000	5.0000	-5.0000	.2000
REPEATER 4	15.0000	0.0000	2.0000	-.2500
REPEATER 5	11.0000	11.0000	1.0000	.2000

TABLE 1: CHARACTERIZATION OF REPEATER MOTION RELATIVE TO THE STATION

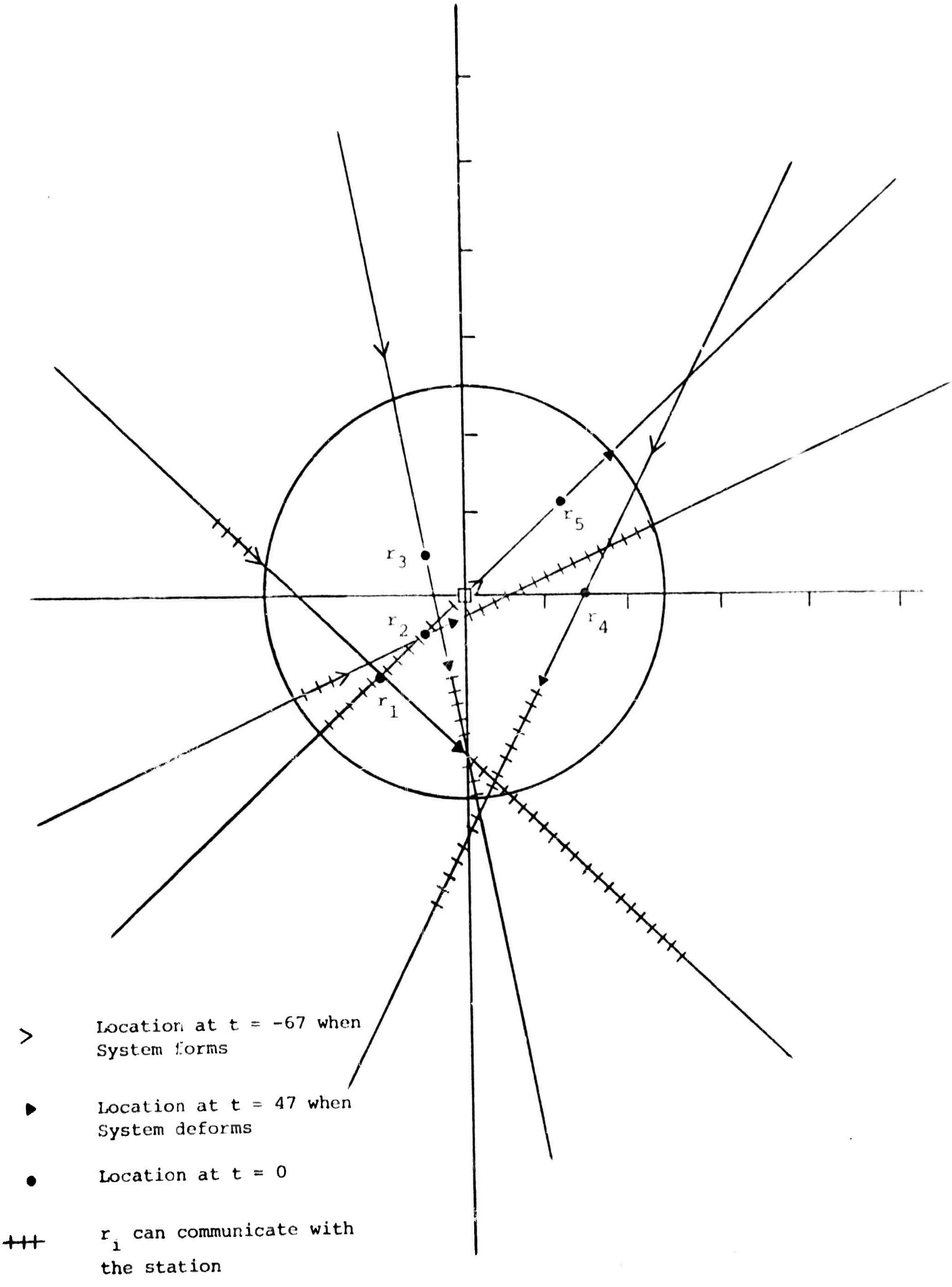
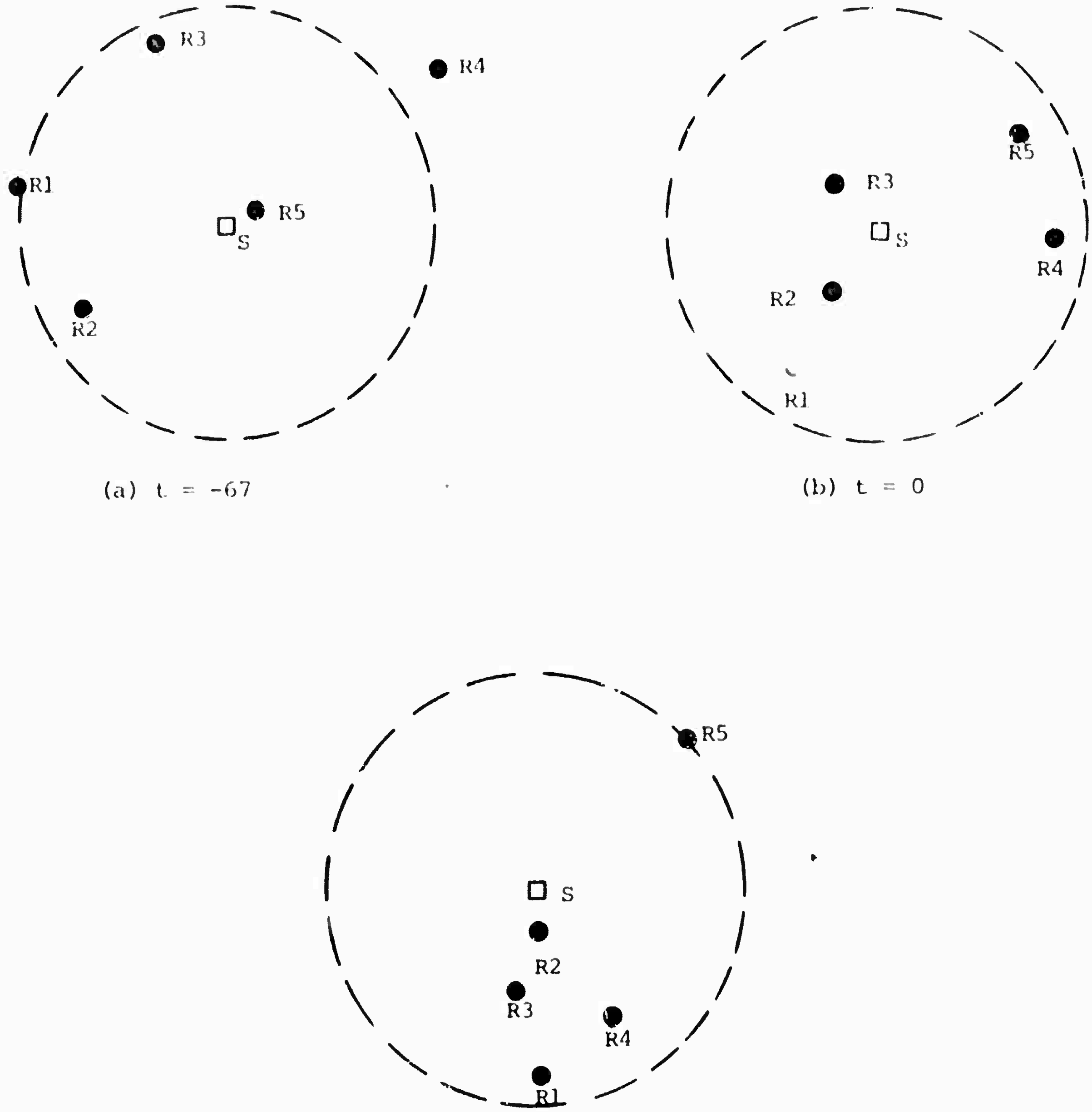


FIGURE 2: TRAJECTORIES FOLLOWED BY REPEATERS



(a) t = -67

(b) t = 0

(c) t = 47

FIGURE 3: STATUS OF THE SYSTEM AT THREE SAMPLE POINTS

	(1)	(2)	(3)	(4b)
r_1	(-65.23 65.23)	(-93.14, 151.95)	(-67.21, 47.12)	1: $(-\infty, -67.21) \cup (47.21, \infty)$
r_2	(-181.91, 316.08)	(-202.78, 316.08		2: $(-\infty, 93.14) \cup (151.95, \infty)$
r_3	(-94.03, 152.86)	(-94.03, 152.86)		3: $(-\infty, 94.03) \cup (152.86, \infty)$
r_4	(-57.54, 11.21)	(-67.21, 152.86)		4: $(-\infty, -202.78) \cup (152.86, \infty)$
r_5	(-202.78, 47.21)	(-202.78, 47.21)		5: $(-\infty, -202.78) \cup (316.08, \infty)$

(1) Interval during which there is direct communication with S.

(2) Interval during which there is communication with S.

(3) Interval during which the network is connected.

(4) Interval during which at least k repeaters are out of range.

TABLE 2: CONNECTIVITY INFORMATION

REFERENCES

[KAHN, 1975] Kahn, R.E., "The Organization of Computer Resources Into a Packet Radio Network," *Proceedings of the National Computer Conference*, May 1975, pp. 177-186.

[FRANK, 1975] Frank, H., I. Gitman, and R.M. Van Slyke, "Packet Radio System-Network Considerations," *Proceedings of the National Computer Conference*, May 1975, pp. 217-231.

[GITMAN, 1976] Gitman, I., R.M. Van Slyke, and H. Frank, "Routing in Packet Switching Broadcast Radio Networks," *IEEE Transactions on Communications*, August 1976.

CHAPTER 4

AN APPROXIMATE ANALYTICAL MODEL FOR INITIALIZATION OF SINGLE HOP PACKET RADIO NETWORKS

CHAPTER 4

TABLE OF CONTENTS

CHAPTER 4

TABLE OF CONTENTS: FIGURES

CHAPTER 4

TABLE OF CONTENTS: TABLES

CHAPTER 4

AN APPROXIMATE ANALYTICAL MODEL FOR INITIALIZATION OF SINGLE HOP PACKET RADIO NETWORKS

1. INTRODUCTION

Among the objectives of the packet radio system are fast network deployment and the capability of communication among mobile devices [KAHN, 1975]. Combinatorial analyses [NAC, 1973] and simulation studies [FRANK, 1975] have demonstrated that the impact of the routing algorithm on network efficiency is of prime significance. That is, unlike in point-to-point packet switching networks in which a sophisticated routing algorithm (as compared to simple fixed routing) can result in an average increase in network throughput of up to 20% to 30%, in packet radio networks, the increase in throughput may be by an order of magnitude.

Efficient routing algorithms for packet radio networks have been presented in [GITMAN, 1976]. The principal idea of the hierarchical routing algorithm* is to transform the broadcast network to a "point-to-point" network for packet transportation, by assigning repeaters a code for routing, called Label. The problem is that the connectivity of devices is changing, and therefore, it is necessary to develop algorithms for dynamically changing the network structure (reconfiguration) under certain conditions. Examples of a changing topology are when the network is mobile (e.g., a Packet Radio System for a fleet of ships), when a repeater's range changes (e.g., from battery drainage) or when repeaters fail. A different type of network change occurs when terminals enter or leave the areas served by a given repeater or station.

---

* This algorithm is presented in [GITMAN, 1976] and has been implemented in the packet radio experimental system.

The initialization procedure assumes that network topology is not known a priori and is changing with time. Hence, the initialization procedure involves mapping of network topology, determining network structure (labels for repeaters), and transmitting labels to repeaters. Initialization and connectivity monitoring algorithms, which are optimum in some sense, were presented in [NAC, 1975]. In this chapter we postulate a simple initialization scheme and perform worst case analysis of initialization time as a function system parameters.

The initialization scheme analyzed assumes the following packet types:

ROP - Repeater On Packet
LLP - Label Packet

ROP's are issued by uninitialized repeaters and inform the station about the existence of the repeater and the link it records. LLP's are sent from the station to repeaters. A single station and single hop packet radio network is analyzed. Realistic models are set up at first; however, to obtain closed form solutions we must assume worst case values for parameters, obtaining in conjunction, worst case values (or upper bounds) for initialization.

The closed form solution for the initialization time is a function of number of repeaters, the retransmission time intervals of repeaters and station, and the interference (connectivity of repeaters). The study, by numerical methods, of the minimum initialization time demonstrates:

a. A good strategy for the station (in the absence of information traffic) is to transmit a label every 4 slots, on the average. This value was demonstrated to be optimum for large variations in all other parameters.

b. For low connectivity (interference) of repeaters, a repeater should transmit a ROP every m•e slots, on the average; where m is the number of repeaters.

2. MODEL

2.1 Single Hop Model Without Terminal Traffic

Figure 1 depicts the network under scrutiny.
The following assumptions are made:

1. Single station with m repeaters, physically at 1-hop.

2. No repeater accepts ROP's from another repeater.

3. Each repeater can interfere with I neighbors (in addition to itself).

4. Common channel, slotted ALOHA.

5. Protocol.

   a. Each repeater broadcasts ROP's, such that the number of slots between any two transmissions is uniformly distributed on $[0,2\Omega]$.

   b. If a label is received by the repeater, it forwards one ETE ACK to the station and it halts its ROP emission except as in (c) below.

   c. If a repeater has received a label, it forwards monitoring ROP's uniformly on $[0,2\phi]$ slots, $\phi >> \Omega$. (We shall disregard this

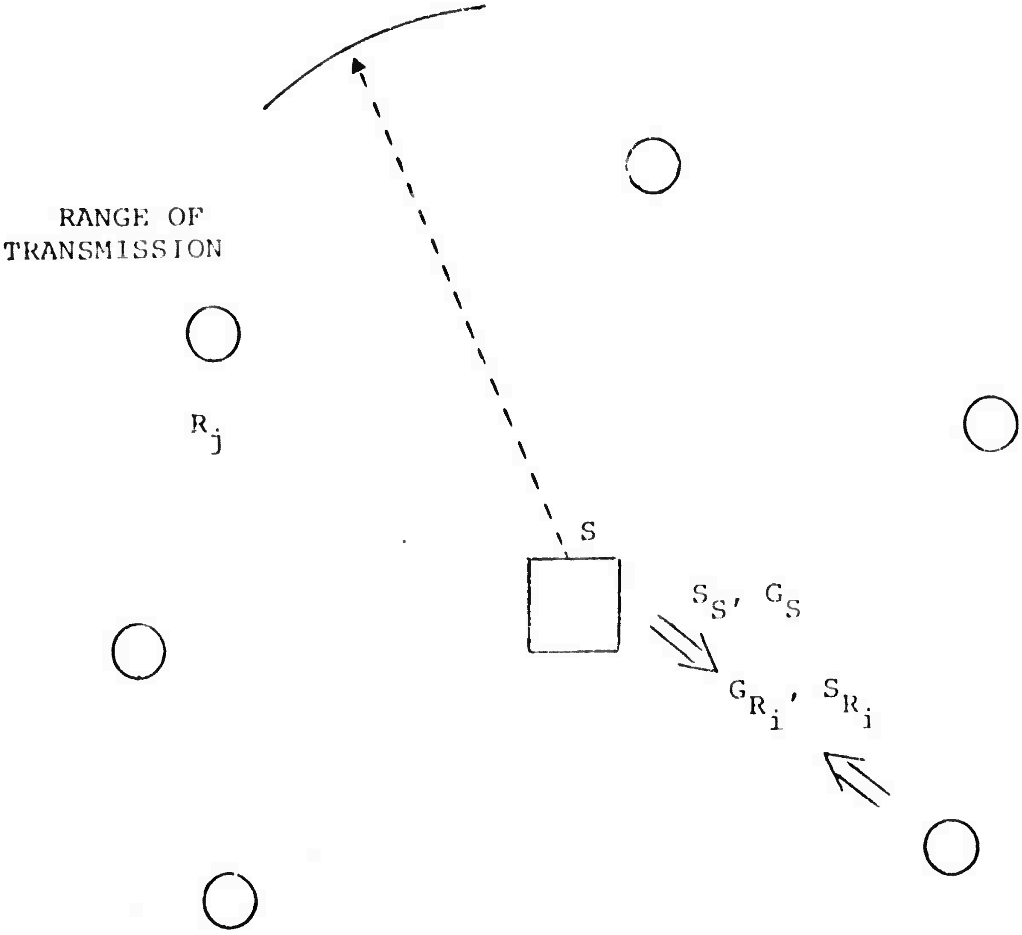


FIGURE 1: 1-HOP NETWORK

effect here, since by taking ϕ large we can in effect say that the repeater in question is essentially quiet after initialization.)

d. If the station has issued a label and it gets no ETE ACK, it retransmits the label at an appropriate time - see below.

e. The station maintains a queue of all the unacknowledged labels to be (re)transmitted.

f. Every time the station receives a ROP it checks the queue of labels. If there already is a label scheduled for transmission to the repeater in question, the station disregards this ROP; otherwise it adds the label to the queue.

g. FIFO for station queue: zero processing and propagation time; infinite buffer size at the station.

2.2 Machinery

Let R_1, R_2, ..., R_m be the repeaters in question, and S the station.

Let A_i = Time when the first ROP from R_i reaches S.

Let Q(t) = Those repeaters whose label is in the station queue at time t.

$Q(t)$ represents those repeaters such that a ROP has been received by the station by time t, but no ETE ACK has been received.

A_i is the time when the label for i enters the queue for the first time.

Let B_i = Time when the label for i is deleted from the queue (namely it is received by R_i and acknowledged).

<u>Definition</u>: By initialization time we mean:

$$\text{Ɨ} = \max_i E(B_i) \tag{1}$$

Let $0_t = |Q(t)|$ = Number of labels to be issued at time t. We call 0_t the cycle length at t.

We now expand assumption 5d:

> (5d') At the end of each cycle, say t*, S scans $Q(t^*)$; the labels found in the queue are then scheduled for transmission with an intertransmission delay distributed $U[0,2w]$.

The remainder of this chapter is dedicated to expected value computations; hence we say that, on the average, one ROP every Ω slots is issued by a non-initialized R_i, and one LLP every w slots is broadcasted by S. With this in mind, we see that after S sends a label L_i to R_i, it times out for 0_{t^*} w slots on the average before forwarding a new label to R_i.

Observe that if t_{j-1} is the expected end of the $(j-1)^{th}$ cycle then

$$t_j = t_{j-1} + 0_{t_{j-1}} w \tag{2}$$

Let q_R and q_S be the probability of successful transmission on the repeater to station hop and on the station to repeater hop, respectively.

OBJECTIVES:

1. Determine q_R, q_S as a function of m, I, Ω, w.

2. Determine Ɨ, given (1).

ANALYSIS:

The phenomenon is complicated. The flow chart of Figure 2 shows the steps involved in initialization.

Consider (for simplicity) a single R_i. It is broadcasting ROP's. When a ROP reaches S, the station tries to send a label to R_i. This may require several retransmissions by S (in the meantime R_i has issued more ROP's). When R_i finally receives a label it stops issuing the ROP's and it issues one ETE ACK. If the ETE ACK is received by S, R_i is eliminated from the queue; otherwise, S retransmits the label until it is received and acknowledged by R_i.

2.3 Determination of q_R and q_S

Let S_R, G_R be the total successful and total transmission rate, respectively, from the repeaters to the station. Let S_S, G_S be the total successful and total transmission rate, respectively, from the station to the repeaters.

Let S_{R_i}, G_{R_i} be the respective rates from R_i to S. Let S_{S,R_i}, G_{S,R_i} be the respective rates from S to R_i.

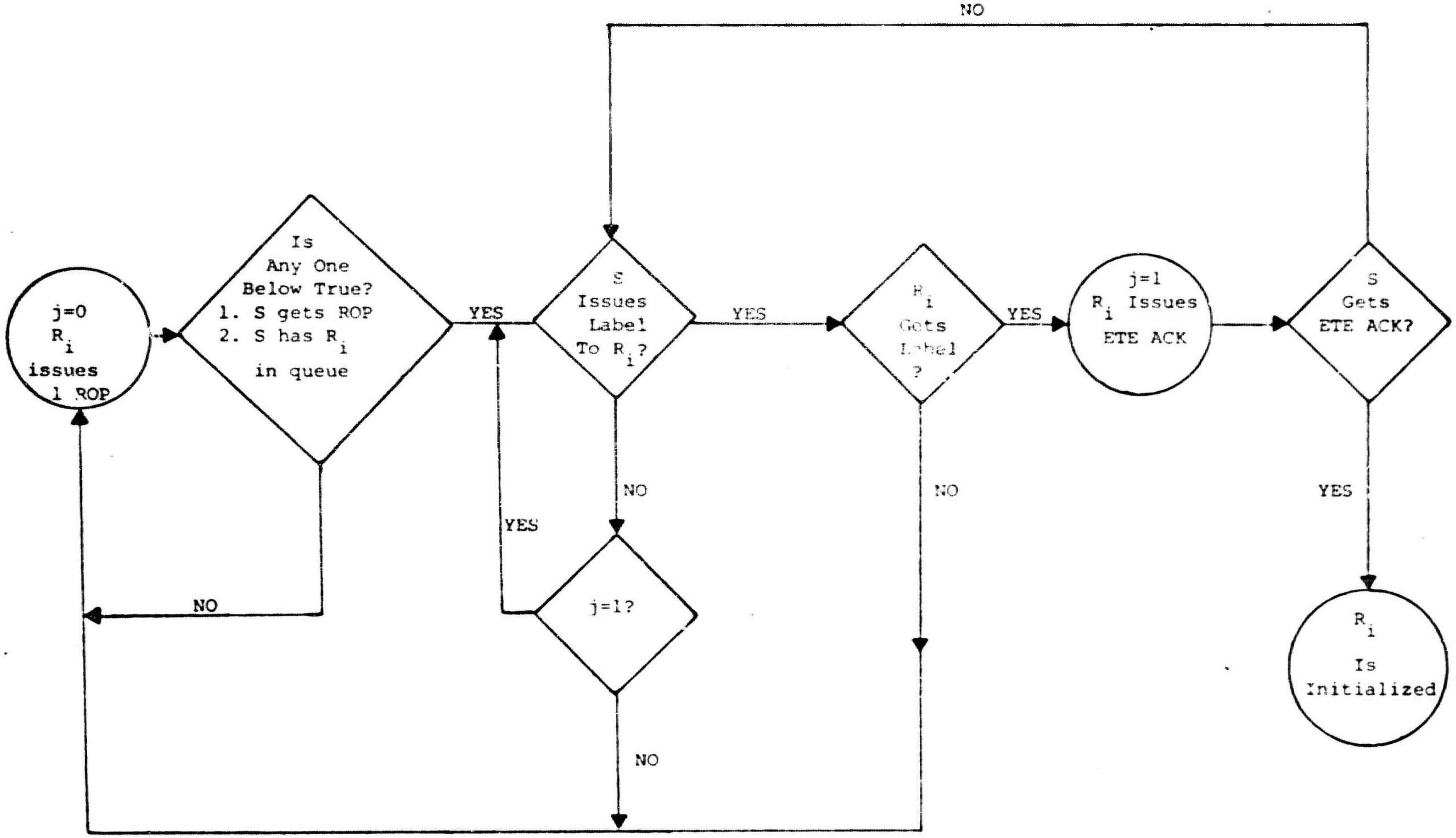


FIGURE 2: INITIALIZATION FLOW CHART

We make the following simplifying assumptions:

$$S_{R_i} = S_R/m; G_{R_i} = G_R/m; S_{S,R_i} = S_S/m; G_{S,R_i} = G_S/m \qquad (3)$$

Consider a specific slot in the ALOHA broadcasting procedure. Repeater R_i is idle if the following conditions are all satisfied.

1. The station is not broadcasting.

2. The neighbors interfering with R_i are not broadcasting.

3. R_i itself is not broadcasting.

Thus,

$$\text{Prob } (R_i \text{ is idle}) = (1-G_S) \prod_{R_j \varepsilon V_{R_i}} (1-G_{R_j}) \; (1-G_{R_i}) \qquad (4)$$

where,

$$V_{R_i} = \{R_j | d(R_i, R_j) < \mu, \mu = \text{range of repeaters}\} - R_i$$

As per assumption 3 above, $|V_{R_i}| = I, \forall_i$; also we postulated that $G_{R_i} = G_R/m$; thus

$$\text{Prob}(R_i \text{ is idle}) = (1-G_S)(1-\frac{G_R}{m})^{I+1} \qquad (5)$$

The station is idle if 1. and 2. below are satisfied:

1. The station is not broadcasting.

2. No repeater is broadcasting.

Thus,

$$\text{Prob(S is idle)} = (1-G_S)\prod_{i=1}^{m}(1-G_{R_i}) = (1-G_S)(1-\frac{G_R}{m})^m \quad (6)$$

Using the above, we can compute the expected throughput on the two links.

$$S_R = \sum_i \text{(traffic put out by } R_i\text{) Prob (S is not disturbed by any repeater except } R_i\text{, and is idle itself)}$$

$$= \sum_i \frac{G_R}{m}(1-\frac{G_R}{m})^{m-1}(1-G_S) = G_R(1-\frac{G_R}{m})^{m-1}(1-G_S) \quad (7)$$

$$S_S = \sum_i \text{(traffic put out by S to } R_i\text{) Prob (} R_i \text{ is not disturbed by any device except S)}$$

$$= \sum_i \frac{G_S}{m}(1-\frac{G_R}{m})^{I+1} = G_S(1-\frac{G_R}{m})^{I+1} \quad (8)$$

Now

$$q_S = \frac{S_S}{G_S} = (1-\frac{G_R}{m})^{I+1} \quad (9)$$

$$q_R = \frac{S_R}{G_R} = (1-\frac{G_R}{m})^{m-1}(1-G_S) \quad (10)$$

If $G_R = \rho(m)m$ and $G_R \to a$ monotonically from below as $m \to \infty$,
$G_S = \sigma(m)m$ and $G_S \to b$ monotonically from below as $m \to \infty$,
$I = \lambda m$, then

$$q_R = (1-\rho(m))^{m-1} \; (1-\rho(m)m) \qquad (11)$$

$$q_S = (1-\rho(m))^{\lambda m+1} \qquad (12)$$

The following relations hold :

a. q_R decreases as m increases.

b. $\lim\limits_{m\to\infty} q_R = e^{-a}(1-b)$

c. q_S decreases as m increases.

d. $\lim\limits_{m\to\infty} q_S = ((e^{-a})$

(on the other hand, if I is fixed then the limit is 1)

Due to the high complexity of the phenomenon, we are in a position to compute only worst case analysis.

Let $\hat{G}_S$ = worst G_S, $\hat{G}_R$ = worst G_R. The worst G_S is obtained when the station label queue is assumed non-empty, resulting:

$$\hat{G}_S = \frac{1}{w} \qquad (13)$$

To obtain $\hat{G}_R$ it is assumed that all repeaters are transmitting ROP's and that all label packets to repeaters are successful, resulting in the additional ETE ACK from repeaters. This results:

$$\hat{G}_R = \frac{m}{\Omega} + \frac{1}{w} \qquad (14)$$

Substituting these values in the expressions for G_R and G_S, we derive:

$$\hat{q}_R = \left(1-\frac{1}{\Omega}-\frac{1}{mw}\right)^{m-1}\left(1-\frac{1}{w}\right) \tag{15}$$

$$\hat{q}_S = \left(1-\frac{1}{\Omega}-\frac{1}{mw}\right)^{I+1} \tag{16}$$

Observe that the consequence of the uniform traffic assumption made above is:

$$\hat{q}_{R_i} = \hat{q}_R, \text{ for all } i$$

$$\hat{E}(A_i) = \hat{E}(A), \text{ for all } i$$

$$\hat{E}(B_i) = \hat{E}(B), \text{ for all } i$$

Naturally, not every repeater enters (leaves) the queue at the same time, but if the experiment were carried out a large number of times, on the average, R_i would enter (leave) the queue at $E(A)$ $(E(B))$.

3. INITIALIZATION COMPUTATIONS

Consider ROP transmissions by R_i with probability of success q_R. Then A_i is a random variable geometrically distributed. Thus, $1/q_R$ represents the expected number of transmissions by R_i before the ROP reaches S. Since the ROP is put out by R_i every Ω slots (on the average),

$$\hat{E}(A_i) = \frac{\Omega}{q_R} \qquad (17)$$

Consider label transmissions by S. $1/q_S$ represents the expected number of label transmissions by S. If we knew the cycle length at each retransmission, we could add these values to obtain the time necessary to deliver the label to R_i, such cycle length is unavailable, hence, we must assume the worst case of mw. Thus, $\frac{mw}{q_S}$ is the time required to deliver one label to R_i. However, it takes $\frac{1}{q_R}$ such receptions and ETE ACK transmissions before the station receives an ACK and eliminates this repeater from the queue.

Finally;

$$E(B_i) = \frac{\Omega}{q_R} + \frac{mw}{q_S}\,\frac{1}{q_R} = \hat{I} \qquad (18)$$

4. OPTIMIZATION OF INITIALIZATION TIME

It is clear that if Ω is small, the delay between ROP transmissions is small, and this would apparently speed up initialization. However, if Ω is small, the traffic generated by the repeaters is large, causing q_S and q_R to decrease and thus slow down initialization.

On the other hand, if Ω is large, the traffic is small causing q_R and q_S to increase and thus speeding initialization; however, since the wait between ROP transmissions is large, the initialization time may still increase.

The same holds true for w. It is apparent that optimal values of w and Ω must exist. From the previous section;

$$\bar{I}(\Omega,w) = \frac{1}{[1-(\frac{1}{\Omega}+\frac{1}{mw})]^{m-1}(1-\frac{1}{w})} \cdot \{\Omega + \frac{mw}{[1-(\frac{1}{\Omega}+\frac{1}{mw})]^{I+1}}\} \quad (19)$$

Close form optimization of this function of two variables is unattainable. We must thus resort to numerical techniques, in particular point-to-point search, which works well in view of the limited domain of w and Ω (w and Ω must be integers since we are in a slotted environment).

The graphs of Figure 3 through Figure 8 show the trends.

Figure 3 depicts the behavior of $\hat{\bar{I}}$ as a function of Ω and w. Basically, we see that the initialization time to the right of the optimum increases rather slowly, while to the left of the optimum increases drastically; hence, we should over estimate Ω^* rather than under estimate. Observe also that the station rate is much larger than the repeater rate; this is so since S must serve m repeaters.

Figure 4 shows the relation between Ω^* and w at $\hat{\bar{I}}^*$. It demonstrates that if the station transmits at a higher interval, so should the repeaters.

Figure 5 describes the behavior of $\hat{\tau}^*$ as a function of w, pointing at the necessity of selecting the optimum value of w.

Figure 6 depicts the behavior of $\hat{\tau}$ as a function of m. The graphs show that as m increases, the optimal initialization time increases approximately linearly with m (for fixed interference level I).

Figure 7 and Figure 8 illustrate the behavior of τ as a function of I. The graphs show that as I increases, the optimal initialization time increases approximately linearly with I (for fixed m).

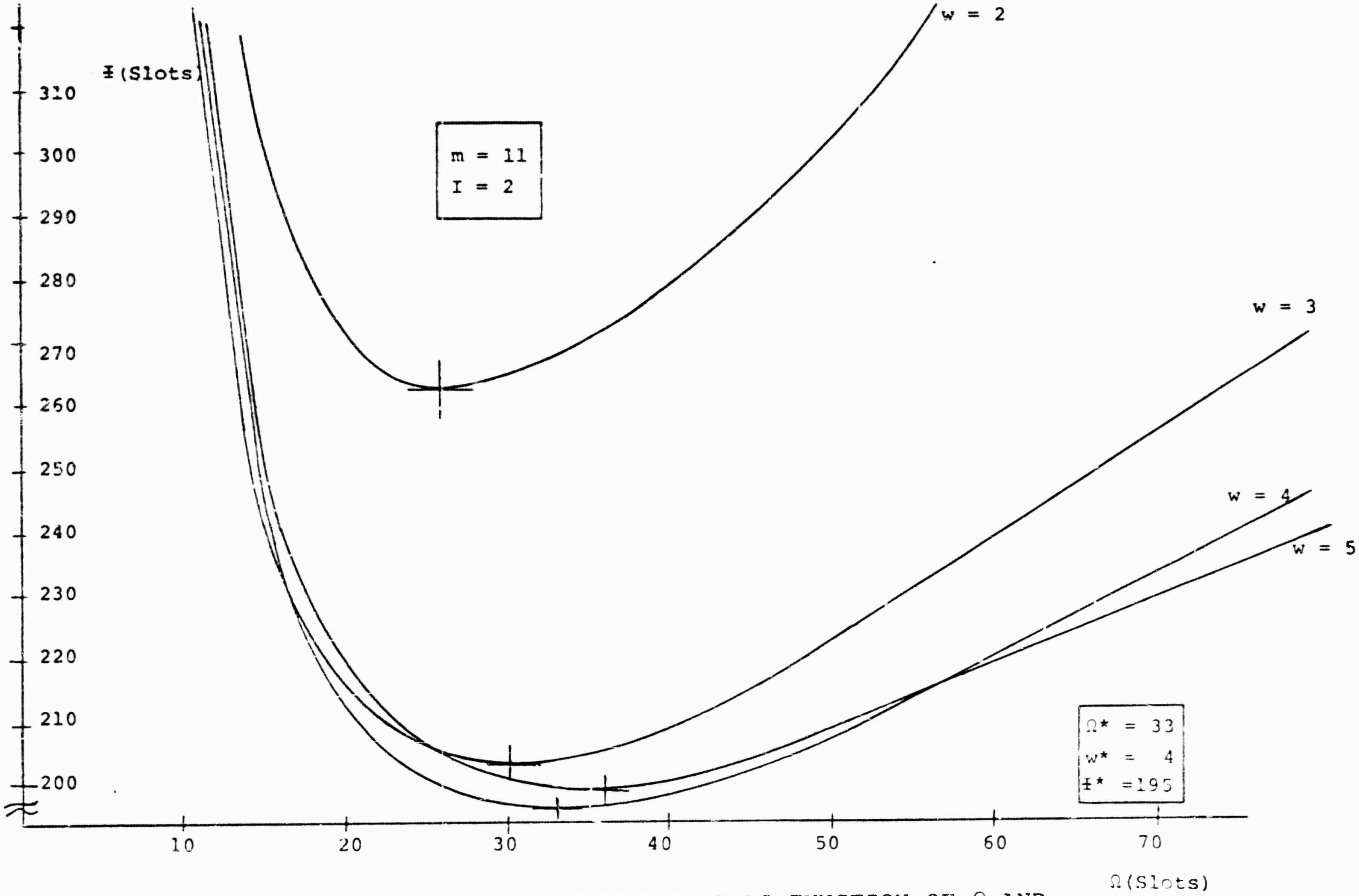


FIGURE 3: INITIALIZATION TIME AS FUNCTION OF Ω AND w

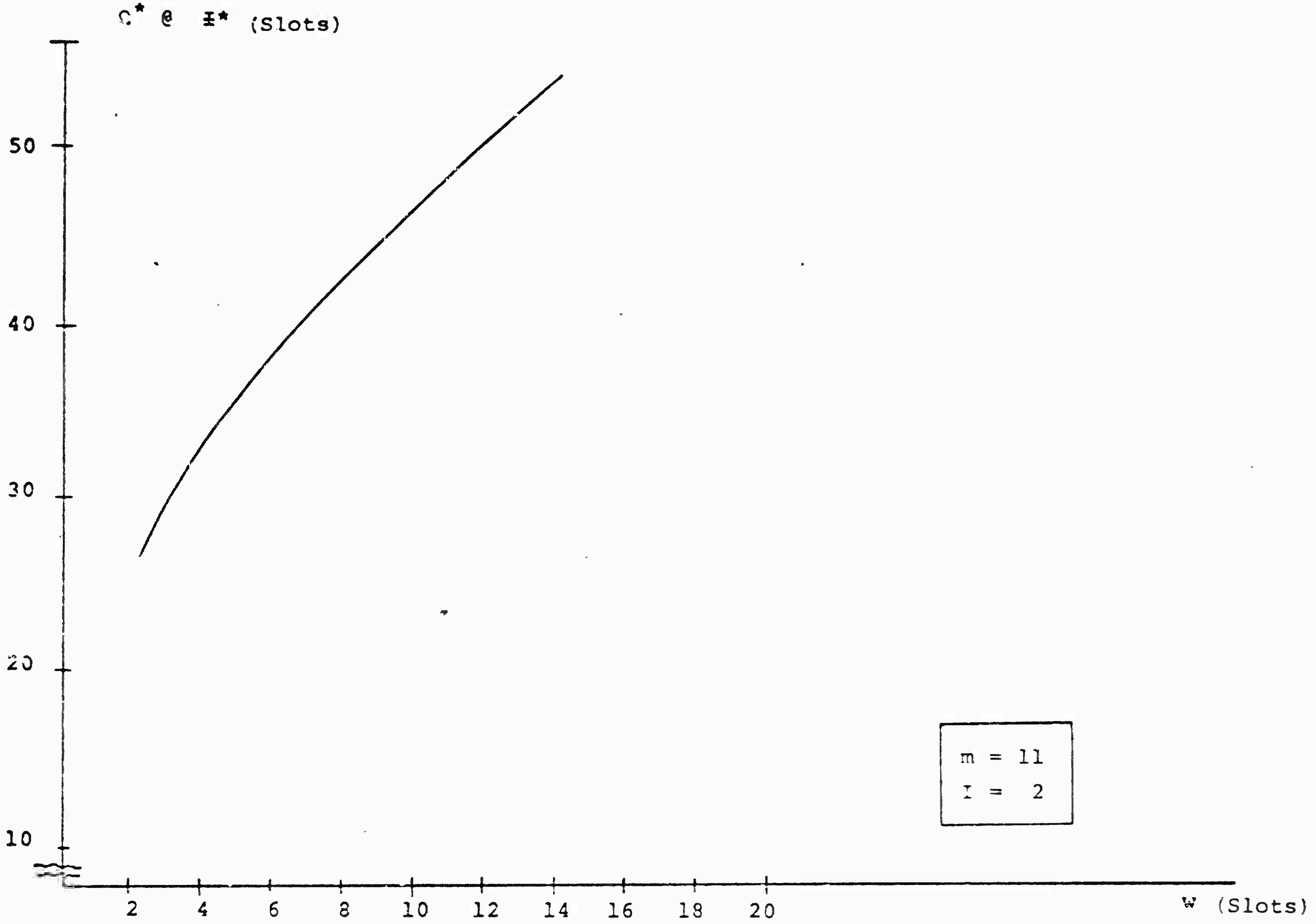


FIGURE 4: RELATION OF w AND Ω*, AT THE OPTIMUM INITIALIZATION TIME

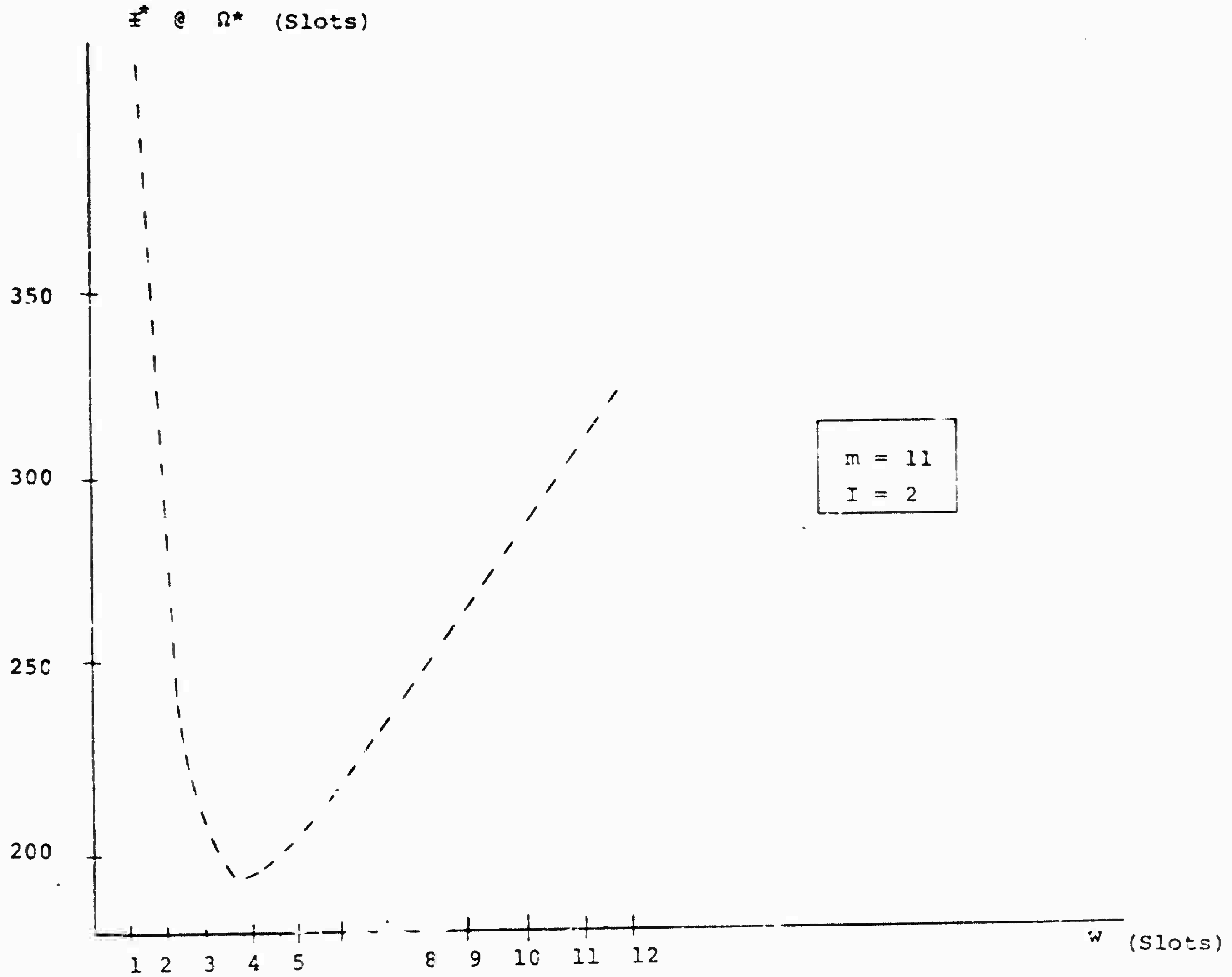


FIGURE 5: RELATION BETWEEN w AND ± AT THE OPTIMUM REPEATER RATE

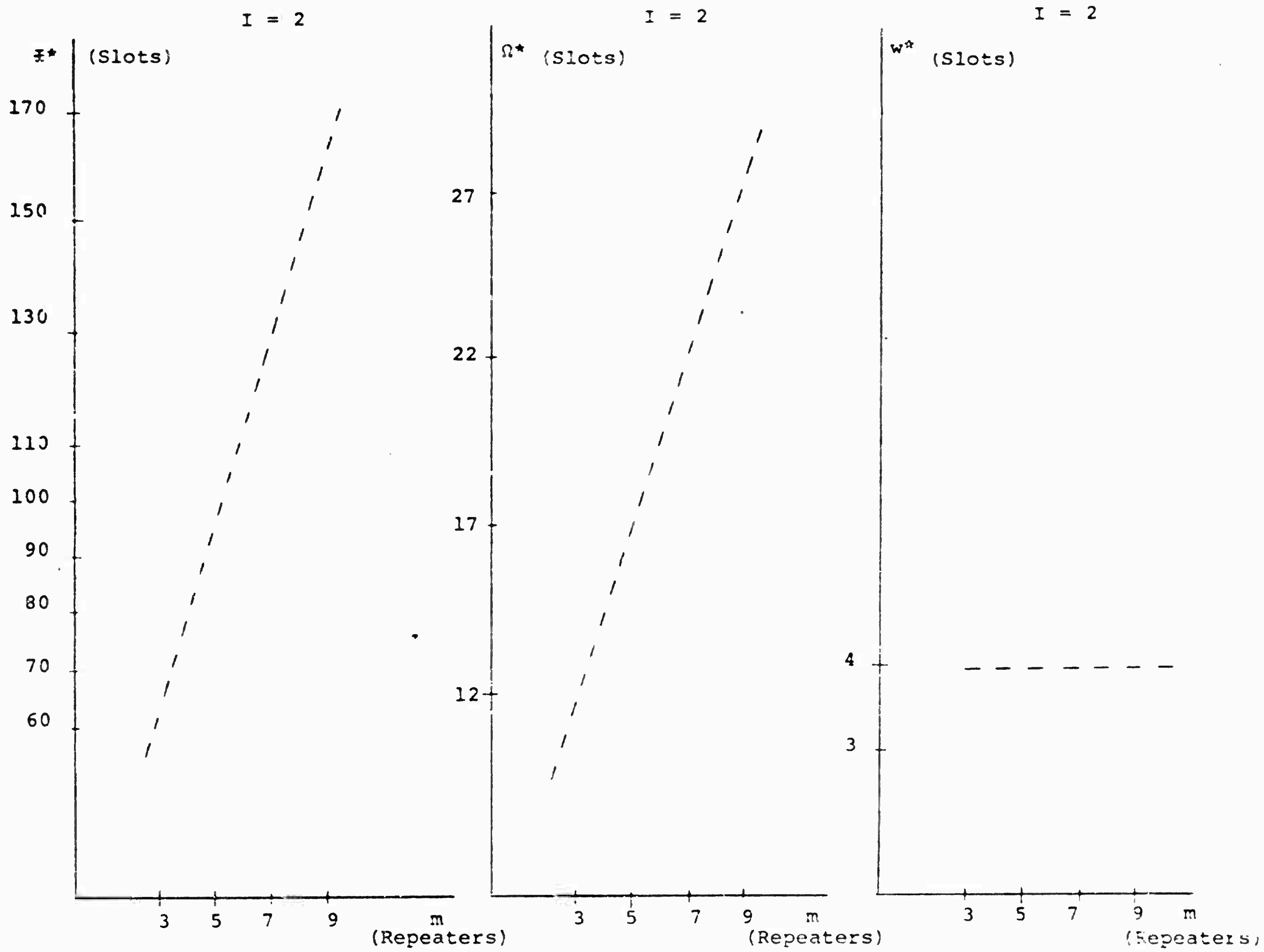


FIGURE 6: RELATION BETWEEN Ŧ*, Ω*, w* AND m

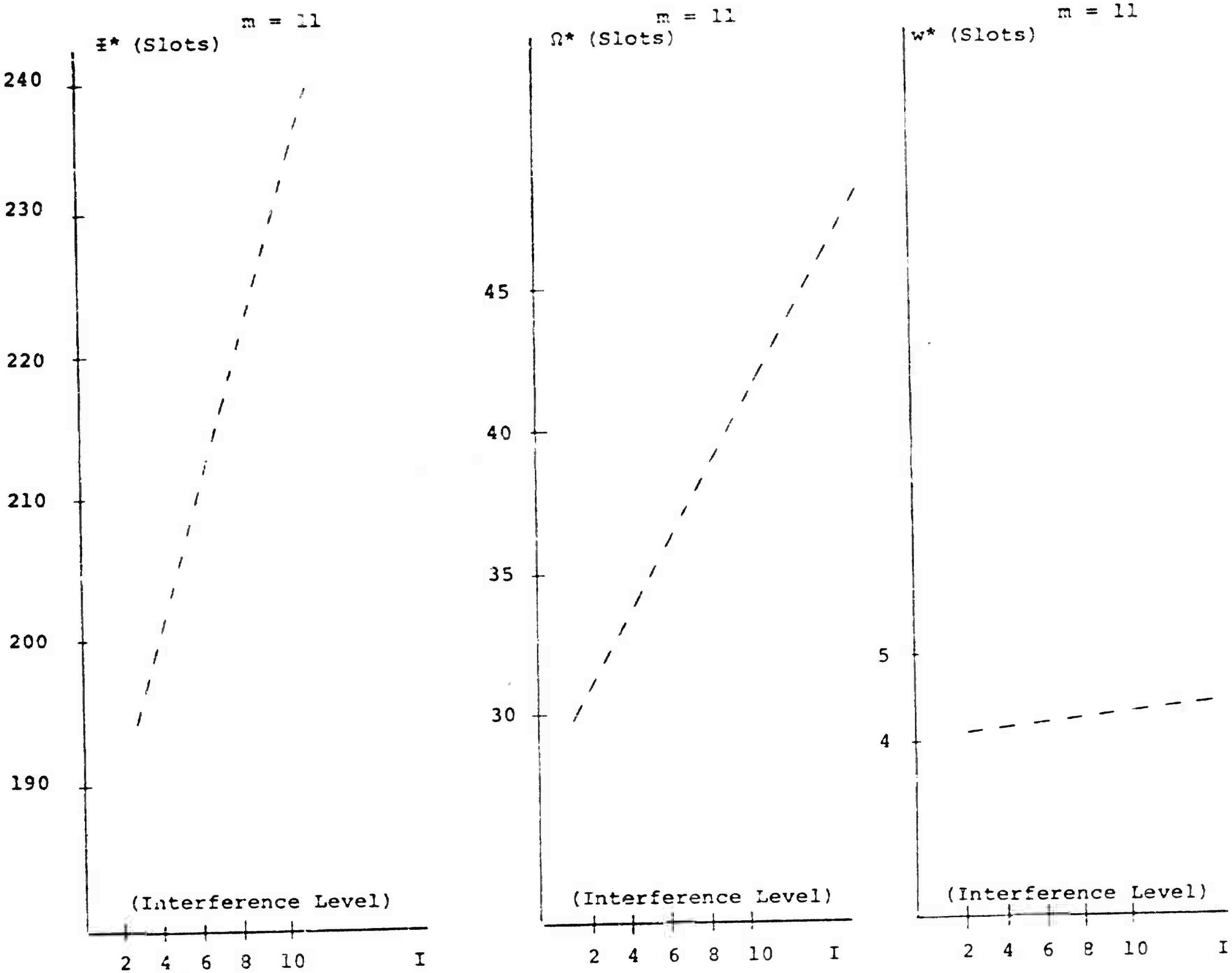


FIGURE 7: RELATION BETWEEN $Ɨ^*$, Ω^*, w^* AND I

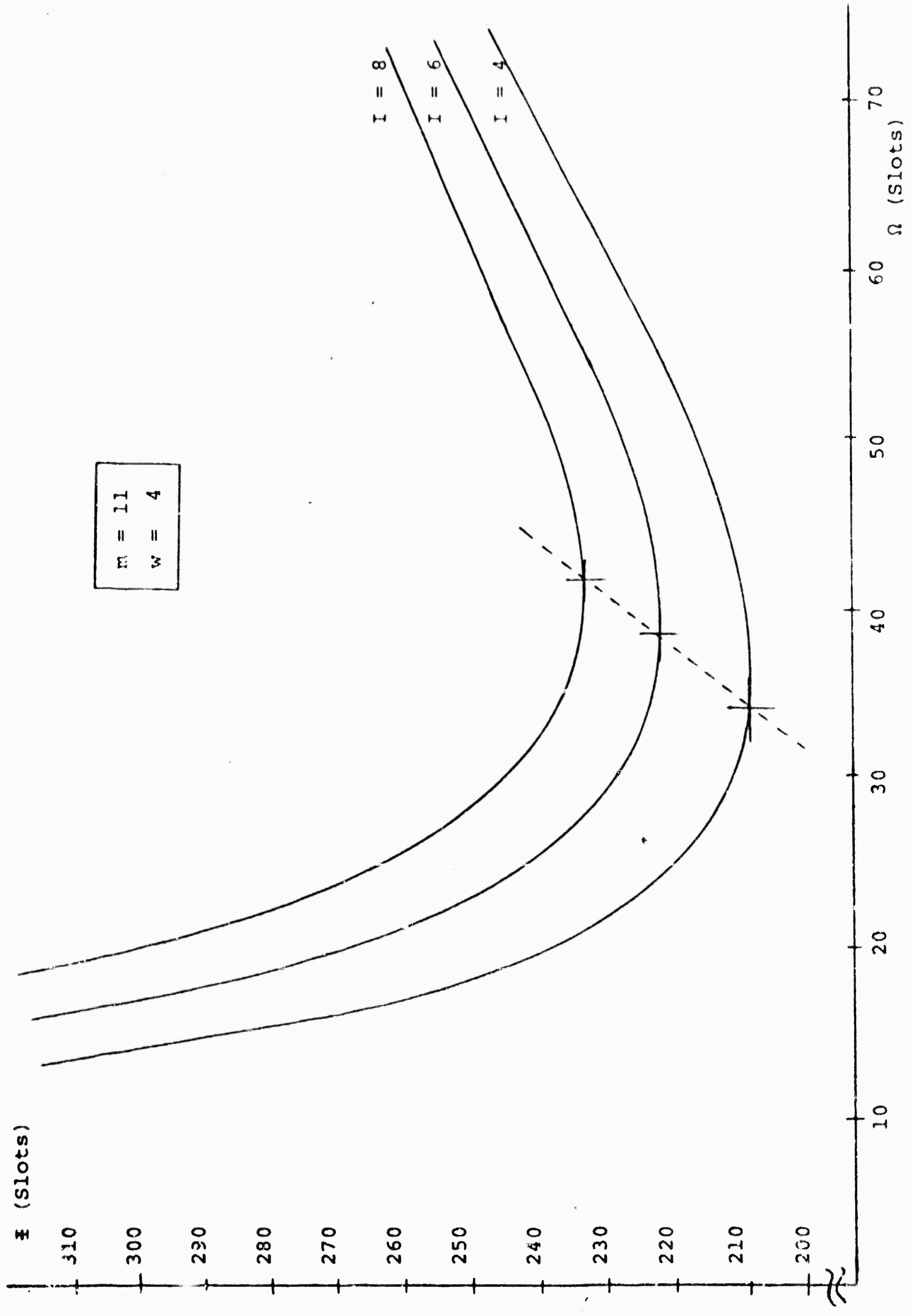


FIGURE 8: RELATION BETWEEN ∓ AND Ω AS A FUNCTION OF I

5. APPROXIMATE OPTIMAL VALUES

In lieu of a close form solution for the optimal values, we can carry out a satisfactory empirical evaluation of the optima.

Part A of Table 1 gives the optima for various values of m and I. Strikingly, $\hat{w}^*$ is constant over a large range of m and I. In fact $\hat{w}^* = 4$ in all cases, except when I is very close to m, in which case $\hat{w}^* = 5$. We thus conclude that:

> $\hat{w}^*$ = optimal station rate, is nearly independent of m and I (except in case of extreme interference) and $w^* = 4$ slots.

Part B of the table gives the values for the quantity

$$h = m(1 + \frac{1}{m})^{m+I/2+1} \tag{20}$$

We see that $\hat{\Omega}^*$ is very close to this expression, we thus declare that

$$\hat{\Omega}^* \simeq m(1+\frac{1}{m})^{m+I/2+1} \tag{21}$$

Observe I = 2,

$$\hat{\Omega}^* \simeq me \tag{22}$$

where e = 2.718 ... (more generally, this holds also for the case where I is small and independent of m).

Actually, both approximations for $\hat{\Omega}^*$ tend to overstate the value of the optimum, as a comparison of Column B in Table 1, with column (d) reveals. We conclude that:

A					B	C	D
(a) m	(b) I	(c) w^*	(d) Ω^*	(e) $\ddagger^*$	Computed Ω^* (rounded)	Computed $\ddagger^*$ (rounded)	Approximation
3	2	4	12	63	12	75	(2)
5	2	4	17	95	18	106	
7	2	4	22	128	23	138	
9	2	4	27	161	28	171	
11	2	4	33	195	34	204	
50	2	4	133	842	140	858	
100	2	4	260	1672	275	1698	
1000	2	4	2566	16610	2722	16818	
10000	2	4	24626	165944	27186	168018	
100	10	4	273	1724	287	1750	(1)
100	50	4	337	1986	350	2049	
100	90	5	429	2250	427	2431	
11	4	4	36	208	37	219	(1)
11	6	4	39	221	40	237	
11	8	4	42	235	44	256	
11	10	5	49	247	48	278	

TABLE 1: NUMERICAL COMPARISON OF EXACT SOLUTION TO APPROXIMATE MODEL AND APPROXIMATE SOLUTION TO APPROXIMATE MODEL

$\hat{\Omega}^*$ = optimal repeater rate, is an increasing function of m and I, and for small values of I, $\Omega^* \simeq me$.

Figure 9 compares the approximate optimal values of the approximate solution with the exact optimal values of the approximate solution using the results of Table 1. Observe the close fit for a large range of m.

With the above approximations, we can compute the worst case, approximate initialization time, as follows:

$$\hat{q}_R = (1 - (\frac{1}{\Omega}+\frac{1}{mw}))^{m-1}(1-\frac{1}{w}) \qquad (23)$$

$$\hat{q}_R^* \simeq [1 - (\frac{1}{me}+\frac{1}{4m})]^{m-1}(1-\frac{1}{4})$$

$$= \frac{3}{4}[1 - \frac{4+e}{4em}]^{m-1} = \frac{3}{4}[1 - \frac{(4+e)/4e}{m}]^{m-1}$$

$$\simeq \frac{3}{4}\, e^{(4+e)/4e} \simeq .40 \qquad (24)$$

Similarly,

$$q_S^* \simeq [1 - (\frac{1}{me}+\frac{1}{4m})]^{I+1} = [1 - \frac{(4+e)/4e}{m}]^{m\frac{I+1}{m}}$$

$$\simeq (e^{(4+e)/4e})^{(I+1)/m} \simeq (.54)^{(I+1)/m} \qquad (25)$$

Finally,

$$\hat{I}^* = \frac{\Omega}{q_R} + \frac{mw}{q_R q_S} \simeq \frac{me}{.40} + \frac{4m}{.4(.54)^{(I+1)/m}}$$

$$= 6.79m + \frac{10m}{(.54)^{(I+1)/m}} \qquad (26)$$

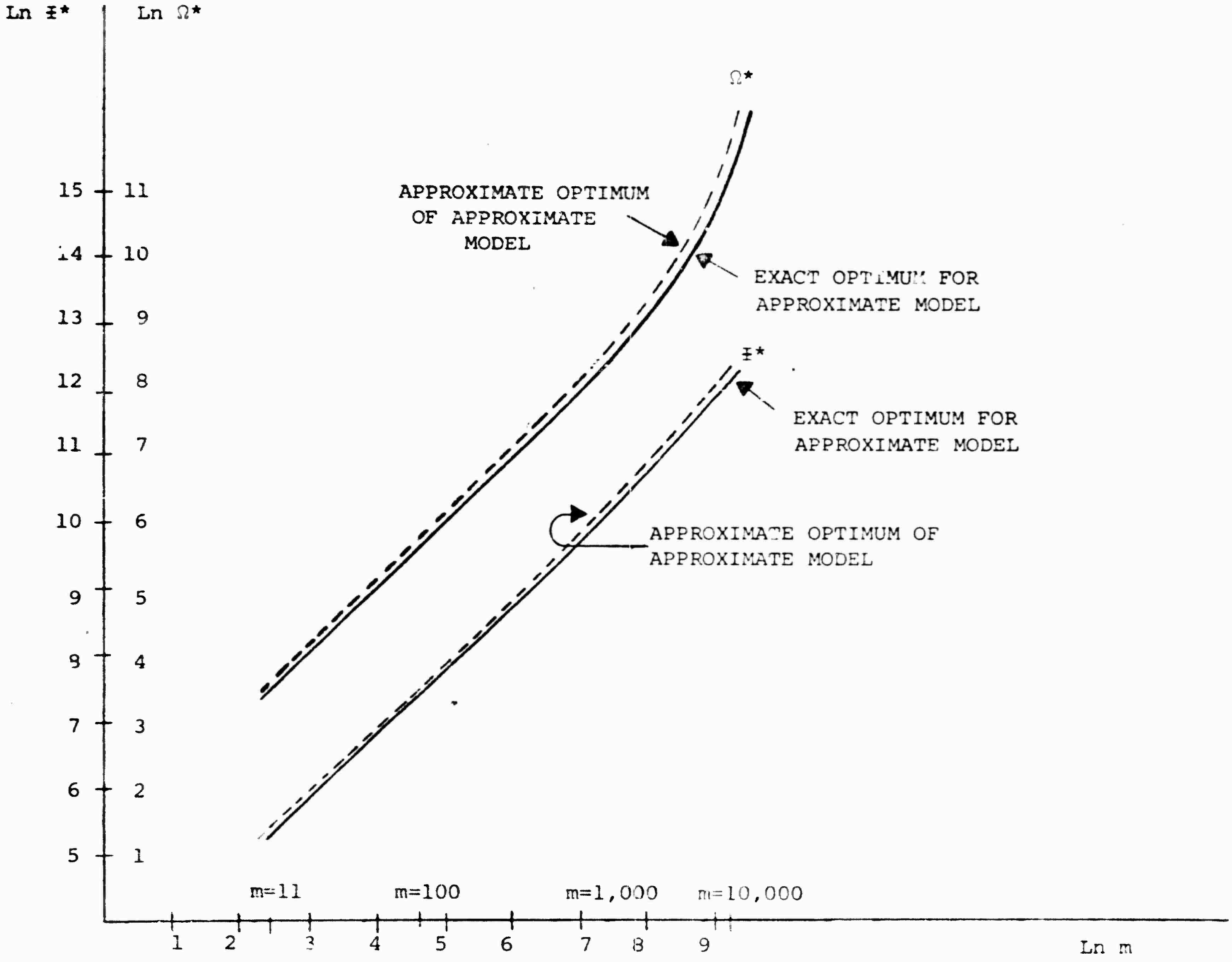


FIGURE 9: COMPARISON OF EXACT SOLUTION TO APPROXIMATE MODEL AND APPROXIMATE SOLUTION TO APPROXIMATE MODEL

Part C of Table 1 shows for comparison the computed values of $\hat{Ɨ}^*$.

So far Ɨ has been quoted in a number of slots in the ALOHA procedure. Typical value for slot length for an ROP is 5×10^{-3} seconds. Hence, for example, in a 100 repeater system at interference level 2, it would take 0.84 seconds to initialize a typical repeater.

6. CONCLUSION

In this chapter a worst case study for initialization has been conducted; it is believed that the same trends with respect to the dependency of Ɨ on I and n_r, etc., hold true in the actual initialization procedure. To ascertain this an exact model via a Markov chain is being developed, and will be compared to the model described presently.

Other approximate models (multiple retransmissions, finite buffer size, 2-hops) are under investigation.

REFERENCES

[FRANK, 1975]	Frank, H., I. Gitman, and R. Van Slyke, "Packet Radio System - Network Considerations," National Computer Conference, May 1975, pp. 217-231.
[GITMAN, 1976]	Gitman, I., R. M. Van Slyke, and H. Frank, "Routing in Packet Switching Broadcast Radio Networks," to be published in the IEEE Transactions on Communications, August 1976.
[KAHN, 1975]	Kahn, R. E., "The Organization of Computer Resources into a Packet Radio Network," National Computer Conference, May 1975, pp. 177-186.
[NAC, 1973]	Network Analysis Corporation, The Practical Impact of Recent Computer Advances on The Analysis and Design of Large Scale Networks, Second Semiannual Technical Report, December 1973, available from Defense Communication Center, Arlington, Virginia.
[NAC, 1975]	Network Analysis Corporation, Local, Regional and Large Scale Integrated Networks, Fifth Semiannual Technical Report, August 1975, available from Defense Communication Center, Arlington, Virginia.

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
None

3. REPORT TITLE
SEVENTH SEMIANNUAL TECHNICAL REPORT, AUGUST 1976, FOR THE PROJECT "INTEGRATED DOD VOICE AND DATA NETWORKS AND GROUND PACKET RADIO TECHNOLOGY"

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
SEVENTH SEMIANNUAL REPORT, AUGUST 1976

5. AUTHOR(S) (Last name, first name, initial)
NETWORK ANALYSIS CORPORATION

6. REPORT DATE
AUGUST 1976

7a. TOTAL NO. OF PAGES
192

7b. NO. OF REFS
28

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

b. PROJECT NO.
ARPA ORDER NO. 2286

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
SEMIANNUAL REPORT 7

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

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES
None

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

13. ABSTRACT
New research results on the following major questions are reported: Methodology and algorithms for mixed terrestrial-satellite network design are developed. Simple cost functions for mixed network design are proposed and demonstrated to be good estimates. The algorithms are applied to AUTODIN II data and sensitivity to variations in cost, traffic level, and other design parameters are presented. A procedure for minimum cost design and for analysis of non-hierarchical circuit switched networks is presented. The problem of connectivity in mobile packet radio networks is solved and implications for predictable networks outlined. Approximate analytical models for initialization of packet radio networks are presented, initialization time is estimated, and operating parameters to obtain minimum initialization time are derived.

14. KEY WORDS
Computer networks, communication networks, terrestrial and satellite networks, mobile packet radio networks, throughput, cost, delay, blocking, ARPA Computer Network, store-and-forward, packet switching, circuit switching.