General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)
State Criminal Justice Telecommunications (STACOM) Final Report

Volume IV: Network Design Software User's Guide

Prepared for

Law Enforcement Assistance Administration, Department of Justice

by

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California
State Criminal Justice Telecommunications (STACOM) Final Report

Volume IV: Network Design Software User’s Guide

Jun-Ji Lee

October 31, 1977

Prepared for
Law Enforcement Assistance Administration, Department of Justice

by
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California
The State Criminal Justice Telecommunications (STACOM) Project consists of two major study tasks. The first entails a study of criminal justice telecommunication system user requirements and system traffic requirements through the year 1985. The second investigates the least cost network alternatives to meet these specified traffic requirements.

Major documentation of the STACOM Project is organized in four volumes as follows:

<table>
<thead>
<tr>
<th>Title</th>
<th>Document No.</th>
</tr>
</thead>
</table>

The above material is also organized in an additional four volumes which provide a slightly different reader orientation as follows:

<table>
<thead>
<tr>
<th>Title</th>
<th>Document No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>State Criminal Justice Telecommunications (STACOM) Functional Requirements - State of Ohio</td>
<td>5030-43*</td>
</tr>
<tr>
<td>State Criminal Justice Telecommunications (STACOM) Functional Requirements - State of Texas</td>
<td>5030-61*</td>
</tr>
<tr>
<td>State Criminal Justice Telecommunications (STACOM) User Requirements Analysis</td>
<td>5030-80*</td>
</tr>
<tr>
<td>State Criminal Justice Telecommunications (STACOM) Network Design and Performance Analysis Techniques</td>
<td>5030-99*</td>
</tr>
</tbody>
</table>

*Jet Propulsion Laboratory internal document.
This document, No. 77-53, Volume IV, entitled, "Network Design Software Users' Guide," describes techniques that are implemented in the STACOM program. It then illustrates the application of this program by providing a run example with detailed input/output listing.

It presents the results of one phase of research carried out jointly by the Jet Propulsion Laboratory, California Institute of Technology, and the States of Texas and Ohio. The project is sponsored by the Law Enforcement Assistance Administration, Department of Justice, through the National Aeronautics and Space Administration (Contract NAS7-100).
ABSTRACT

A users' guide is provided in this volume for the network design software developed during the State Criminal Justice Telecommunications (STACOM) project sponsored by the Law Enforcement Assistance Administration (LEAA).

The network design program is written in FORTRAN V and implemented on a UNIVAC 1108 computer under the EXEC-8 operating system which enables the user to construct least-cost network topologies for criminal justice digital telecommunications networks. A complete description of program features, inputs, processing logic, and outputs is presented. Also included is a sample run and a program listing.
CONTENTS

1 INTRODUCTION --------------------------------------------- 1-1
1.1 PURPOSE AND SCOPE -------------------------------------- 1-1
1.2 SUMMARY ----------------------------------------------- 1-1
1.2.1 The STACOM Program ---------------------------------- 1-1
1.2.2 State Criminal Justice Communication Network and its Optimization -------------------------- 1-2
1.2.3 Functions Performed by the STACOM Program -------------- 1-2
1.2.4 Operational Procedure ------------------------------- 1-4
1.2.5 Aborting and Recovering a Run ------------------------ 1-5
2 THE STACOM PROGRAM -------------------------------------- 2-1
2.1 INTRODUCTION ------------------------------------------- 2-1
2.1.1 State Criminal Justice Information System ----------- 2-1
2.1.2 State Digital Communication Network ------------------ 2-1
2.1.3 A STACOM Communication Network ----------------------- 2-2
2.1.4 Communication Network Configurations ------------------ 2-3
2.1.5 Network Optimization ------------------------------- 2-3
2.1.6 The STACOM Program and its Purposes ----------------- 2-4
2.1.7 Functions Performed by the STACOM Program ----------- 2-4
2.2 MAIN FEATURES ------------------------------------------ 2-5
2.2.1 Structure --------------------------------------------- 2-7
2.2.2 Response Time Algorithm ------------------------------- 2-12
2.2.3 Flexibility ------------------------------------------ 2-13
2.2.4 Programming Language ------------------------------- 2-14
2.2.5 Operating System Requirements ------------------------ 2-14
2.2.6 Functional Limitations ------------------------------- 2-15
2.3 INPUT ----------------------------------------------- 2-15
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1</td>
<td>Data Requirements</td>
<td>2-15</td>
</tr>
<tr>
<td>2.4</td>
<td>PROCESSING LOGIC</td>
<td>2-18</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Traffic Calculation</td>
<td>2-18</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Distance Calculation</td>
<td>2-19</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Formation of Regions</td>
<td>2-21</td>
</tr>
<tr>
<td>2.4.4</td>
<td>Selection of Regional Switchers</td>
<td>2-24</td>
</tr>
<tr>
<td>2.4.5</td>
<td>Formation of Regional Star Networks</td>
<td>2-27</td>
</tr>
<tr>
<td>2.4.6</td>
<td>Optimization of Regional Networks</td>
<td>2-29</td>
</tr>
<tr>
<td>2.4.7</td>
<td>Formation of an Interregional Network</td>
<td>2-37</td>
</tr>
<tr>
<td>2.4.8</td>
<td>Optimization of an Interregional Network</td>
<td>2-37</td>
</tr>
<tr>
<td>2.5</td>
<td>OUTPUT</td>
<td>2-41</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Printer</td>
<td>2-41</td>
</tr>
<tr>
<td>2.5.2</td>
<td>CalComp Plot</td>
<td>2-43</td>
</tr>
<tr>
<td>2.6</td>
<td>SYSTEM CONFIGURATION</td>
<td>2-44</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Hardware</td>
<td>2-44</td>
</tr>
<tr>
<td>3</td>
<td>PROGRAM OPERATIONS</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1</td>
<td>INTRODUCTION</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>ENVIRONMENT</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Hardware</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Software</td>
<td>3-2</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Functional Limitations</td>
<td>3-2</td>
</tr>
<tr>
<td>3.3</td>
<td>RUN DESCRIPTION</td>
<td>3-3</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Initialization and Setup</td>
<td>3-3</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Run Options</td>
<td>3-4</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Control Instruction and Sequences</td>
<td>3-9</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Program Listing</td>
<td>3-11</td>
</tr>
<tr>
<td>3.4</td>
<td>SAMPLE RUN</td>
<td>3-11</td>
</tr>
</tbody>
</table>
REFERENCES

APPENDIXES

A	STACOM PROGRAM LISTING

B	GLOSSARY

Figures

Example of a Digital Communication Network

Basic Communication Network Configurations

Example of Initial Regional Networks and an Initial Interregional Network

Example of Optimized Regional Networks and an Optimized Interregional Network

A Tree with A as its Root

Internal Representation of the Tree in Figure 2-5

STACOM Program Structure

Flow Chart for Formation of Regions

Example of Region Network Formation and Regional Switcher Selection

Flow Chart for RSC Selection

Flow Chart for Regional Star Network Formation

Typical Star Network

Typical Multidrop Network of Optimization

Flow Chart for Subroutine ESSWIL

Relationship among K, L, KI, and M Parameters

Flow Chart for Subroutine TRYLNK
77-53, Vol. IV

<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-17</td>
<td>Flow Chart for Subroutine RSPNSE</td>
</tr>
<tr>
<td>2-18</td>
<td>Flow Chart for Intraregional Line Selection</td>
</tr>
<tr>
<td>2-19</td>
<td>Basic Topology of Line Elimination</td>
</tr>
<tr>
<td>2-20</td>
<td>Flow Chart for Interregional Network Optimization</td>
</tr>
<tr>
<td>3-1</td>
<td>CalComp Plot from the Example Run</td>
</tr>
</tbody>
</table>

**Tables**

<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Examples of Line Configurations obtained by Subroutine LINNUM</td>
</tr>
<tr>
<td>3-1</td>
<td>Formats for Input Data</td>
</tr>
<tr>
<td>3-2</td>
<td>Input Data for the Example Run</td>
</tr>
<tr>
<td>3-3</td>
<td>Printer Output from the Example Run</td>
</tr>
<tr>
<td>3-4</td>
<td>Unit 6 Printer Output from the Example Run</td>
</tr>
</tbody>
</table>
SECTION 1
INTRODUCTION

1.1 PURPOSE AND SCOPE

The STACOM (STAte Criminal Justice COMMunication) network topology program is a software tool which has been developed and utilized during the STACOM project. This Software User's Guide provides:

1. A detailed description of the program, i.e., what it does and how it does it.
2. Details of the STACOM storage structure and of its program structure so that a user can easily comprehend its capabilities and limitations.
3. Details of the options available, a functional block diagram, and a program listing with comment statements so that a user can expand/improve the program capabilities by either changing parameter values or modifying the program itself.
4. Details of a sample run stream used as a reference run for correct operation, and an input/output example, so that a user can easily operate the program as a tool for network design.

The STACOM program was developed and implemented with the FORTRAN-V programming language, which is one of several high-level languages available in the UNIVAC 1108 computer systems at the Jet Propulsion Laboratory. EXEC-8 is the operating system used in these systems. With this in mind, usage of this program in a similar UNIVAC system may require some degree of conversion effort. For a facility with computers other than the UNIVAC type, a considerable effort would be required in converting this program into one compatible with the operating system of that facility.

The balance of this document consists essentially of two parts. The first deals with the functional design portion of the STACOM topology program (Section 2); the other is concerned with the operational aspect (Section 3).

1.2 SUMMARY

1.2.1 The STACOM Program

The development of the STACOM (STAte Criminal Justice COMMunication) network topology program was performed to support the primary STACOM objective of providing the tools needed for designing and evaluating intrastate communication networks. The STACOM project goals are to:
Develop and document techniques for intrastate traffic measurement, analysis of measured data, and prediction of traffic growth.

(2) Develop and document techniques for intrastate network design, performance analysis, modeling, and simulation.

(3) Illustrate applications of network design and analysis techniques to typical existing network configurations and new or improved configurations.

(4) Develop and illustrate a methodology for establishing priorities for cost-effective expenditures to improve capabilities in deficient areas.

A task involving the development of a software package for the synthesis and analysis of alternate network topologies was undertaken.

In the following subsections, we describe a typical law enforcement communication network, what the STACOM program does, how it does it, and a general operating procedure for using the program.

1.2.2 State Criminal Justice Communication Network and its Optimization

A State law enforcement communication network is defined as a network which contains a set of system terminations connected by a set of links. Each system termination consists of one or more physical terminals or computers located at the same city, called a terminal city. The main purpose of the communication network is to provide to the terminal users rapid access to and response from the data base system, and rapid response time for intra-agency communication.

Various ways of connecting a given set of terminals may be used, depending on different requirements. Because the operating costs for a given communication network depend very much on its layout, some cost reduction is possible through an initial investment in a configuration analysis.

The activity of designing a network with the lowest costs which satisfy loading requirements, called network optimization, uses various existing techniques which provide means for such purposes.

1.2.3 Functions Performed by the STACOM Program

The STACOM program is a software tool which has been developed to design optimal networks that will achieve lower operating costs. It utilizes a modified Esau-Williams technique to search for those direct links between system terminations and a regional switching center (RSC) which may be eliminated in order to reduce operating costs without impairing system performance. The RSC provides either a switching capability, a data base center, or both.
Inputs for the STACOM program contain data such as traffic, terminal locations, and functional requirements. The network may be divided into any number of desired regions in any given program run. Each region has a Regional Switching Center (RSC) which serves terminals in its region. RSCs are, finally, interconnected to form the complete network. Upon receipt of a complete set of input data, the STACOM program first performs the formation of regions and, if needed, the selection of RSCs. The program then builds a regional network in which only system terminations in the region are connected. The program subsequently optimizes the regional network for each region requested by the user.

The formation of regions is performed by the program on the basis of attempting to arrive at near-equal amounts of traffic for all regions. After finding the farthest unassigned system termination from the system centroid (a geographical center), the program starts formation of the first region by selecting unassigned system terminations close to this system termination until the total amount of traffic for that region is greater than a certain percentage (90% in this implementation) of the average regional traffic. The average regional traffic is simply the total network traffic divided by the number of desired regions. The same process is repeated by the program in forming the rest of the regions.

The selection of an RSC is based on the minimal traffic-distance product sum. In the selection process, each system termination is chosen as a trial RSC, and the sum of traffic-distance products is then calculated. The location of the system termination which provides the minimal sum is then selected as the RSC, although the location of the RSC for a given region may also be specified by the user. The optimization process consists of two basic steps, i.e., searching for lines whose elimination yields the best cost saving, and updating the network. The two steps are repeated until no further saving is possible.

Before performing network optimization, the STACOM program constructs an initial star network in which each system termination is directly connected to the regional center. It then starts the optimization process. At the termination of this process, a multidrop network is generally developed. In a multidrop network, some lines have more than one system termination; these are called multidrop lines.

When needed, the STACOM program will continue to form an optimized interregional network, which consists of inter-connections between regional centers.

The process for interregional network optimization involves the same two steps: searching and updating. However, the searching step is primarily to find the alternate route, for diverting traffic between two regional switching centers, that provides the best saving.

Based on the data provided, a successful run of the STACOM program generates a regular printer output and, if requested, a CalComp plot. The printer output contains data such as initial regional network and optimized network costs, assignments of system terminations, etc. The CalComp plot shows the geographical connections of the optimized network detailing multidrop line connections to all of the system terminations.
1.2.4 Operational Procedure

1.2.4.1 Initialization and Setup. When the STACOM program is executed from an 80-character/line demand terminal, an alternate file, 100, to be used as a printer output file, must be defined. Otherwise, all printout data will be directed to the terminal which will produce interleaving output. The file is defined by the statement @ASG,UP 100.

In addition to the redirection of output file destination, the user must direct the punch card file to a proper unit for a CalComp plotter. As an example, the statement @SYM,P PUNCH$,G9PLTF will direct the punch card images to a CalComp plotter designated with G9PLTF.

1.2.4.2 Starting a Run.

1.2.4.2.1 Batch Mode. Following is a list of control statements required when running the STACOM program as a batch run:

- @RUN run-ID, account-no., project-ID, SUP-time, pages/cards
- @ASG,UP 100
- @SYM,P PUNCH$,plotter-ID
- @XQT file.STACOM
  (INPUT DATA)
- @BRKPT 100
- @FREE 100
- @SYM 100,,printer-ID
- @FIN

The RUN card gives the following information: designated run ID, user's account number, project-ID, expected SUP-time usage (sum of CPU time, I/O time, and control/execute request time), limited number of printer pages, and number of cards which may be generated from the run. Plotter-ID gives the logical ID of the CalComp pen plotter and file is the file which contains the absolute element of the STACOM program. Printer-ID gives the logical ID of the line printer. INPUT DATA as shown is the input data required. When all of these data items are in order and ready, the deck can be submitted to the operator for processing.

1.2.4.2.2 Demand Mode. If program execution is to be performed via a demand terminal, the user can converse interactively with the program. The user may also run the program as a batch job by having all input data prepared and added after the @XQT statement.

Under the conversational mode, the user acts as a respondent who answers the requests for data made by the program. This mode of operation provides the user with an understanding of how the program is progressing. A user can very often terminate a run before a complete set of input data is given if he has some knowledge of the progress being made. This capability can prevent the user from an unnecessary waste of time. For example, if a run encounters a system which has more oversized distance data than allowed, a message from the program will be printed out.
on the terminal. This will force the user to modify the program in order to handle the large number of oversized distance data.

1.2.4.3 Normal Termination. When a STACOM program run proceeds successfully and terminates normally, the normal file unit 6 will contain messages for each successful regional network optimization. After a normal termination, the user can direct the output file 100 to a printer device, and the CalComp plot will be generated by the designated CalComp pen plotter.

1.2.5 Aborting and Recovering a Run

When a run encounters trouble resulting from incorrect input data, the user can use the normal aborting procedure to terminate its execution if it is a demand job. A statement of $XX$ after interrupting the line communication by pressing the BREAK key, will terminate a program execution at any time. On the other hand, the EXEC-8 may abort a run when certain serious violations occur during its execution.

If a program run has been interrupted because of a system outage, no recovery of the run is possible.
2.1 INTRODUCTION

Two types of analysis are involved in designing a communication network. The first is concerned with arriving at acceptable line loadings; the second involves the achievement of optimal line configurations. The STACOM program was developed to accomplish both of these types of analysis.

Before describing the STACOM program itself, a State criminal justice information system with its communication network is examined as a typical existing communication network. The goal of the STACOM program is then discussed.

2.1.1 State Criminal Justice Information System

An information system is usually developed to provide a systematic exchange of information between a group of organizations. The information system is used to accept (as inputs), store (in files or a data base), and display (as outputs) strings of symbols that are grouped in various ways. While an information system may exist without a digital computer, we will consider only systems which contain digital computers as integral parts.

Information systems can be classified in various ways for various purposes. If classification is by the type of service rendered, the type of information system which serves a criminal justice community within a State can be considered as an information storage and retrieval system. This type of information system is the subject of our interest. For example, the State of Ohio has an information system with a data base located at Columbus. The data base contains records on wanted persons, stolen vehicles, and stolen license plates. Also included in the same computer are files of the Bureau of Motor Vehicles (BMV) which contain records on all licensed drivers and motor vehicles in that State.

2.1.2 State Digital Communication Network

For a given State information system, the storage and retrieval of data to/from the data base can be accomplished in various ways for different user requirements. In general, the users of a State criminal justice information system are geographically distant from the central data base computer. Because a fast turn-around time is a necessity for this particular user community, direct in-line access to the central data base by each criminal justice agency constitutes the most important of the user's requirements. In addition, it is required to move message data quickly from one agency to another at a different location. These goals require the establishment of a data communication network. Because the computer deals only with digital data, only digital data communication networks are considered here.
A digital communication network consists mainly of a set of nodes connected by a set of links. The nodes may be computers, terminals, or other types of communication control units that are placed in various locations, and the links are the communication channels providing data paths between the nodes. These channels are usually private or switched lines that are leased from a common carrier. A simple example of a network is given in Figure 2-1, where the links between modems are communication lines leased from a common carrier. The communication control unit in city E is used to multiplex or concentrate several low-speed terminals onto a high-speed line. The line which connects cities C, D, and others is called a multidrop line, and this line connects several terminals to the data base computer.

2.1.3 A STACOM Communication Network

For the purposes of the STACOM study, a communication network was defined as a set of system terminations connected by a set of links. Each system termination consists of one or more physical terminals or computers located at the same city.

Figure 2-1. Example of a Digital Communication Network
2.1.4 Communication Network Configurations

The communication network for an information system with a central data base computer is one of three basic network configurations: the star, the multidrop, or distributed connection. These three types are shown in Figure 2-2.

As shown in Figure 2-2, the star network consists of four direct connections, one for each system termination. Each connection is called a central link. The multidrop network has one line with two system terminations and two central links. In the distributed network shown, more than one path exists between each individual system termination and the central data base.

2.1.5 Network Optimization

Given a communication network, the operating costs for the various types of lines or common carrier facilities required are governed by tariffs based upon location, circuit length, and type of line. Experience suggests that the operating cost of a network can often be substantially reduced by an initial investment in a configuration analysis. In other words, some efforts in network optimization generally provide cost-saving.

*Figure 2-2. Basic Communication Network Configurations*
There are two ways of constructing a communication network in a geometrical sense. One can divide a communication system into several regions, construct an optimal regional communication network for each region, and then build an inter-regional network connecting all of the regional centers to the central data base center. Each regional center is responsible for switching messages issued from and returned to each system termination in the region. Alternatively, one can consider the whole system as a region which is entirely made up of system terminations, and perform the optimization for that region.

2.1.6 The STACOM Program and its Purposes

One of the objectives in the STACOM study is to design optimal and effective communication networks which will satisfy predicted future traffic loads for both selected model states, Ohio and Texas. In order to achieve this objective, the STACOM program was developed and utilized for the analysis and synthesis of alternative network topologies. It is also the project's goal that the final product be a portable software package which can be used as a network design tool by any user.

In network design, two major problems are the selection of a cost-effective line configuration for given traffic, and the design of an optimal network to arrive at lower operating costs.

The goal of the STACOM program is to provide a user with a systematic method for solving both problems. In other words, the main purpose of the STACOM program is to provide the network designer with a tool which he can use for line selection and for obtaining optimal line connections.

2.1.7 Functions Performed by the STACOM Program

The STACOM program can be used to generate an optimal network configuration for a communication system if traffic to/from each system termination is provided. In addition to performing the normal input/output functions, the program will:

1. Define regions, based on equal traffic distribution.
2. Select regional centers, based on minimal traffic-distance product sum.
3. Form a regional star network with the selected regional center as the regional switching center (RSC).
4. Perform regional network optimization.
5. Form an optimized inter-regional network if required.

In performing initial network formation and subsequent optimization, line selection is done by the STACOM program to satisfy the following conditions:
(1) The line utilization factor does not exceed a specific number

(2) The average terminal-response time is less than a preselected unit of time

(3) The number of terminals on a multidrop line is less than a preselected number.

In the process of regional network optimization, the STACOM program utilizes a modified Esau-Williams method (Reference 1). Starting with a star network, in which each system termination has a central link to the regional center, the optimization process searches for a central link, the elimination of which will provide the best savings in cost; the program then provides an alternate route for the traffic that would have been carried by the link eliminated. The process is repeated until no further cost saving is possible. The result of this process is a multidrop network.

When a communication system has more than two regions, the STACOM program can also be used to generate an optimal inter-regional network. It first constructs an initial inter-regional network in which every Regional Switching Center (RSC) has a direct link to every other RSC, it then performs line elimination by diverting traffic through other routes.

Figure 2-3 gives examples of regional star networks and an initial inter-regional network; Figure 2-4 gives examples of optimized regional networks and inter-regional network obtained from Figure 2-3.

2.2 MAIN FEATURES

As described in Paragraph 2.1, the STACOM program has been developed for the purpose of performing analysis and synthesis of alternative network topologies. The following is a list of features which characterize the STACOM program:

(1) The Esau-Williams routine has been modified, tested and utilized for determining near optimal network topology.

(2) A tree type structure is used as the storage structure in the program.

(3) The program execution has been made flexible; for example, constraint on response time for a multidrop line is now an input parameter.

(4) A response-time algorithm has been implemented in the program.

(5) A CalComp plotting routine has been included for drawing resulting multidropped networks.
In the rest of this subsection, these main features are discussed in detail.

2.2.1 Structure

2.2.1.1 Storage. Since a multidrop network can be viewed as a tree composed of sub-trees, it was determined that a tree-type data structure would be appropriate and convenient for representing a multidrop network.

---

**Figure 2-3. Example of Initial Regional Networks and an Initial Interregional Network**
A tree-type storage structure is therefore needed in the program. This tree-type storage structure is implemented by defining a set of storage cells.

Each system termination (data) is represented internally by a storage cell in the program. Each cell consists of five fields and each field occupies one word (i.e., a 36-bit word for UNIVAC 1108 computers).

Figure 2-4. Example of Optimized Regional Networks and an Optimized Interregional Network
Defining that system termination \( X \) is a successor of \( Y \) and \( Y \) a predecessor of \( X \) if \( X \) branches out from \( Y \), and \( X \) is the root of a tree if it has no predecessor before it, then the basic storage cell for system termination \( A \) can be described as follows:

\[
\begin{array}{c|c|c|c|c|c}
\hline
& f_1 & f_2 & f_3 & f_4 & f_5 \\
\hline
A & & & & & \\
\hline
\end{array}
\]

Let \( c(f_i) \) = content of \( i \)-th field in a storage cell \( I_A \), where \( I_A \) is an internal index for a system termination \( A \) (data), then

\[
\begin{align*}
& c(f_1) = \text{the number of system terminations under } A \\
& c(f_2) = \text{a pointer which points to the first successor of } A \\
& c(f_3) = \text{a pointer which points to the next system termination whose predecessor is the same as } A' \text{'s} \\
& c(f_4) = \text{a pointer which points back to the previous system termination whose predecessor is the same as } A' \text{'s} \\
& c(f_5) = \text{a pointer which points to } A' \text{'s predecessor}
\end{align*}
\]

When there is a 'zero' in a field, this indicates there is no one relating to \( A \) under that specific relationship. Given a tree as Figure 2-5, \( A \) is root of the tree; it has 4 successors, i.e., \( B, C, D, \) and \( E \). Figure 2-6 is the internal representation of that relationship among indices \( I_A, I_B, I_C, I_D, \) and \( I_E \) which are internal cardinal numbers for system terminations \( A, B, C, D, \) and \( E \).

The first field of storage cell \( I_A \) indicates that there are four system terminations under \( I_A \); the pointer to \( I_B \) says that \( I_B \) is its first successor. Since \( I_A \) is the root of the tree, the other three fields are left with zeros.

![Figure 2-5. A Tree with A as its Root](image-url)
In the case of IC, ID is the next successor of IA and the previous successor of IA is IB. Its third field has a pointer pointing to ID, and its fourth field a pointer pointing to IB.

2.2.1.2 Program. The STACOM program consists of twelve functionally independent routines. Figure 2-7 shows the basic structure of the program. The functional interrelationship is indicated by arrows.

An arrow from routine A to routine B indicates that routine B will be called upon by routine A during its execution. All of these routines communicate to each other through the COMMON block in addition to the normal subroutine arguments.

Major functions of eleven of these routines are given below. RSPNSE Routine is described in the following paragraph.

(1) MAIN Routine

This is the master routine of the STACOM program. In its execution, it reads in all the data required from an input device (card reader or demand terminal) and performs calculations of distances between any two system terminations. It assigns system terminations to regions, and, if necessary, selects the regional switching center by finding the system termination in the region with the minimal traffic-distance product sum. It calls upon routine RGNNET to build a star network and then performs network optimization, if required, for each of these regions.
It also performs the construction of an inter-regional network and its optimization by calling subroutine IRNOP.

In addition to these processings, the MAIN routine also prints out distance matrix, traffic matrix, and lists of system terminations by region.

(2) RGNNET Routine

This routine is called upon only by the MAIN routine. Its main functions are the formation and optimization of regional star networks. During the formation of a regional star network, each system termination is linked directly to the designated or selected Regional Switching Center (RSC) by assigning the RSC index to the last field of each associated storage cell. Tree relationships are built among system terminations by assigning pointers to the third and fourth fields of each storage cell. The resulting star network is then printed on the printer.
The optimization process utilizes the Esau-Williams algorithm (Reference 1) with some modifications. It consists of two steps: searching for a central link (a direct link from a system termination to RSC) with best cost savings under constraints (such as response-time requirement), and subsequent network updating. This network optimization process is executed only upon request. When no further cost improvement is possible, this routine prints a resulting network with data such as number of system terminations and the response time, traffic, cost, etc., associated with each multidrop line. Routine PLOTPT is then called upon to plot the resulting network layout.

(3) IRNOP Routine

This routine is called upon to act by routine MAIN. It forms an interregional network and then performs its optimization. The interregional lines are assumed to be full-duplex lines. During the optimization process, no line between two RSCs can be eliminated if traffic between them cannot be handled through only one intermediate RSC. Also, each RSC requires at least two lines to other RSCs.

(4) LINNUM Routine

This routine provides an estimated line configuration required to satisfy a given traffic load and is mainly called upon by routine RGNNET. During its execution, utilization of selected lines are calculated against the given traffic load by calling RHOFUN so that effective line utilization is less than the pre-determined number.

(5) RHOFUN Routine

This routine calculates the line effective utilization for a given traffic and line configuration.

(6) ICOSTJ Routine

Given the line configuration and indices for any two system terminations, this routine calculates the installation costs and annual recurring costs for the line and other chargeable items required. In calculating line costs, it calls upon routine DIST for distance data between two given system terminations. Resulting cost data are arranged by chargeable item type.

(7) DIST Routine

This routine retrieves distance data between any two system terminations by calling routine PACK. When the
distance is greater than 510 miles, it retrieves distance data by calling routine RECOVR.

(8) PACK Routine

This routine stores or retrieves distance data between any two system terminations. It is called upon by routine MAIN for distance data depositing, and called upon by routine DIST for its retrieval. For the purpose of saving storage, distance data has been compressed, and each 36-bit word has been divided into four sub-words of 9 bits. Therefore, any distance datum with value equal to or greater than 511 is stored in another specified area; its retrieval calls upon routine RECOVR.

(9) RECOVR Routine

During distance data retrieval in the execution of the DIST routine, if the return value from routine PACK is 511, this routine will be called upon to provide the actual distance data, which is equal to or greater than 511.

(10) LINK Routine

Since the distance between any two system terminations I and J is independent of how I and J are referred to, the routine LINK provides a mechanism for preserving such an independency by mapping I and J into an absolute index.

(11) PLOTPT Routine

This routine provides instructions for plotting a given point on a CalComp plotter. Location of a point is calculated by its associated Vertical-Horizontal (V-H) coordinates (defined under Paragraph 2.4.2).

2.2.2 Response Time Algorithm -- RSPNSE Routine

There is a limit on the number of terminals which can be linked together by a multidrop line due to constraints on reliability and response time. However, it would be an oversimplification to just use a particular number as the main constraint in determining how many terminals a multidrop line can have. In reality, the response time of a given multidrop line depends on the amount of traffic, the number of terminals on the line, and very heavily, on the number of transactions to be processed in the data base computer system.

In the STACOM program, a response time algorithm is implemented in such a way that during the network optimization process it is used to accept or reject the addition of a given terminal to a multidrop line. This response time routine calculates the average response time.
on the given multidrop line, given the number of terminals and amount of peak traffic on the line. This average response time accounts for the following types of delays: the wait-for-line time and line service time for the inquiry message from a terminal to the central switcher (i.e., a switcher which either contains data bases or communicates directly with the data base computer), the computer turnaround time at the switcher, and the wait-for-line time and line service time for the returned message to the terminal. When there is an RSC between a terminal and the central switcher, the turnaround time at the RSC and the line service time between the RSC and the central switcher are counted as part of the average response time. Before its inclusion in the STACOM program, the fidelity of this algorithm was evaluated by simulation and found to be acceptable.

2.2.3 Flexibility

At the outset of the STACOM project it was anticipated that the STACOM program would be used for states with varying traffic requirements; it was decided that the resulting program should be as flexible and general as possible. With this in mind, the STACOM program has been implemented with the following features which make it flexible and thereby enhance its capabilities:

(1) Rate Structures, Line Types, and Chargeable Items

Because a State can have more than one rate structure (tariff) applicable at any one time, the STACOM program has been designed to accommodate this.

Under a specific rate structure, any combination of line types with their names, line capacities, and basic cost figures can be prescribed to the program. In addition to the line cost, any number of chargeable items associated with each line type can be prescribed to the program. For example, any combination of cost items such as service terminations, drops, modem and others can be used. Furthermore, under the Multischedule Private Line (MPL) tariffs given by AT&T for interstate communication lines, the monthly line charge between any two terminals is now a function of both the inter-city distance and the traffic densities of both terminal cities. The STACOM program has been implemented in such a way that it can take line-cost figures based on MPL tariffs or other tariffs.

(2) Region Formation, Switcher Selection, and Network Optimization

Given a set of system terminations dividing them into regions can be performed in either of the following ways: the user can pre-assign some or all of the terminations into preselected regions, alternatively, the user can let the program perform the region
formation by simply providing the system centroid. Following the formation process, the STACOM program will start selecting regional switching centers for regions without a preassigned switching center. The process of regional network formation and its optimization will then follow.

(3) Number of Terminals per Multidrop Line.

It may be desirable to set a limit on the number of terminals on a multidrop line. In its implementation, the STACOM program takes this number from the user's input data as a constraint during its optimization process.

(4) Average Terminal Response Time.

Besides the limit on the number of terminals allowed on a multidrop line, a good network design also requires a constraint on the average terminal response time on a multidrop line. The STACOM program allows a user to specify the limit on a run basis.

2.2.4 Programming Language

The STACOM program is implemented with the FORTRAN V language of UNIVAC systems, compiled with the EXEC-8 FORTRAN processor, and mapped by its MAP processor.

Detailed features of FORTRAN V programming language are described in Reference 2.

2.2.5 Operating System Requirements

Because the EXEC-8 operating system of the UNIVAC 1108 computer was used in the development of the STACOM program, the current edition of the STACOM program can only be executed under the EXEC-8 system. Furthermore, since a CalComp routine is linked with the program, the plotter must be part of the operating system. If such a hardware unit is not included in the system, the STACOM program must be updated to reflect this environment.

In addition, the current STACOM program was designed with the feature that all the desired output be put into a FORTRAN file designated as 100. Before executing this program, a file with the name 100 must be assigned. Otherwise, regular WRITE unit 6 will be the destination output file, e.g., the print output will go the user's demand terminal when it is run as a demand job.

As an example, the following is a complete list of EXEC-8 control statements which need to be prepared or typed in after the run card for properly executing the STACOM program.
The &SYM,P command directs the resulting plot card images to a Cal Comp plotter designated G9PLTF. The last &SYM command directs print output to a slow hardcopy printer designated T4.

2.2.6 Functional Limitations

While the STACOM program was designed and implemented with the intention that it be applicable as widely as possible, it does have certain limitations. These are due mainly to the limit of the program size (sum of I and D bank) allowed under the EXEC-8 system for simplistic programs. The maximum program size allowed is 65k words per program. Although it is more convenient for later use to assign all parameters with maximum values (as long as the overall program size is within the 65K-word limit) this results in greater expense in use of the program due to the higher core-time product. Therefore, it is recommended that all parameters be set at values just high enough for anticipated use.

After setting parameter values, the STACOM program capabilities are then limited to these assigned values. If a run requires that a certain parameter value be exceeded, the STACOM program must be recompiled and remapped.

2.3 INPUT

2.3.1 Data Requirements

A setup of input data is needed before starting a STACOM program run. The list of data items which need to be provided by the user are given here in temporal order and explained briefly. Detailed FORTRAN V formats for these are described in Table 3-1 of Section 3.

2.3.1.1 Number of Regions. The first datum needed by the STACOM program is the exact number of regions under consideration. This number (designated internally as NR1) instructs the program to divide all of the system terminations into NR1 regions.
2.3.1.2 Number of System Terminations, Number of Data Bases, and Number of Terminal Cities. The number of system terminations is the actual number of system terminations to be operated on by the STACOM program, and is designated internally as N1. In anticipation of possible multiple data bases at different locations, the number of data bases (designated internally as N7) informs the program that each system termination has N7 pairs of data (one pair per data base).

The number of terminal cities (NCITY) informs the program that NCITY V-H coordinates are to be provided later.

2.3.1.3 Identification of Data Bases and V-H Coordinates. N7 identifications provides the exact locations of data bases under consideration. All of the V-H coordinates for NCITY terminal cities are needed for calculating distances between any two cities.

2.3.1.4 Descriptions of System Terminations. For each of the system terminations under consideration, the set of data, i.e., identification, name, city location index, and traffic to all of N7 data bases are needed in order to properly execute the STACOM program.

2.3.1.5 Rate Structure and its Application Rule. There may exist one or more line tariffs applicable to different portions of any given state. The STACOM program has been designed with a capability to handle this situation. The number of applicable rate structures (line tariffs) and the rule governing their applications have to be input to the program by the user.

2.3.1.6 Traffic Density and Applicable Rate Structure for each System Termination. In order to accommodate the fact that costs for lines between high traffic density cities are much lower than for others, (e.g., TELPAK lines), the traffic density index and applicable rate structure for each system termination informs and directs the program to properly perform costing on lines connected to this termination.

2.3.1.7 Descriptions of Applicable Lines. The user dictates to the STACOM program the types of applicable communication lines by providing number of lines, their names and capacities, their desired maximum utilizations and their uses.

2.3.1.8 Descriptions of Chargeable Items. In addition to costs for lines, there are several other chargeable items such as modems, service terminals and drop charges. The user must provide the number of chargeable items and their names. Furthermore, the user has to provide the STACOM program with installation and monthly recurring costs for each chargeable item as a function of rate structure, line type, traffic density, and duplexing mode. This costing information is required to estimate overall cost of the to-be-designed communication network.
2.3.1.9 **Line Cost Data.** Installation and monthly recurring costs for lines for each applicable line type as a function of rate structure, traffic density, and duplexing mode are also required.

2.3.1.10 **Constraints on Formation of Regions.** The user can preload any number of system terminations to preselected regions if so desired by assigning them to their specific destinations (regions). He can also put constraints on preselected regions by not allowing any insertion of system terminations to these regions.

2.3.1.11 **Options on Regional Network Optimization.** The user can direct the STACOM program to perform regional network optimization on regions if required. This is done by simply specifying such requests to the program.

2.3.1.12 **Protocol Characteristics for Multidrop Lines.** The user must provide characteristics of line protocol to the program. For example, characteristics such as number of polling characters, NAK response characters, and message overhead characters are required. These data, along with the other line traffic characteristics data, enable the STACOM program to estimate the average terminal response time for a given multidrop line.

2.3.1.13 **Characteristics of Future Traffic.** Characteristics for future line traffic are also required. Data such as number of message types, their ratios, and average lengths allow the program to compute line service time and line utilization, which, in turn, are used to estimate the average terminal response time.

2.3.1.14 **Preloading System Terminations to Preselected Regions and Pre-Assigning Regional Switching Centers.** If the user wishes to assign certain system terminations to preselected regions and to pre-assigned regional switching centers, he can now proceed to do so. Otherwise, the program will perform these functions automatically.

2.3.1.15 **Assigning System Centroid.** If the STACOM program is required to divide system terminations into regions and to select regional switching centers, the system centroid is required so that the program can divide them properly (in a geographical sense).

2.3.1.16 **Descriptions of the Central Switcher.** Data describing the central switcher are needed to compute switcher turn-around time for a given transaction. These data include the estimated message rate at the switcher, number of transactions entering the switcher for completing a message, average service time per transaction, and number of processors available.
2.3.1.17  **Constraints on Multidropped Lines and Average System Response Time.** The user can impose a constraint on the number of terminals allowed on a multidrop line either by limiting the number of terminals on a multidrop line, or by setting up a maximum average response time limit to the multidrop line or both.

2.3.1.18  **CalComp Plot.** The user can request a CalComp plot of the final multidrop communication network if so desired. Of course, some installations may not have such a device and the STACOM needs to be recompiled without plotting routine.

2.4  **PROCESSING LOGIC**

The previous section described the type of input data needed by the STACOM program. This subsection will be devoted to the processing logic implemented in the program.

2.4.1  **Traffic Calculation**

2.4.1.1  **Traffic Conversion.** In the STACOM program, each system termination is provided with a set of traffic figures which represent outgoing traffic to and incoming traffic from each data base in the system. The unit of traffic is specified as characters per minute.

The traffic data for all system terminations are read into the matrix TRAFD(N1, 2, N7) during the data input phase, where N1 is the number of system terminations and N7 is the number of data bases. While the input traffic data are given in characters per minute, the STACOM program is designed to deal with traffic in terms of bits per second (BPS). Thus, at the time of program execution, all traffic data are converted into units of bits per second by multiplying them by a factor of 8/60. Here, we assume that synchronous communication is to be used.

2.4.1.2  **Origin and Destination Traffic by System Terminations.** Summations across the last subscript of the TRAFD matrix are performed to give total traffic originating from and destined for each system termination. The resulting data are stored in TRAFIT (N1) and TRAFDN (N1), respectively. More specifically, originating and destination traffic totals are given by

\[
TRAFIT(i) = \sum_{j=1}^{N7} TRAFD(i, 2, j)
\]

and

\[
TRAFDN(i) = \sum_{j=1}^{N7} TRAFD(i, 1, j)
\]
2.4.2 Distance Calculation

2.4.2.1 V-H Coordinates. The length of the line plays a major role in determining line costs on communication networks. While the common carrier is free to route the line over any desired path, and may switch the line to different paths to circumnavigate breakdowns or overloads, the line charges are normally independent of actual line layout and are based on the straight line distance between the points connected.

The AT&T has a system in which they have divided the United States by horizontal and vertical grid lines. By means of these lines, they give almost every city/location a vertical (V) and horizontal (H) coordinate, these coordinates provide the layout-free way of distance calculation.

2.4.2.2 Distances between System Terminations. With V-H coordinates as defined by the AT&T, the distance between any two locations is calculated as follows (Reference 3):

1. Obtain the V and H coordinates for these two locations.
2. Obtain the difference between the V coordinates and the difference between the H coordinates of these two locations.
3. Square each difference obtained in 2 and take a summation of both squares.
4. Divide the sum obtained in 3 above by 10. Round to next integer number if any fraction is obtained.
5. Obtain the square root of the result obtained in 4 above. This is the distance between the given locations in miles. (fractional miles being considered as full miles.)

For example, to calculate the distance between Austin and Dallas, Texas, we proceed as follows:

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin</td>
<td>9005</td>
<td>3996</td>
</tr>
<tr>
<td>Dallas</td>
<td>8436</td>
<td>4034</td>
</tr>
<tr>
<td>Difference</td>
<td>569</td>
<td>38</td>
</tr>
</tbody>
</table>

\[
\text{Distance} = \sqrt{\frac{(569)^2 + (38)^2}{10}} = \sqrt{\frac{323761 + 1444}{10}}
\]

\[
= \sqrt{32521} = 181 \text{ miles}
\]
When a specific location in the United States is not designated with specific V and H coordinates, it is normally assigned with the same V and H coordinates as the closest location.

Following the procedures as given above, the distance between any given two system terminations is calculated and stored in arrays DSTDCE or IVHD.

2.4.2.3 Distance Data Compression and Overflow Table. Given N system terminations, there are N(N-1)/2 combinations in choosing two system terminations from them. Furthermore in any given state, there exist only a few large inter-terminal distances. These two facts indicate that some reduction in resulting STACOM program size can be made by performing compression of distance data. Two efforts have been undertaken for that purpose.

Under the UNIVAC system, each computer word is 36 bits long. We divide each word into four 9-bit segments. Each segment is used to store one distance datum with values ranging from 0 to 511. To compensate for the fact that some distances data may be greater than 511, an overflow table IVRD is provided to collect oversized distance data. In other words, given two system terminations with indices I and J, its distance is recorded into DSTDCE as follows:

(1) Find corresponding V-H coordinates of locations for both system terminations.
(2) Calculate distance D according to the procedure given in Paragraph 2.4.2.2.
(3) Find a unique and absolute location L in DSTDCE, by using the following equation:
   \[ L = I \times NPC + J - \Delta(I) \]
   where
   \[ \Delta(I) = \sum_{i=1}^{I} i, \]
   and
   \[ I < J \]
   NPC = number of distinctive locations in the system

This mapping function is performed by subroutine LINK,

(4) Define
   \[ L1 = \lfloor (L-1)/4 \rfloor + 1 \]
   \[ S1 = (L-1) \mod 4 + 1 \]
where \( \lfloor x \rfloor \) = the integer part of \( x \) and

\[
D_1 = D \text{ if } D < 511 \\
511 \text{ if } D \geq 511
\]

(5) Store \( D_1 \) in segment \( S_1 \) of entry \( L_1 \) of table \( DSNCE \).

(6) If \( D \geq 511 \), store \( L \) and \( D \) in next available space of table \( IVRD \).

On the other hand, given two system terminations with indices \( I \) and \( J \), the retrieval of distance is performed as follows:

(1) Calculate \( L, L_1 \) and \( S_1 \) as described above.

(2) Retrieve the content \( D_1 \) in segment \( S_1 \) of entry \( L_1 \) of table \( DSNCE \). If \( D_1 < 511 \), it is the distance.

(3) If \( D_1 = 511 \), retrieve the second element of the row of table \( IVRD \), whose first element contains value \( L \). The retrieval value is the distance.

2.4.3 Formation of Regions

After traffic summations and distance table formation are completed, the STACOM program starts to form regions. It assigns all of the non-preloaded system terminations to regions which can accommodate them. Figure 2-8 illustrates the process of such a function.

The process begins with an estimation of the traffic per region, called \( TPR \), which is obtained by averaging the total non-binding traffic, i.e.,

\[
TPR = \frac{TPR_1}{ANR_1}
\]

with

\[
TPR_1 = \sum_{1 \leq i \leq N1} \left[ TRAFIT(i) + TRAFDN(i) \right]
\]

where

\( I \) = the set of system terminations in preloaded regions which do not allow other system terminations to be inserted to them

\( ANRI = NR_1 - \) [number of preloaded regions which do not allow any insertions]
Figure 2-8. Flow Chart for Formations of Regions
When the number of regions is 1, all of the system terminations are assigned to the region and no other region formation process is performed. Otherwise, the program starts assigning system terminations to regions (in a cardinal order) which allow their entries.

The following two subsections describe the detailed processes for assigning system terminations to a region either with preloading or without preloading.

2.4.3.1 Assigning System Terminations to a Region without Preloading. When a region NREG is not preloaded with any system termination, processing continues with the finding of the farthest unassigned system termination (NS1) from the system centroid (NSCC1). This system termination is then assigned to the region NREG; its incoming and outgoing traffic is added to the partial sum traffic, called TRFS. The resulting TRFS is then tested. If it is greater than TPRL, (lower bound), which is equal to 0.9 x TPR, assignment processing for region NREG ends with re-estimating TPR and TPRL which are obtained as follows:

\[ TPR = \frac{TPR1}{ANR1 - 1.} \]

On the other hand, if TRFS is less than or equal to TPRL, additional system terminations can be assigned to this region. The next system termination for addition to this region is selected by finding the nearest unassigned system termination, called NS2, from NS1. NS2 is then assigned to region NREG and its traffic added to TRFS. The value of TRF is again tested against TPRL to determine if other additions are possible.

This process is repeated until partial regional traffic sum TRFS is greater than TPRL. At this point, the region is considered full and addition of system terminations to this region stops. However, if the region being filled is the last one, all remaining system terminations are placed into this last region. Otherwise, the program continues to work on the next region. Before leaving region NREG, it re-estimates TPR and TPRL as shown before.

2.4.3.2 Assigning System Terminations to a Region with Preloading. If the region NREG is a preloaded region, i.e., it has been preloaded with system terminations, the program continues with a test. The test is needed to determine whether region NREG will accept any additional system terminations. If other insertions to the region are not allowed, the processing on this region stops and continues to the next region.

Otherwise, the program starts adding traffic to all preloaded system terminations to TRFS and finding the farthest unassigned system termination NS1 from the system centroid. It then tests whether TRFS is
greater than TPRL. If it is greater, the program stops here and continues to process the next region.

When TRFS is less than TPRL, the program checks whether there is a preselected RSC for the region NREG. If there is, the program uses the RSC as the NS1. Then it follows the same procedure as described in paragraph 2.4.3.1 to add more system terminations to the region.

It should be noted that STACOM has been implemented in such a way that when it is desired to preload some or all regions, the last one need not be specified. The program will assign the rest of the unassigned system terminations to the last region.

2.4.3.3 Example for Formation of Regions. Figure 2-9 illustrates the results of applying the formation of region logic to a Texas communication system with 265 system terminations. In this example run, neither preloading of system terminations nor preselection of regional switching centers are requested. In other words, the program is asked to perform automatic regional formations and to select the regional switching centers. System termination Austin is chosen as the system centroid.

The total amount of traffic, TPR1 is at a rate of 1585.02/bps, and the number of regions is 2. Therefore, at the beginning, TPR is given as 1585.02/2=792.51 bps, and TPRL = 713.26 bps. In the process of assigning system terminations to region 1, El Paso is found to be the farthest location from Austin, i.e., NS1 = the internal index for system termination El Paso. With NS1 available, the program starts the procedure of searching for NS2, adding its traffic to partial sum TRFS and testing whether TRFS is greater than TPRL. It repeats the same procedure 123 times until TRFS has reached the value of 750.08 bps which is greater than TPRL.

2.4.4 Selection of Regional Switchers

Selection of regional switching centers follows formation of regions as described in Paragraph 2.4.3. For a given region, its regional switching center (RSC) can be either preselected by the user or be chosen by the program. In the latter case, the program selects the system termination within the region such that total intra-region traffic-distance products are minimized.

The functional flow chart of RSC selection is depicted in Figure 2-10. Processing begins with assigning 10^{12} to WCASE (as base for traffic-distance product sum). It then calculates the estimated sum of all traffic-distance products with each system termination in the region as an RSC site. The sum, called SUMT, is obtained as follows:

\[
\text{SUMT} = \sum_{i=1}^{\text{NMBR}} \left( \text{TRAFDN}(i) + \text{TRAFIT}(i) \right) \times \text{DIST}(i,K)
\]
where

\[ \text{NMBR} = \text{number of system terminations in the region under consideration} \]

\[ K = \text{the index of the system termination considered as the trial RSC site} \]

\[ \text{DIST}(i,K) = \text{the distance between system termination } i \text{ and the RSC trial site } K \]

The resulting SUMT is then compared with WCASE. If SUMT is found to be less than WCASE, the value for WCASE is replaced by the value of SUMT and the corresponding index for the RSC trial site is the updated RSC, called NRSC.

Figure 2-9. Example of Region Network Formation and Regional Switcher Selection
NRSC = INDEX FOR REGIONAL SWITCHING CENTER
SUMT = \sum_{i} \left( \text{TrafFDN}(i) \times \text{TrafIT}(i) \right) \times \text{DIST}(i, K)
NMBR = NUMBER OF SYSTEM TERMINATIONS IN THE REGION
DIST(i, K) = DISTANCE BETWEEN SYSTEM TERMINATIONS i AND K

Figure 2-10. Flow Chart for RSC Selection
After the above processing has been repeated \textit{NMBR} times, the resulting \textit{NRSC} is the index for the selected RSC and \textit{WCASE} the region's minimal traffic-distance product sum.

When a regional switching center is preselected by the user, the program skips the process as described here.

Following the selection of a regional switching center for a given region, the program continues to perform regional network formation and network optimization before it repeats the selection of regional switching centers for remaining regions.

The process of regional network formation and optimization is discussed in Paragraphs 2.4.5 and 2.4.6.

\textbf{2.4.4.1 Example for Selecting a Regional Switching Center.} Following the formation of regions in the example given in Paragraph 2.4.3.3, the program has chosen Brownwood of Brown county as the switcher location for Region 1 and Tomball of Harris county as the switcher location for Region 2. Both locations have been found to provide the minimal traffic-distance product sums for respective regions. These two cities are shown in Figure 2-9.

\textbf{2.4.5 Formation of Regional Star Networks}

Formation of a regional network starts with a star network and then continues with an optimization process which, most of the time, results in a cost-saving multidrop network. This subsection describes the process of forming a star network, which is depicted in Figure 2-11. The initial regional network is formed by directly connecting each system termination to the regional switching center. Selection of these intra-region lines is constrained by the rule that each selected line should maintain the line utilization factor, called \textit{RHO}, at a value less or equal to a preselected number, say, 0.7.

For each system termination in the region, the program finds incoming and outgoing traffic, TRFOUT and TRFIN, and also its distance, DSTN, from the RSC for a system termination in the region. The program calls subroutine LINNUM, which constructs a line configuration LDUMMY and calculates its line utilization, based on the values of TRFIN and TRFOUT provided. The processing continues to calculate both the cost, \textit{COST}, for the derived line configuration LDUMMY and its response time RSPTIN. Finally, all these data are stored for later printout and comparisons.

The derivation of line configuration LDUMMY by subroutine LINNUM and the associated cost, \textit{COST}, deserves more explanation. The program assumes that the duplexing mode for all line types under consideration to be half-duplexed. Therefore, subroutine LINNUM will sum up TRFIN and TRFOUT and find an applicable line with the least capacity which assures less than 0.7 of utilization. When the highest capacity line cannot handle the traffic, the routine will try to add one additional line with least capacity until the constraint of 0.7 utilization factor is satisfied. With line configuration LDUMMY obtained, calculation of cost,
Figure 2-11. Flow Chart for Regional Star Network Formation
COST, for the direct link between system termination K and the RSC is performed by ICOSTJ. The routine ICOSTJ calculates all of the related installation and annual recurring costs for lines and other chargeable items. All of these itemized costs are then summarized as COST. Cost calculations are performed on the basis of the rate structures applicable to system terminations at both ends.

2.4.5.1 Examples of Line Selections. Table 2-1 lists some examples of line configurations results obtained by LINNUM, and illustrates how the LINNUM subroutine selects lines for given traffic. The first column of the table represents total traffic (sum of TRFIN and TRFOUT). In this example, it is assumed that only line types with capacities of 300 bps, 1200 bps and 4800 bps are under consideration. Line utilization factor has been constrained to not greater than 0.7.

2.4.6 Optimization of Regional Networks

After completing the formation of a regional star network, the program proceeds to the optimization process, if requested. The optimization process basically utilizes a technique developed by L. R. Esau and K. C. Williams (Reference 1) and is used to minimize line operating costs. The actual implementation of the technique has been made with several additional constraints for practical reasons.

Before going into detail, here is a brief explanation of the goal and process of network optimization of a regional star network. Figure 2-12 depicts a typical star network in which each system termination has a direct link, called central link, to the central regional center. The goal of optimization is to reduce line costs by eliminating as many central links by connecting the associated system terminations to their nearby system terminations as possible, until it is no longer cost-effective to do so. Figure 2-13 shows a typical multidrop network.

Table 2-1. Examples of Line Configurations Obtained by Subroutine LINNUM

<table>
<thead>
<tr>
<th>Traffic (bps)</th>
<th>Line Type</th>
<th>300 bps</th>
<th>1200 bps</th>
<th>2400 bps</th>
<th>4800 bps</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>500</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>850</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1300</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3500</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
after this goal is met, assuming that the line cost is only a function of distance. While this example as given depicts the main concept of network optimization, it does not illustrate the process of sizing each newly formed multidrop line to reflect the increase of traffic resulting from the addition of new system terminations.

The following section describes the logic implemented in this program.

2.4.6.1 Network Optimization by Esau-Williams Technique. Before explaining the logic for network optimization implemented in the STACOM program, a brief explanation of the Esau-Williams network optimization process is appropriate. With a given star network, the basic process of the Esau-Williams technique is to repeat two basic steps until it is no longer possible to derive any cost saving.
For the convenience of the following discussion we define a sub-network (subnet) as a tree-type multidrop line consisting of one or more system terminations and having a central link connected to the regional center. Each central link of a given star type network is a simple sub-network by definition.

The first step involves searching for the best central link of a system termination, K, so that its elimination and the subsequent reconnection of the rest of the sub-network to a nearby sub-network L provides the best cost saving. In other words, for each system termination, i, with a central link to the regional center, this routine estimates the best saving, $S_i$, resulting from eliminating the given central link and reconnecting the rest of the subnetwork to a nearby subnetwork beginning with $L_i$. If we express it as a formula, then

$$K = i \text{ such that } S_i = \max_{j \in C} \{S_j\}$$

$$L_i = j \text{ for which the integration of } K \text{ and } L \text{ sub-networks provides } S_K \text{ which is the best saving}$$

where

- $C = \text{the set of system terminations with central links to regional center}$
- $j = \text{the first system termination of sub-network } L$

The other step involves network updates after it has been determined that the central link from system termination K is to be eliminated; this step will integrate remaining subnet K with subnet L utilizing an alternate route.

It should be noted that although this network optimization process will generate the best network most of the time, it does not always provide the best one. In other words, this technique generates the local optimal solution rather than the global solution. This is because the first selection of a central link for elimination dictates the final network to be created by repeating the process as described above. However, as shown in Reference 4, the process does provide a solution which is always close to, if not, the best.

2.4.6.2 Network Optimization Logic in STACOM Program. The optimization logic as implemented in the STACOM program basically utilizes the Esau-Williams technique. However, constraints have been incorporated into it in order to satisfy project requirements and to eliminate unnecessary searching. Figure 2-14 shows the functional flow chart for the overall logic.

The optimization process starts with the test to see whether there is only one sub-network left. If this is true, it stops. Otherwise, the program, utilizing four variables K, L, M and KI, starts evaluating possible cost saving by eliminating central link K and
Figure 2-14. Flow Chart for Subroutine ESSWIL
reconnecting the rest of sub-network K to sub-network L through system terminations M of L and KI of K as shown in Figure 2-15.

Selections of values for variables K, L, M, and KI are in the following way. For each processing cycle, searching and updating, K is assigned the index values from the first sub-network to the last one of the existing network. For each K, L is assigned index values from the first sub-network to the last one except K ≠ L. With values for K and L chosen, M is assigned the index values of all the system terminations on sub-network L and KI the index values of all the system terminations on the sub-network K.

For each given set of K and L, the program tests whether the sum, NT, of numbers of system terminations for both sub-networks exceeds the value of NTERMS which constrains the number of system terminations on a multidrop line. If this is true, it skips the process of calling subroutine TRYLNK, because it is not possible to integrate both sub-networks without violating the said constraint. Otherwise, it continues to the distance test.

\[ K = \text{THE SUBNETWORK BEGINNING WITH SYSTEM TERMINATION } K \]
\[ L = \text{THE SUBNETWORK BEGINNING WITH SYSTEM TERMINATION } L \]
\[ M = \text{THE SYSTEM TERMINATION ON SUBNET L TO WHICH KI IS TO BE CONNECTED} \]
\[ KI = \text{THE SYSTEM TERMINATION ON SUBNET K FROM WHICH SUBNET K IS CONNECTED TO M OF SUBNET L} \]
\[ DREF = \text{THE DISTANCE BETWEEN SYSTEM TERMINATIONS } K \text{ AND THE RSC} \]
\[ DTTRY = (\text{THE DISTANCE BETWEEN SYSTEM TERMINATIONS } KI \text{ AND M})/2 \]

Figure 2-15. Relationship among K, L, KI, and M Parameters
The program first calculates the distance $D_{REF}$ between system termination $K$ and the region switching center for each $K$, and then the $D_{TRY}$ which is half of the distance between system terminations $K_I$ and $M$ for each combination.

If $D_{TRY}$ is greater than $D_{REF}$, the program skips the process of calling on subroutine $TRYLNK$. Otherwise, it calls on subroutine $TRYLNK$. The purpose of subroutine $TRYLNK$ is to estimate the possible cost saving resulting from eliminating central link $K$, and integrating sub-networks $K$ and $L$ by connecting system terminations $K_I$ of $K$ and $M$ of $L$. If the saving is better than the maximum saving obtained so far, it is used as the up-to-date best cost saving under the set of values for $K$, $L$, $K_I$, and $M$. A detailed description of functions performed by subroutine $TRYLNK$ is given in Paragraph 2.4.6.3. After all possible combinations for $K$, $L$, $K_I$, and $M$ have been tested and it has been found that the up-to-date best cost saving is positive, the program performs the second function of network optimization, i.e., updating the network. It then repeats the whole process on the newly updated network which happens to have one less central link.

If the up-to-date maximum cost saving is non-positive, the optimization process stops here.

2.4.6.3 Function Performed by Subroutine $TRYLNK$ for a Given Set of Values $K$, $K_I$, $L$, and $M$. The processing, as shown in Figure 2-16, starts with estimating the total amount of traffic that a single multidrop line (sub-network) of integrating subnetworks $K$ and $L$ needs to handle. It then estimates the required line configuration, $LDUMMY$, by calling subroutine $LINNUM$ which has been described in Paragraph 2.4.5.1. Based on $LDUMMY$, the program estimates the average response time and tests it against the user-provided response time limit by calling subroutine $RSPNSE$. If the estimated response time is not satisfied, the program updates the line configuration $LDUMMY$ to the next higher line type and repeats the process of estimating its average response time and testing it against the given constraint. This process ends when either there is a satisfied line configuration or it is not possible to upgrade any further.

When a satisfied line configuration is obtained, the program continues to estimate its cost saving, based on the assumed integrated sub-network. If the resulting cost saving is better than the up-to-date best cost saving, it replaces all of the maximum saving parameters, which are used to keep tracking the up-to-date best network changes; it then returns to its calling routine. If there is no line configuration satisfying the response time constraint, the process stops and the program returns to its calling routine.
START

FIND
1. TRFIN
2. TRFOUT

FIND LDUMMY

CALL LINNUM

UPGRADE LDUMMY

ESTIMATE RESPONSE TIME RSPTIM

CALL RSPNSE

IS LINE UPGRADE POSSIBLE?

IS RSPTIM SATISFIED?

Y

FIND COST IF SUBNETWORKS K AND L ARE INTEGRATED

N

IS SAVING BETTER?

Y

UPDATE MAXIMUM SAVING PARAMETERS

TERMINATE

Figure 2-16. Flow Chart for Subroutine TRYLNK
2.4.6.4 Functions Performed by Subroutine RSPNSE. Figure 2-17 shows the flow chart of the subroutine RSPNSE. This subroutine calculates six items of delays: polling, message transmission time from a terminal to the central switch, input buffer queue time, service time, output buffer queue time, and returned message transmission time from the central switcher to the same terminal.

After summing up these delays as RSPTIM, this subroutine compares its value with the upper bound response time as set up by the user. It assigns 1 to IOK as an indication of satisfying response time requirement and returns.

![Flow Chart for Subroutine RSPNSE](image)

Figure 2-17. Flow Chart for Subroutine RSPNSE
2.4.6.5 Network Updates by Subroutine UPNETW. If there is a positive cost saving after trying all possible combinations for parameters K, KI, L, and M, subroutine UPNETW is called upon to perform the other function for each cycle of the network optimization process as described in Paragraph 2.4.6.1.

In the STACOM program, subroutine UPNETW performs the following main functions: (1) updating of network descriptions, (2) revision of relevant accounting data (such as the number of terminals on the new L sub-network, its average response time, and total traffic).

2.4.7 Formation of an Interregional Network

The inter-regional network is formed by erecting communication lines between the regional switching centers (RSCs). The initial network has a direct line between any two RSCs.

As shown in Figure 2-18, for each combination of two RSCs I and J, the maximum traffic in either direction is considered as the design traffic between these two RSCs. This is different from intraregional line selection because it is assumed that full duplex lines are to be used. The traffic matrix TRM contains traffic data between RSCs. With this information, line configuration LINEQU between RSCs I and J is obtained by calling subroutine LINNUM.

Cost of line configuration LINEQU is then estimated and added to the total cost.

2.4.8 Optimization of an Interregional Network

After the initial interregional network is completed, the program starts a line elimination process in order to obtain a cost-effective network.

Figure 2-19 shows the basic topological consideration involved in line elimination. In considering whether line I-J can be eliminated, the algorithm tries to divert I-J traffic to other lines with excess capacity, for example, over route 1-4-3. If there is no alternate route with enough excessive capacity to handle I-J traffic, the program begins adding capacity to alternate routes in order to accommodate the required traffic. It then estimates the cost saving under the proposed modifications.

The algorithm iterates the above described process for all combinations and records the best cost saving and the best line elimination. It then updates the network.

This cycle of searching for the best cost saving and updating the network repeats continuously until cost savings can no longer be realized.
Figure 2-18. Flow Chart for Intraregional Line Selection
2.4.8.1 Interregional Network Optimization Logic Implemented. Figure 2-20 depicts the functional flow chart for the interregional network optimization as implemented in the STACOM program.

A parameter, I, is used to select one of the RSC nodes to be considered for line elimination. A test is then made on RSC I to insure that at least three links to other RSCs exist. If I has at least three links, another parameter, J, is used to select any other RSC node for trying to eliminate its link to I. J is tested to insure that it has three links to other RSCs and J is different from I. Another test is made to insure that I and J are connected to each other. If any of these conditions are not met, RSC node J + 1 is selected and these three tests are repeated.

If these conditions are met, a test is carried out to see if sufficient network connectivity will still be maintained if connection I-J is removed. Due to the consideration of availability, the program is designed in such a way that each RSC node will have at least two communication links to other RSCs and each RSC node will be connected to every other RSC node through no more than one intermediate node.

If the network connectivity requirement can be maintained with the removal of link I-J, the program searches for alternate routes with excess capacity in an effort to re-route the I-J traffic load without increasing network capacities. If all I-J traffic can be successfully diverted in this manner, the I-J link is eliminated and the network traffic matrix and costs are re-calculated; the process then begins anew.
Figure 2-20. Flow Chart for Interregion Network Optimization
If all I-J traffic cannot be diverted through existing network routes with excess capacities, the capacity of the first available alternate route is increased to handle the remaining traffic. The cost saving is determined as equal to the original cost of the line removed minus the cost for the capacity increase. If the cost saving is an improvement over previous trials, line and traffic data are saved to reflect the up-to-date best modification of the network.

At the conclusion of each cycle, if the cost saving is positive, the line and traffic data associated with the best saving are used to eliminate the line, update the network, and recost the network.

The process is continually iterated for each updated network configuration until cost savings are no longer positive.

2.5 OUTPUT

The STACOM program generates regular printer output and a CalComp plotter output. In addition, when the program is run as a demand job, run-status output will show up on the interactive terminal. This part of the printout provides information on the progress of the run.

Details of data contained in the regular printer output are given in Paragraph 2.5.1; Paragraph 2.5.2 describes the CalComp plot.

2.5.1 Printer

The printer output contains all the data resulting from the running of the STACOM program. The amount of printout data depends upon the number of system terminations operated and also upon the number of functions executed in each specific run.

Following is a list, in temporal order, of the data items which a run may produce.

2.5.1.1 Line Type and Transmission Line Characteristics. The first set of data are the line type and transmission line characteristics as used in the run. For each line type, the polling protocol data and modem turn-around time data, etc., are provided.

2.5.1.2 Message Characteristics. Message characteristics are the next set of data output from the program. They include average input message length, average output message length and overall average message length.

2.5.1.3 Preloading of System Terminations and Preselection of Regional Switcher Locations. If there are any pre loadings of system terminations and/or pre-selections of switcher locations, this information will be provided in the printout. Otherwise, no data will be shown in this regard.
2.5.1.4 Traffic and Distance Tables. These are tables which show both traffic from/to all system terminations and distances between system terminations.

The first table gives the traffic data from each system termination to/from each data base; the next one gives the traffic data destined to and originating from each system termination. The last table shows the distance data between any two system terminations.

2.5.1.5 System Centroid and the Utilization Factor of the Central Switcher. The system centroid as designated by the user is printed next as a reminder. After this, the CPU utilization factor of the central switcher as calculated by the program is printed to indicate the load.

2.5.1.6 System Terminations in a Region and its Regional Switching Center. For each region, the program prints out the identification and name of each system termination in the region. These system terminations may have been pre-loaded or assigned to the region by the program. The program also prints out the location of the ASC for the region, which is either pre-assigned by the user or selected by the program.

2.5.1.7 Star Network and its Costs. After showing what system terminations are in the region, the program prints out the regional star networks and costs associated with each central link. It also provides summarized costs. Detailed descriptions for each central link are given below.

2.5.1.7.1 Line Configuration and Effective Utilization. The line configuration for each central link is printed as a column vector, which has the same number of line types used in the run. The effective line utilization is also printed to show the traffic load from the system termination.

2.5.1.7.2 Distance. The distance from the system termination to the regional switching center is printed.

2.5.1.7.3 Line Traffic and Effective Response Time. The amount of traffic from/to the system termination is printed before the effective line response time as calculated by the program is printed. The calculation is based on the line configuration and traffic as shown and should be better than the response time requirement.

2.5.1.7.4 Installation and Annual Recurring Costs. The installation and annual recurring costs for providing the central link are given in terms of chargeable items such as service terminal, modem, line and drop. Partial sums for the line are also printed. Finally, total installation and annual recurring costs for each chargeable item and for the overall star network are printed.
2.5.1.8 Final Optimized Network and its Costs. After performing optimizations on the star network, the program prints out descriptions for each multidrop line in the final optimized network. The following list shows the data items which may be printed.

2.5.1.8.1 Multidrop Line Configuration. Each multidrop line has an index, the beginning terminal and number of terminals on the line. The exact line configuration is printed as a column vector, with only one non-zero element. The content of that non-zero element must be one, due to the fact that multidropped terminals can only perform on one line.

2.5.1.8.2 Line Utilization, Mileage, Traffic, and Response Time. The line utilization, total mileage and incoming/outgoing traffic on each multidrop line are printed. The program next prints the average response time, which should be better than that required by design, to be expected by each user terminal on the line.

2.5.1.8.3 Installation and Annual Recurring Costs. The amount of installation and recurring costs are then listed in terms of chargeable items as explained in Section 2.5.1.7.4.

Finally, total installation and annual recurring costs for each chargeable item and for the overall network are printed.

2.5.1.9 Network Drawing. A network diagram in terms of tree-type relationship is last printed. It uses the system termination identification as nodal notation.

2.5.1.10 Initial Interregional Network. If formation and optimization of the interregion network is required, the program will perform these functions and print its initial and optimized network. For each pair of HSCs, the program prints out line names, configuration, utilization, and installation and recurring costs. Total network cost is also provided.

2.5.1.11 Optimized Interregional Network. The program prints out similar data for the final optimized interregional network after completing the network optimization.

2.5.2 CalComp Plot

A CalComp plot subroutine has been incorporated into the STACON program for the purpose of providing a visual plot of each optimized regional network obtained by the optimization process. The subroutine converts each final optimized regional network into a two dimensional plot, utilizing the CalComp plotter. It should be noted that the CalComp plot is an optional product. If desired the user can command the STACOM program not to generate the plot.
2.6 SYSTEM CONFIGURATION

In this section, we will describe the basic computer system required to run the STACOM program.

2.6.1 Hardware

The following list describes the hardware units that should be part of the computer system on which the STACOM program is run.

2.6.1.1 Central Processing Unit. Due to the fact that the STACOM program is coded with the FORTRAN V language and compiled and mapped under the EXEC-8 operating system of the UNIVAC 1108 systems, a UNIVAC 1108 CPU or one equivalent to it is a prerequisite of the use of the STACOM program. When this type of CPU is not available, some conversion efforts on the STACOM program may be required.

2.6.1.2 Main Core Storage. Although the core size required by the STACOM program varies by parameter values assigned, it is generally true that 65K words would be a minimal requirement.

2.6.1.3 CalComp Pen Plotter. A CalComp pen plotter is required for the use of the STACOM program. If other types of CalComp plotters, e.g., CalComp Model 1675 are to be used, the plotting subroutine of the STACOM program needs to be revised.

2.6.1.4 Line Printer. A regular printer to receive FORTRAN output files is needed. It will print out all run results collected by file 100.

2.6.1.5 Demand Terminal. A demand terminal provides the user with an alternate way of running the STACOM program, although the program can be run as a batch job. With the demand terminal, a user can interactively perform the program execution.
SECTION 3
PROGRAM OPERATIONS

3.1 INTRODUCTION

This section is intended for use as a reference manual for the user, both to prepare input data and to operate the STACOM program. With this in mind, this section is devoted to an explanation of how input data are prepared, how the program is executed, and what the input/output of the program is to be.

3.2 ENVIRONMENT

3.2.1 Hardware

The following list describes the hardware units that should be part of the computer system on which the STACOM program is run.

3.2.1.1 Central Processing Unit (CPU). Because the STACOM program is coded with the FORTRAN language and compiled and mapped under the EXEC-8 operating system of the UNIVAC 1108 systems (see Paragraph 1.1), a UNIVAC 1108 CPU or one equivalent to it is a prerequisite for using the STACOM program. When this type of CPU is not available, some conversion effort on the STACOM program may be required.

3.2.1.2 Main Core Storage. Although the core size required by the STACOM program varies with the parameter values assigned, it is generally true that 65k words would be a minimal requirement.

3.2.1.3 CalComp Pen Plotter. A CalComp pen plotter is required for the use of the STACOM program. If other types of CalComp plotters, e.g., CalComp Model 1675, are to be used, the plotting subroutine of the STACOM program has to be revised.

3.2.1.4 Line Printer. A regular line printer to receive FORTRAN output files is needed. It is to print out all run results collected by file 100.

3.2.1.5 Demand Terminal. A demand terminal provides the user with an alternate way of running the STACOM program, although the program can be run as a batch job. With the demand terminal, a user can interactively perform the program execution.
3.2.2 Software

3.2.2.1 Programming Language. The STACOM Program is implemented with the FORTRAN V language of the UNIVAC system, compiled by the EXEC-8 FORTRAN Processor FOR, and mapped by the mapping processor MAP. Because of the inclusion of a plotting subroutine, the system library file LIB*PLOT$ is required during mapping.

An understanding of the FORTRAN V features is available in Reference 2.

3.2.2.2 Operating System. The EXEC-8 operating system of the UNIVAC 1108 computer system is used in the development of the STACOM program. As this operating system has been used for executing regular FORTRAN V programs this same operating system must be used for executing the current edition of the STACOM program.

The STACOM program has been designed so that all of the desired printer output will be dumped to file 100. Therefore, before executing the STACOM program, an alternate file 100 must be assigned. Otherwise, regular WRITE unit 6 will be the destination device; this will make it awkward when runs are performed via a demand terminal since most of the output from the program uses 132 characters per line.

Furthermore, an execution of the program will generate a punch-card image file. It is, therefore, recommended that a file be assigned to store the punch-card file, and that this later be directed to a CalComp plotter. An alternative is to have a command statement which requests the operating system to $SYM the output punch-card file to a CalComp pen plotter.

3.2.3 Functional Limitations

While the STACOM program has been designed and implemented with the intent that it be as widely applicable as possible, it does have certain limitations. Following is a list of functional limitations that exist in the program.

3.2.3.1 Program Size. Under the EXEC-8 operating system, the size for regular programs is limited to 65k words per program. Because of this, assignments of parameter values during the compilation stage are conditioned to this limit of the overall program size when mapped. Although it will be more convenient for later uses of the STACOM program if all of the parameters are assigned with maximum values within the limit of 65k words, this will increase the run cost. This is because of the core-time product charge.

3.2.3.2 Parameter Variables. The PARAMETER statement of the FORTRAN language is one of those commands which make the language a powerful tool in problem solving.
To accomplish the goal of making the STACOM program a widely usable tool for network design, it has been implemented with several parameter variables. For each compilation of the program, a set of values is assigned to the parameter variables. Therefore, any subsequent use of the STACOM program will be limited to cases where the actual values assigned to the variables are within the parameter values defined during compilation. Any run whose input data violates this rule will need modification of the parameter values of the program, recompilation, and remapping. For example, NP1 is a parameter variable which is used to make the number of system terminations allowed in a system a variable. A choice of NP1 as 105, for example, dictates that the STACOM program can only be used in systems where 105 or less system terminations are under consideration. Any run which has a number of system terminations greater than 105 will result in either an abnormal run termination or a normal run termination with unwanted output.

3.2.3.3 Response Time. The response time algorithm implemented in the program is based on the model (Reference 5). In applying this program to a given system, some consideration of the applicability of the response time algorithm is required. If the central switcher does not behave similar to this model the response time subroutine RSPNSE has to be revised and recompiled and the STACOM program has to be re-mapped.

3.2.3.4 CalComp Plot. The graphic output portion of the STACOM program has been implemented with the plotting routines designed for the CalComp pen plotter. If other types of CalComp plotters, e.g., CalComp Model 1675, are to be used, the plotting subroutine of this program needs to be revised and recompiled and the STACOM program has to be re-mapped.

3.3 RUN DESCRIPTION

3.3.1 Initialization and Setup

When the STACOM program is executed from an 80-character/line demand terminal, an alternate file, 100, to be used as a printer output file, must be defined. Otherwise, all printout data will be directed to the terminal which will produce interleaving output. The statement @EASY,UP 100 defines the alternate file.

In addition to the redirection of output file destination, the user has to direct the punch-card file to a proper unit for a CalComp plotter. As an example, the statement @SYM,P PUNCH$,,9PLTF will direct the punch-card images to a CalComp plotter designated with G9PLTF.

The preparation of input data can be best described by referring to Table 3-1 which shows all of the data items with their required formats. The table is self-explanatory, but some of the data items deserve additional description.
Because the exact number of data bases varies from State to State, the format for item 5 allows a maximum of 5 data bases wherein the last three pairs of entries must be given on a separate card.

The notation $\lceil X \rceil$ for item 8 indicates that the exact value is equal to the next integer which is greater than or equal to X. The format for line recurring costs has been designed with the assumption that both linear and nonlinear functions will be used in tariffs for line services. Because of this, the STACOM program provides options for either scheme. When a cost function is nonlinear, it is assumed to be stepwise and only eight steps are allowed. If eight are not enough, the program has to be updated.

The amount of input data for item 15 varies from one run to another. The program has default values of zeros for all entries in IACTN (NR1,2). A zero for the first element indicates the acceptance of additional system terminations into a region when it is a preloaded region; a zero for the second element indicates that the optimization process for the region is not needed.

When a user decides either to exclude the addition of other system terminations into a preloaded region, or to request an optimization process performed upon a specific region he must so inform the STACOM program by adding data cards with two integer numbers. The first number gives the region index; the second number indicates the action: 1 indicates insertion exclusion, and 2 indicates optimization. When all requests for actions have been made, a card with two zeroes is required to indicate that fact.

Finally, item 20 provides the tool for a user to preload system terminations to certain regions, and/or preselect the regional switching center. Three numbers are needed for each action. The first number, called NCODE, directs the specified action: 1 assigns a system termination to a specific region; 2 assigns a system termination as the RSC for a specific region. The second number, called NSTATE, gives the identification number for a system termination to be assigned to a region or to be selected as an RSC. The third number, called NREGQ, designates the region to be acted upon. When the first number has a value of three, the assignment selection activity terminates.

3.3.2 Run Options

As indicated in Table 3-1, there are several independent variables provided only at the time of execution. This provides additional capabilities to the STACOM program.
### Table 3-1. Formats for Input Data

<table>
<thead>
<tr>
<th>Item No.</th>
<th>Item Description</th>
<th>Names of Internal Variables/Arrays</th>
<th>Number of Cards Needed</th>
<th>Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>No. of regions under consideration</td>
<td>NR1</td>
<td>1</td>
<td>(I3)</td>
</tr>
<tr>
<td>2.</td>
<td>No. of system terminations, no. of data bases, and no. of distinctive cities under consideration</td>
<td>N1, N7, NCITY</td>
<td>1</td>
<td>(3I5)</td>
</tr>
<tr>
<td>3.</td>
<td>IDs for data bases</td>
<td>NBASE(N7)</td>
<td>1</td>
<td>(3(1X,A4))</td>
</tr>
<tr>
<td>4.</td>
<td>V-H Coordinates for cities</td>
<td>IVERT(NCITY), IHORZN(NCITY)</td>
<td>NCITY</td>
<td>(33X,15,2X,15)</td>
</tr>
<tr>
<td>5.</td>
<td>ID, name, city index, additional no. of terminals and traffic to/from each data base for each system termination</td>
<td>INDXPT(N1), NAMEST(H,N1), IADD(N1)</td>
<td>a. N1 if N7&lt;2</td>
<td>(A4,1X,3A6,A4, 12,14,4F10.2/6F10.2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAPADR(N1), TRAFD(N1,2,N7)</td>
<td>b. 2N1 if N7&gt;2</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>No. of rate structures</td>
<td>N2</td>
<td>1</td>
<td>(I3)</td>
</tr>
<tr>
<td>7.</td>
<td>Rate application matrix</td>
<td>IRATEJ(N2,N2)</td>
<td>N2</td>
<td>(10I2)</td>
</tr>
<tr>
<td>8.</td>
<td>Traffic density index and applicable rate structure for each city</td>
<td>IRAND(NCITY,2)</td>
<td>[NCITY/40]</td>
<td>(80I1)</td>
</tr>
<tr>
<td>9.</td>
<td>No. of applicable line types</td>
<td>N3</td>
<td>1</td>
<td>(I3)</td>
</tr>
<tr>
<td>Item No.</td>
<td>Item Description</td>
<td>Names of Internal Variables/Arrays</td>
<td>Number of Cards Needed</td>
<td>Formats</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>10.</td>
<td>Name, capacity, utilization limit, usage and duplexing mode for each line</td>
<td>LINAME(N3), LINCAP(N3), LINMIX(N3), IDUPLX(N3)</td>
<td>N3</td>
<td>(A6,1X,I6,1X, F3.2,2(1X,I1))</td>
</tr>
<tr>
<td>11.</td>
<td>No. of chargeable items</td>
<td>N4</td>
<td>1</td>
<td>(I3)</td>
</tr>
<tr>
<td>12.</td>
<td>Names of chargeable items</td>
<td>NAMEHW(N4)</td>
<td>1</td>
<td>(10(A6,1X))</td>
</tr>
<tr>
<td>13.</td>
<td>Installation and recurring costs for chargeable items WRT rate structure, traffic density and duplexing mode for each line type</td>
<td>AINSTC(N2,N3,N4,3,2,2), RECRC(N2,N3,N4,3,2,2)</td>
<td>2xN2xN3xN4x3x2</td>
<td>(2F9.2)</td>
</tr>
<tr>
<td>14.</td>
<td>Linear installation and recurring costs for lines WRT rate structure type, density, and duplex mode</td>
<td>IFLAG(N2,N3), ANSTLN(N2,N3,3,2,2), RECRLN(N2,N3,3,2,16)</td>
<td>a. N2x(2+N3x3x2)</td>
<td>a. (4F9.2/I1/10F8.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b. 2xN2(2+N3x3x2)</td>
<td>b. (4F9.2/I1/10F8.3/3x2) if non linear</td>
</tr>
<tr>
<td>15.</td>
<td>Action indices for regions</td>
<td>NRFG,NCODE for IACTN(NR1,2)</td>
<td>Variable</td>
<td>(2I2)</td>
</tr>
</tbody>
</table>
Table 3-1. Formats for Input Data (Continuation 2)

<table>
<thead>
<tr>
<th>Item No.</th>
<th>Item Description</th>
<th>Names of Internal Variables/Arrays</th>
<th>Number of Cards Needed</th>
<th>Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>No. of polling characters no. of NAK characters, no. of polling overhead characters, no of NAK overhead characters, message overhead characters, Modem turnaround time, and other delay for each line type</td>
<td>NPL(N3), NAK(N3), NPL(N3), NAKOH(N3), MOH(N3) TAMP(N3), TAPD(N3)</td>
<td>N3</td>
<td>(514,2F7.5)</td>
</tr>
<tr>
<td>17</td>
<td>No. of message types</td>
<td>NTYP</td>
<td>1</td>
<td>(I4)</td>
</tr>
<tr>
<td>18</td>
<td>Message name, input message length, output message length, input percentage and output percentage with priority 1 and 2</td>
<td>MSGNAM(NTYP), MSLIN(NTYP) MSGOV(NTYP), RATIOI(NTYP,2) RATIO(NTYP,2)</td>
<td>NTYP</td>
<td>(A6,2(I4,2F6.3))</td>
</tr>
<tr>
<td>19</td>
<td>Average CPU service time per transaction</td>
<td>CPUAVG</td>
<td>1</td>
<td>(F7.4)</td>
</tr>
<tr>
<td>20</td>
<td>Preloading system terminations and/or preselecting regional centers</td>
<td>NCODE, NSTATE, NREQQ</td>
<td>Variable</td>
<td>(I1,1X,A4,A5)</td>
</tr>
<tr>
<td>21</td>
<td>System Central</td>
<td>NSCC1</td>
<td>1</td>
<td>(A4)</td>
</tr>
<tr>
<td>Item No.</td>
<td>Item Description</td>
<td>Names of Internal Variables/Arrays</td>
<td>Number of Cards Needed</td>
<td>Formats</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------------------</td>
<td>------------------------------------</td>
<td>------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>22.</td>
<td>Total no. of messages per second and no. of requests made at the central switcher</td>
<td>XSAC, NREQSW</td>
<td>1</td>
<td>(F8.5, I3)</td>
</tr>
<tr>
<td>23.</td>
<td>Limit on no. of terminals on a multi-dropped line, response time requirement and no. of CPU processors for computer</td>
<td>NTERMS, TIMREQ, MPROC</td>
<td>1</td>
<td>(I3, F5.2, I2)</td>
</tr>
<tr>
<td>24.</td>
<td>Plot request</td>
<td>MPLLOT</td>
<td>1</td>
<td>(I3)</td>
</tr>
</tbody>
</table>
Following is a list of run options for the STACOM program.

(1) The user can preload system terminations to regions and/or preselect regional switching centers.

(2) The user can select certain regions for which the optimization process will be performed.

(3) Type of lines and chargeable items can be selectively chosen.

(4) The user can put a limit on the number of terminals on a multidrop line as described and can limit the average terminal response time.

(5) The number of central processor units in the central computer system can be 1, 2, or 4.

(6) The CalComp plot can be skipped.

3.3.3 Control Instruction and Sequences
3.3.3.1 Starting a Run.

3.3.3.1.1 Batch Mode. Following is a list of control statements required when running the STACOM program as a batch run:

```
@RUN run-ID, account-no., project-ID, SUP-time, pages/cards
@ASG,UP 100
@SYM,P PUNCH$,plotter-ID
@XQT file.STACOM
   (DATA)
@BRKPT 100
@FREE 100
@SYM 100,,printer-ID
@FIN
```

The RUN card gives the following information: designated run-ID, user's account number, project-ID, expected SUP-time usage and limited number of printer pages, and number of cards which may be generated from the run. Plotter-ID gives the logical ID of the CalComp pen plotter and file is the file which contains the absolute element of the STACOM program. Printer-ID gives the logical ID of the regular printer. DATA as shown is the input data described in Paragraph 3.2.1; the user should arrange the data in the same order. When all of these data items are in order and ready, the deck can be submitted to the operator at the computer site for processing.

3.3.3.1.2 Demand Mode. If program execution is to be performed via a demand terminal, the user can converse interactively with the program. The user may also run the program as a batch job by having all input data prepared and added after the @XQT statement.
Under the conversation mode, the user acts as a respondent who answers the requests for data made by the program. This mode of operation provides the user with some understanding of program progress. A user can very often terminate a run before a complete set of input data is given. This is possible because the user has some knowledge of the progress being made. This capability can prevent the user from an unnecessary waste of time. For example, if a run encounters a system which has more oversized distance data than allowed, a message from the program will be printed out on the terminal. The user will be alarmed by this fact and may decide to terminate the program run.

3.3.3.2 Run Progression. After receiving all of the required data, the program will perform all functions as designed and requested by the user whether a batch or a demand job has been executed. The program will perform formation of regions, selection of regional switching centers, formation of a regional star network and its optimization if requested, and finally, formation and optimization of an interregional network. All of the desired output data will go to the alternate file, 100.

3.3.3.3 Normal Termination. When a STACOM program run proceeds successfully and terminates normally, the normal file unit 6 will contain two lines of messages for each successful regional network optimization. These two lines are:

(1) TRYLNK has been accessed for xxxxx times.

(2) UPNETW has been accessed for xxxxx times.

The first message indicates the number of subroutine calls to TRYLNK that have been made during the process of searching for a better network. The second message indicates the number of optimization cycles which the run has gone through before the optimization process stops. After a normal termination, the user can direct the output file 100 to a printer device and the punch card file to a CalComp pen plotter if file PUNCH$ has been directed to an alternate file.

3.3.3.4 Aborting and Recovering a Run. When a run encounters trouble resulting from incorrect input data, the user can use the normal aborting procedure to terminate its execution if it is a demand job. A statement of @EX after interrupting the line communication by pressing the BREAK key will terminate a program execution at any time. On the other hand, the EXEC-8 may abort a program execution when certain serious violations occur during its execution, e.g., number of punch cards exceeding the limit on the run card.

If a program run has been interrupted due to system outage, no recovery of the run is possible.
3.3.4 Program Listing

A listing of the STACOM program elements is given in Appendix A.

3.4 SAMPLE RUN

To illustrate how STACOM can be run, a sample run is provided in the following subsections along with detailed explanations.

3.4.1 Run Stream

The following list of control statements shows the typical batch mode run stream used to execute the STACOM program.

```
@RUN JIL,J6C3YL,51928,20,90/1000
@ASG,UP 100
@SYM,P PUNCH$,G9PLTF
@QT LEE.STACOM
@ADD LEE.DATA
@BRKPT 100
@FREE 100
@SYM,U 100,T4
@FIN
```

The first control statement is a run request which specifies its run ID as JIL, identifies its account number as J6C3YL, assigns project ID as 51928, requests a maximum of 20 minutes of SUP-time and finally asks for a limit of 90 printer pages and 1,000 punch cards. The limits on SUP-time, number of printer pages and number of punch cards deserve some attention when making a run. If there is an underestimate in any of these three limits, the run may abort due to insufficient resource assignment.

The second statement is used to assign an alternate FORTRAN output file as required by the program. It is intended to be a one-day file.

Statement 3 requests the system to direct the punch card image file to the CalComp pen plotter with the name G9PLTF.

Statement 4 is a command for executing the STACOM program which is designated with the element name STACOM in file LEE.

The next statement asks the operating system to use the content of element DATA as its input data.

Statements 6 and 7 are used to close file 100 and catalog it for later use.

Statement 8 asks the operating system to send the printer file 100 to an on-site low speed printer with JD T4. The U option retains the FORTRAN print file after a copy is printed.
The last statement terminates the run with a request for a detailed description of run changes and run history. The number of pages in the print file and the number of punch-card images are part of the data given by the accounting subsystem when a run terminates.

When the same program is to be executed via a demand terminal, the content of element LEE.DATA can be divided into several individual elements plus certain key-in control statements. Essentially, however, the same amount of input data must be provided to succeed in running the program.

3.4.2 Input

As a specific example, Table 3-2 gives the list of data which have been used in analyzing the South Plains portion of Texas under the Council-of-Governments structure.

Encircled numbers have been written on the left hand side so that Table 3-2 and Table 3-1 are made compatible. Data associated with each encircled number in Table 3-2 corresponds to the data item with the same index in Table 3-1. Items 1 and 2 indicate that the run is concerned with 1 region case, a total of 25 system terminations, 4 data bases, and a total of 358 distinctive cities. Items 3 gives the IDs for locations of those four data bases, and item 4 lists the names of all 358 cities which have distinctive V-H coordinates (four digit integers). Since the number of data bases is greater than 2, two input cards are needed for each system termination; therefore a total of 50 cards are needed as listed under item 5. Since there is only one rate structure, one card is needed for rate application matrix (see items 6 and 7).

Item 8 shows the traffic density and rate application table for which 9 cards are required. Items 9 and 10 indicate that only 3 type of lines (with rates 1200 bps, 2400 bps, and 4800 bps) are considered; 0.7 is the line utilization limit for all of them. Three chargeable items are applicable as shown in items 11 and 12. Item 13 is somewhat complicated, the following explanation should enable the reader to understand it. These 108 data cards are divided into 3 groups with the first group given to the first line type, i.e., 1200 bps, and so on.
Table 3-2. Input Data for the Example Run

```
5192R*STACOM(01).INPUT/0777

    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56

   1. PALESTINE ANDERSON 8550 3750
   2. ROCKPORT ARANSAS 9455 3694
   3. ARCHER CITY ARCHER 8596 4410
   4. Jourdanton ATASCOSA 9342 4039
   5. BELVILLE BAILEY 8548 3710
   6. MULESHOE BAYLOR 8518 5157
   7. SEYMOUR BEACON 8437 4818
   8. BEEVILLE BEAUMONT 9378 3850
   9. BELTON HARRIS 8027 4010
  10. BELTON HARRIS 8032 4070
  11. ROCKPORT RAYNE 8032 4063
  12. KILLEEN BELL 8062 4063
  13. NOLANVILLE BELL 8032 4038
  14. TEMPLE BELL 8012 3992
  15. ALAMO HEIGHTS BEXAR 9225 4062
  16. FT. SAM HOUSTON BEXAR 9225 4062
  17. LEON VALLEY BEXAR 9223 4092
  18. SAN ANTONIO BEXAR 9225 4062
  19. UNIVERSAL CITY BEXAR 9167 4077
  20. CLIFTON BOSQUE 8640 4089
  21. MERIDIAN BOSQUE 8660 4112
  22. TEXARKANA BOYER 8111 3626
  23. ALVIN BRAZORIA 8966 3466
  24. ANGLETON BRAZORIA 8966 3466
  25. CLUTE BRAZORIA 8966 3466
  26. FREEPORT BRAZORIA 8966 3466
  27. LAKE JACKSON BRAZORIA 8966 3466
  28. PEARLAND BRAZORIA 8970 3506
  29. BRYAN BRAZOS 8927 3798
  30. COLLEGE STATION BRAZOS 8927 3798
  31. ALPINE BREWER 8664 4507
  32. FAULKNER HAUS 9655 3927
  33. BROWNWOOD BROWN 8797 4537
  34. CALDWELL BURLESON 8600 3834
  35. PORT LAVACA CALHOUN 8218 3466
  36. BROWNSVILLE CAMERON 9861 3506
  37. HARLINGEN CAMERON 9820 3663
  38. PORT ISABEL CAMERON 9807 3663
  39. SAN BENITO CAMERON 9826 3648
  40. LINDEN CASS 8217 3649
  41. Dumas CASTRO 8427 4100
  42. ANAHUAC CHAMBERS 8884 3418
  43. JACKSONVILLE CHEFROKE 8492 3700
  44. CHILDRESS CHILDRESS 8328 4745
  45. MORTON CHEROKEE 8622 4100
  46. ROBERT LEE COKE 8857 4608
  47. COLEMAN COLEMAN 8804 4013
  48. FRISCO COLLINS 8342 4056
  49. MCKINNEY COLLINS 8342 4056
  50. PLANO COLLINS 8342 4056
  51. WELLINGTON COLLINSW 8240 4776
```
Table 3-2. Input Data for the Example Run (Continuation 1)

<table>
<thead>
<tr>
<th>57</th>
<th>COLUMBUS</th>
<th>COLORADO</th>
<th>7032</th>
<th>3744</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
<td>NEW BRAUNFELS</td>
<td>COMAL</td>
<td>9145</td>
<td>4018</td>
</tr>
<tr>
<td>59</td>
<td>COMANCHE</td>
<td>COMANCHE</td>
<td>8735</td>
<td>4275</td>
</tr>
<tr>
<td>60</td>
<td>GAINESVILLE</td>
<td>COOKE</td>
<td>8269</td>
<td>4162</td>
</tr>
<tr>
<td>61</td>
<td>COPPERAS COVE</td>
<td>CORTELL</td>
<td>8844</td>
<td>4899</td>
</tr>
<tr>
<td>62</td>
<td>GATESVILLE</td>
<td>CRANE</td>
<td>8773</td>
<td>4090</td>
</tr>
<tr>
<td>63</td>
<td>CRANE</td>
<td>CRANETTE</td>
<td>9144</td>
<td>4692</td>
</tr>
<tr>
<td>64</td>
<td>OZONA</td>
<td>DALLAS</td>
<td>8129</td>
<td>4294</td>
</tr>
<tr>
<td>65</td>
<td>DALHART</td>
<td>DALLAS</td>
<td>8477</td>
<td>4087</td>
</tr>
<tr>
<td>66</td>
<td>ADDISON</td>
<td>DALLAS</td>
<td>8486</td>
<td>4078</td>
</tr>
<tr>
<td>67</td>
<td>CEDAR HILL</td>
<td>DALLAS</td>
<td>8436</td>
<td>4034</td>
</tr>
<tr>
<td>68</td>
<td>DALLAS</td>
<td>DALLAS</td>
<td>8478</td>
<td>4030</td>
</tr>
<tr>
<td>69</td>
<td>DESOTO</td>
<td>DALLAS</td>
<td>8469</td>
<td>4044</td>
</tr>
<tr>
<td>70</td>
<td>DUNCANVILLE</td>
<td>DALLAS</td>
<td>8414</td>
<td>4062</td>
</tr>
<tr>
<td>71</td>
<td>FARMERS BRANCH</td>
<td>DALLAS</td>
<td>8400</td>
<td>4018</td>
</tr>
<tr>
<td>72</td>
<td>GARLAND</td>
<td>DALLAS</td>
<td>8458</td>
<td>4066</td>
</tr>
<tr>
<td>73</td>
<td>GRAND PRAIRIE</td>
<td>DALLAS</td>
<td>8436</td>
<td>4034</td>
</tr>
<tr>
<td>74</td>
<td>HIGHLAND PARK</td>
<td>DALLAS</td>
<td>8440</td>
<td>4064</td>
</tr>
<tr>
<td>75</td>
<td>IRVING</td>
<td>DALLAS</td>
<td>8470</td>
<td>4013</td>
</tr>
<tr>
<td>76</td>
<td>LANCASTER</td>
<td>DALLAS</td>
<td>8426</td>
<td>4000</td>
</tr>
<tr>
<td>77</td>
<td>MESQUITE</td>
<td>DALLAS</td>
<td>8369</td>
<td>4035</td>
</tr>
<tr>
<td>78</td>
<td>RICHARDSON</td>
<td>DALLAS</td>
<td>8447</td>
<td>3940</td>
</tr>
<tr>
<td>79</td>
<td>SEAGOVILLE</td>
<td>DALLAS</td>
<td>8436</td>
<td>4034</td>
</tr>
<tr>
<td>80</td>
<td>UNIVERSITY PARK</td>
<td>DAWSON</td>
<td>8779</td>
<td>4919</td>
</tr>
<tr>
<td>81</td>
<td>LAVEZA</td>
<td>DEAF SMTH</td>
<td>8378</td>
<td>5143</td>
</tr>
<tr>
<td>82</td>
<td>HEREFORD</td>
<td>DENTON</td>
<td>8372</td>
<td>4127</td>
</tr>
<tr>
<td>83</td>
<td>DENTON</td>
<td>DFNTON</td>
<td>8386</td>
<td>4095</td>
</tr>
<tr>
<td>84</td>
<td>LEWISVILLE</td>
<td>DEWITT</td>
<td>9209</td>
<td>3923</td>
</tr>
<tr>
<td>85</td>
<td>CIFRO</td>
<td>DICKENS</td>
<td>8560</td>
<td>4744</td>
</tr>
<tr>
<td>86</td>
<td>SPIRISVILLE</td>
<td>DUVAL</td>
<td>9542</td>
<td>3988</td>
</tr>
<tr>
<td>87</td>
<td>SAN DIEGO</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>88</td>
<td>EASTLAND</td>
<td>ECTOR</td>
<td>8982</td>
<td>4630</td>
</tr>
<tr>
<td>89</td>
<td>ODessa</td>
<td>ELLIS</td>
<td>8514</td>
<td>3970</td>
</tr>
<tr>
<td>90</td>
<td>ENNIS</td>
<td>ELLIS</td>
<td>8517</td>
<td>4011</td>
</tr>
<tr>
<td>91</td>
<td>WAYAHACHE</td>
<td>EL PASO</td>
<td>9231</td>
<td>5655</td>
</tr>
<tr>
<td>92</td>
<td>EL PASO</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>93</td>
<td>STEPHENVILLE</td>
<td>EL PASO</td>
<td>9231</td>
<td>5655</td>
</tr>
<tr>
<td>94</td>
<td>MARLIN</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>95</td>
<td>ENNIS</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>96</td>
<td>BONHAM</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>97</td>
<td>FLOYDADA</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>98</td>
<td>FAIRFIELD</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>99</td>
<td>PEARSSAL</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>100</td>
<td>RICHMOND</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>101</td>
<td>ROSENBERG</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>102</td>
<td>SEMINOLE</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>103</td>
<td>FRIENDSWOOD</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>104</td>
<td>GALVESTON</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>105</td>
<td>HITCHCOCK</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>106</td>
<td>LA MARQUE</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>107</td>
<td>LEAGUE CITY</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>108</td>
<td>TEXAS CITY</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>109</td>
<td>POST</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>110</td>
<td>FREDERICKSBURG</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>111</td>
<td>GOLIAD</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>112</td>
<td>GONZALES</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>113</td>
<td>潘帕</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run (Continuation 2)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>114</td>
<td>DEMISON</td>
<td>GRAYSON</td>
<td>825</td>
<td>4069</td>
</tr>
<tr>
<td>115</td>
<td>SHERMAN</td>
<td>GRAYSON</td>
<td>835</td>
<td>4072</td>
</tr>
<tr>
<td>116</td>
<td>GLANEWATER</td>
<td>GREGG</td>
<td>834</td>
<td>3669</td>
</tr>
<tr>
<td>117</td>
<td>KILGORE</td>
<td>GREGG</td>
<td>837</td>
<td>3674</td>
</tr>
<tr>
<td>118</td>
<td>LONGVIEW</td>
<td>GREGG</td>
<td>834</td>
<td>3660</td>
</tr>
<tr>
<td>119</td>
<td>NAVASOTA</td>
<td>GRIMES</td>
<td>8665</td>
<td>3715</td>
</tr>
<tr>
<td>120</td>
<td>SEGUIN</td>
<td>GUADALUPE</td>
<td>9161</td>
<td>3981</td>
</tr>
<tr>
<td>121</td>
<td>PLAINVIEW</td>
<td>HALE</td>
<td>8465</td>
<td>4001</td>
</tr>
<tr>
<td>122</td>
<td>HAMILTON</td>
<td>HAMILTON</td>
<td>8744</td>
<td>4177</td>
</tr>
<tr>
<td>123</td>
<td>SPEARMAN</td>
<td>HANSFORD</td>
<td>8026</td>
<td>5637</td>
</tr>
<tr>
<td>124</td>
<td>QUANAH</td>
<td>HARDMAN</td>
<td>834</td>
<td>4654</td>
</tr>
<tr>
<td>125</td>
<td>KOUNTZE</td>
<td>HARRIN</td>
<td>8735</td>
<td>3905</td>
</tr>
<tr>
<td>126</td>
<td>SILSBEE</td>
<td>HARRIN</td>
<td>8730</td>
<td>3900</td>
</tr>
<tr>
<td>127</td>
<td>BAYTOWN</td>
<td>HARRIS</td>
<td>8916</td>
<td>3966</td>
</tr>
<tr>
<td>128</td>
<td>BELLAIRE</td>
<td>HARRIS</td>
<td>8938</td>
<td>3936</td>
</tr>
<tr>
<td>129</td>
<td>DEEP PARK</td>
<td>HARRIS</td>
<td>8909</td>
<td>3491</td>
</tr>
<tr>
<td>130</td>
<td>GALENA PARK</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>131</td>
<td>HOUSTON</td>
<td>HARRIS</td>
<td>8881</td>
<td>3540</td>
</tr>
<tr>
<td>132</td>
<td>HUMBLE</td>
<td>HARRIS</td>
<td>8938</td>
<td>3556</td>
</tr>
<tr>
<td>133</td>
<td>JARINTO CITY</td>
<td>HARRIS</td>
<td>8925</td>
<td>3581</td>
</tr>
<tr>
<td>134</td>
<td>JERSEY VILLAGE</td>
<td>HARRIS</td>
<td>8965</td>
<td>3518</td>
</tr>
<tr>
<td>135</td>
<td>KATY</td>
<td>HARRIS</td>
<td>8829</td>
<td>3470</td>
</tr>
<tr>
<td>136</td>
<td>LA PORTE</td>
<td>HARRIS</td>
<td>8930</td>
<td>3536</td>
</tr>
<tr>
<td>137</td>
<td>PASADENA</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>138</td>
<td>SEABROOK</td>
<td>HARRIS</td>
<td>8945</td>
<td>3679</td>
</tr>
<tr>
<td>139</td>
<td>SOUTH HOUSTON</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>140</td>
<td>SOUTH SIDE PLACE</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>141</td>
<td>SPRING VALLEY</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>142</td>
<td>TOMBALL</td>
<td>HARRIS</td>
<td>8889</td>
<td>3609</td>
</tr>
<tr>
<td>143</td>
<td>VILLAGE</td>
<td>HARRIS</td>
<td>8934</td>
<td>3536</td>
</tr>
<tr>
<td>144</td>
<td>WEBSTER</td>
<td>HARRIS</td>
<td>8928</td>
<td>3648</td>
</tr>
<tr>
<td>145</td>
<td>WEST UNIV PL</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>146</td>
<td>MARSHALL</td>
<td>HARRIS</td>
<td>8311</td>
<td>3502</td>
</tr>
<tr>
<td>147</td>
<td>HASKELL</td>
<td>HASKELL</td>
<td>8555</td>
<td>4567</td>
</tr>
<tr>
<td>148</td>
<td>SAN MARCOS</td>
<td>HAYS</td>
<td>9006</td>
<td>4001</td>
</tr>
<tr>
<td>149</td>
<td>CANADIAN</td>
<td>HEMPHILL</td>
<td>8036</td>
<td>4882</td>
</tr>
<tr>
<td>150</td>
<td>ATHENS</td>
<td>HENDERSON</td>
<td>8444</td>
<td>3226</td>
</tr>
<tr>
<td>151</td>
<td>DONNA</td>
<td>HIDALGO</td>
<td>9849</td>
<td>3728</td>
</tr>
<tr>
<td>152</td>
<td>EDINBURG</td>
<td>HIDALGO</td>
<td>9840</td>
<td>3745</td>
</tr>
<tr>
<td>153</td>
<td>Hidalgo</td>
<td>HIDALGO</td>
<td>9946</td>
<td>3764</td>
</tr>
<tr>
<td>154</td>
<td>MCALLEN</td>
<td>HIDALGO</td>
<td>9946</td>
<td>3764</td>
</tr>
<tr>
<td>155</td>
<td>MERCEDES</td>
<td>HIDALGO</td>
<td>9945</td>
<td>3701</td>
</tr>
<tr>
<td>156</td>
<td>MISSION</td>
<td>HIDALGO</td>
<td>9941</td>
<td>3781</td>
</tr>
<tr>
<td>157</td>
<td>PHARR</td>
<td>HIDALGO</td>
<td>9948</td>
<td>3758</td>
</tr>
<tr>
<td>158</td>
<td>WESLACO</td>
<td>HIDALGO</td>
<td>9847</td>
<td>3716</td>
</tr>
<tr>
<td>159</td>
<td>HILLSBORO</td>
<td>HILL</td>
<td>8612</td>
<td>4026</td>
</tr>
<tr>
<td>160</td>
<td>LEVELLAND</td>
<td>HOCKLEY</td>
<td>8629</td>
<td>5033</td>
</tr>
<tr>
<td>161</td>
<td>SULPHUR SPRINGS</td>
<td>HOPKINS</td>
<td>8241</td>
<td>3861</td>
</tr>
<tr>
<td>162</td>
<td>BIG SPRING</td>
<td>HOWARD</td>
<td>8847</td>
<td>4000</td>
</tr>
<tr>
<td>163</td>
<td>COMMERCE</td>
<td>HUNT</td>
<td>8240</td>
<td>3921</td>
</tr>
<tr>
<td>164</td>
<td>GRENVILLE</td>
<td>HUNT</td>
<td>8317</td>
<td>3949</td>
</tr>
<tr>
<td>165</td>
<td>BORGER</td>
<td>HUTCHINSON</td>
<td>8146</td>
<td>5033</td>
</tr>
<tr>
<td>166</td>
<td>JACKSBORO</td>
<td>JACKSON</td>
<td>9442</td>
<td>4304</td>
</tr>
<tr>
<td>167</td>
<td>EDNA</td>
<td>JACKSON</td>
<td>9186</td>
<td>3694</td>
</tr>
<tr>
<td>168</td>
<td>BEAUMONT</td>
<td>JEFFERSON</td>
<td>8777</td>
<td>3444</td>
</tr>
<tr>
<td>169</td>
<td>NEDERLAND</td>
<td>JEFFERSON</td>
<td>8749</td>
<td>3316</td>
</tr>
<tr>
<td>170</td>
<td>PORT ARTHUR</td>
<td>JEFFERSON</td>
<td>8406</td>
<td>3298</td>
</tr>
<tr>
<td></td>
<td>ALICE</td>
<td>JIM WILLS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
<td>-----------</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>171</td>
<td>BURLESON</td>
<td>JOHNSON</td>
<td>8522</td>
<td>4103</td>
</tr>
<tr>
<td>173</td>
<td>CLARKE</td>
<td>JOHNSON</td>
<td>8563</td>
<td>4152</td>
</tr>
<tr>
<td>174</td>
<td>ANSON</td>
<td>JONES</td>
<td>8647</td>
<td>4563</td>
</tr>
<tr>
<td>175</td>
<td>STAMFORD</td>
<td>JONES</td>
<td>8603</td>
<td>4569</td>
</tr>
<tr>
<td>176</td>
<td>KARNES CITY</td>
<td>KARNES</td>
<td>9246</td>
<td>3974</td>
</tr>
<tr>
<td>177</td>
<td>KAUFMAN</td>
<td>KAUFMAN</td>
<td>8442</td>
<td>3924</td>
</tr>
<tr>
<td>178</td>
<td>TERRELL</td>
<td>KAUFMAN</td>
<td>8410</td>
<td>3943</td>
</tr>
<tr>
<td>179</td>
<td>BURTON</td>
<td>KENDALL</td>
<td>9168</td>
<td>4123</td>
</tr>
<tr>
<td>180</td>
<td>BOWEN</td>
<td>KENT</td>
<td>8589</td>
<td>4718</td>
</tr>
<tr>
<td>181</td>
<td>KEPPVILLE</td>
<td>KERR</td>
<td>9143</td>
<td>4226</td>
</tr>
<tr>
<td>182</td>
<td>JUNCTION</td>
<td>KIMBLE</td>
<td>9097</td>
<td>4373</td>
</tr>
<tr>
<td>183</td>
<td>KINGSVILLE</td>
<td>KLEBERG</td>
<td>9566</td>
<td>4801</td>
</tr>
<tr>
<td>184</td>
<td>BENJAMIN</td>
<td>KNOX</td>
<td>8472</td>
<td>4619</td>
</tr>
<tr>
<td>185</td>
<td>PARTS</td>
<td>LAMAR</td>
<td>8173</td>
<td>3997</td>
</tr>
<tr>
<td>186</td>
<td>LITTLEFIELD</td>
<td>LAMB</td>
<td>8558</td>
<td>5649</td>
</tr>
<tr>
<td>187</td>
<td>OLTON</td>
<td>LAMAS</td>
<td>8490</td>
<td>4054</td>
</tr>
<tr>
<td>188</td>
<td>LAMPASAS</td>
<td>LAMPASAS</td>
<td>8347</td>
<td>4137</td>
</tr>
<tr>
<td>189</td>
<td>HALLETTSVILLE</td>
<td>LAVACA</td>
<td>9114</td>
<td>3749</td>
</tr>
<tr>
<td>190</td>
<td>YORK</td>
<td>LAVACA</td>
<td>9157</td>
<td>3914</td>
</tr>
<tr>
<td>191</td>
<td>CENTERVILLE</td>
<td>LEON</td>
<td>8682</td>
<td>3768</td>
</tr>
<tr>
<td>192</td>
<td>CLEVELAND</td>
<td>LIBERTY</td>
<td>8801</td>
<td>3540</td>
</tr>
<tr>
<td>193</td>
<td>LIBERTY</td>
<td>LIBERTY</td>
<td>8335</td>
<td>3463</td>
</tr>
<tr>
<td>194</td>
<td>MEYET</td>
<td>LIMESTONE</td>
<td>8635</td>
<td>3494</td>
</tr>
<tr>
<td>195</td>
<td>GEORGE WEST</td>
<td>LIVE OAK</td>
<td>9419</td>
<td>3918</td>
</tr>
<tr>
<td>196</td>
<td>LUBBOCK</td>
<td>LUBBOCK</td>
<td>8590</td>
<td>4662</td>
</tr>
<tr>
<td>197</td>
<td>SLATON</td>
<td>LUBBOCK</td>
<td>8616</td>
<td>4916</td>
</tr>
<tr>
<td>198</td>
<td>TAHOKE</td>
<td>LYNN</td>
<td>8680</td>
<td>4924</td>
</tr>
<tr>
<td>199</td>
<td>MADISONVILLE</td>
<td>MADISON</td>
<td>8740</td>
<td>3733</td>
</tr>
<tr>
<td>200</td>
<td>JEFFERSON</td>
<td>MARION</td>
<td>8267</td>
<td>3614</td>
</tr>
<tr>
<td>201</td>
<td>BAY CITY</td>
<td>MATAGORDA</td>
<td>9135</td>
<td>3578</td>
</tr>
<tr>
<td>202</td>
<td>EAGLE PASS</td>
<td>MAVERICK</td>
<td>9505</td>
<td>4370</td>
</tr>
<tr>
<td>203</td>
<td>BRADY</td>
<td>MCCULLOCH</td>
<td>8938</td>
<td>4344</td>
</tr>
<tr>
<td>204</td>
<td>BELLMEAD</td>
<td>MCLENNAN</td>
<td>8706</td>
<td>3903</td>
</tr>
<tr>
<td>205</td>
<td>BEVERLY HILLS</td>
<td>MCLENNAN</td>
<td>8706</td>
<td>3903</td>
</tr>
<tr>
<td>206</td>
<td>KACO</td>
<td>MCLENNAN</td>
<td>8706</td>
<td>3903</td>
</tr>
<tr>
<td>207</td>
<td>WOODWAY</td>
<td>MCLENNAN</td>
<td>8706</td>
<td>3903</td>
</tr>
<tr>
<td>208</td>
<td>HONDO</td>
<td>MEDINA</td>
<td>9285</td>
<td>4174</td>
</tr>
<tr>
<td>209</td>
<td>MENARD</td>
<td>MENARD</td>
<td>9011</td>
<td>4417</td>
</tr>
<tr>
<td>210</td>
<td>MITLAND</td>
<td>MIDLAND</td>
<td>8934</td>
<td>4484</td>
</tr>
<tr>
<td>211</td>
<td>CAMPRON</td>
<td>MILAN</td>
<td>8835</td>
<td>3910</td>
</tr>
<tr>
<td>212</td>
<td>ROCKDALE</td>
<td>MILAM</td>
<td>8877</td>
<td>3898</td>
</tr>
<tr>
<td>213</td>
<td>COLORADO CITY</td>
<td>MITCHELL</td>
<td>8718</td>
<td>4706</td>
</tr>
<tr>
<td>214</td>
<td>BOWIE</td>
<td>MONTAIGNE</td>
<td>8351</td>
<td>4275</td>
</tr>
<tr>
<td>215</td>
<td>MONTAGUE</td>
<td>MONTAGUE</td>
<td>8323</td>
<td>4241</td>
</tr>
<tr>
<td>216</td>
<td>CONROE</td>
<td>MONTGOMARY</td>
<td>8832</td>
<td>4605</td>
</tr>
<tr>
<td>217</td>
<td>DUNAS</td>
<td>MOORE</td>
<td>8141</td>
<td>5144</td>
</tr>
<tr>
<td>218</td>
<td>DANGERFIELD</td>
<td>MORRIS</td>
<td>8240</td>
<td>3704</td>
</tr>
<tr>
<td>219</td>
<td>NACOGDOCHES</td>
<td>NACOGDOCHES</td>
<td>8515</td>
<td>3569</td>
</tr>
<tr>
<td>220</td>
<td>CONSICANA</td>
<td>NAVARRO</td>
<td>8593</td>
<td>3921</td>
</tr>
<tr>
<td>221</td>
<td>SWEETWATER</td>
<td>NOLAN</td>
<td>8737</td>
<td>4632</td>
</tr>
<tr>
<td>222</td>
<td>CORPUS CHRISTI</td>
<td>NUECES</td>
<td>9475</td>
<td>3729</td>
</tr>
<tr>
<td>223</td>
<td>ROSTOWN</td>
<td>NUECES</td>
<td>9496</td>
<td>3786</td>
</tr>
<tr>
<td>224</td>
<td>PEPPERTON</td>
<td>OCHILTREE</td>
<td>7962</td>
<td>4917</td>
</tr>
<tr>
<td>225</td>
<td>VEGA</td>
<td>OLDHAM</td>
<td>8292</td>
<td>5177</td>
</tr>
<tr>
<td>226</td>
<td>ORANGE</td>
<td>ORANGE</td>
<td>8746</td>
<td>3291</td>
</tr>
<tr>
<td>227</td>
<td>MINERAL WELLS</td>
<td>PARKER</td>
<td>8520</td>
<td>4214</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run (Continuation 4)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>228</td>
<td>CARTHAGE</td>
<td>PANOLA</td>
<td>8385</td>
</tr>
<tr>
<td>229</td>
<td>WEATHERFORD</td>
<td>PARKER</td>
<td>8580</td>
</tr>
<tr>
<td>230</td>
<td>FARWELL</td>
<td>FANNIN</td>
<td>8203</td>
</tr>
<tr>
<td>231</td>
<td>FRONtera</td>
<td>FARMER</td>
<td>8459</td>
</tr>
<tr>
<td>232</td>
<td>FORT STOCKTON</td>
<td>PECOS</td>
<td>9207</td>
</tr>
<tr>
<td>233</td>
<td>AMARILLO</td>
<td>POTTER</td>
<td>8266</td>
</tr>
<tr>
<td>234</td>
<td>CANYON</td>
<td>RANDALL</td>
<td>8317</td>
</tr>
<tr>
<td>235</td>
<td>CLARKSVILLE</td>
<td>RED RIVER</td>
<td>8147</td>
</tr>
<tr>
<td>236</td>
<td>PECOS</td>
<td>REEVES</td>
<td>9136</td>
</tr>
<tr>
<td>237</td>
<td>BIG LAKE</td>
<td>REGAN</td>
<td>9062</td>
</tr>
<tr>
<td>238</td>
<td>HEARNE</td>
<td>ROBERTSON</td>
<td>8802</td>
</tr>
<tr>
<td>239</td>
<td>ROCKWALL</td>
<td>ROCKWALL</td>
<td>8384</td>
</tr>
<tr>
<td>240</td>
<td>BALLINGER</td>
<td>RUNNELS</td>
<td>8855</td>
</tr>
<tr>
<td>241</td>
<td>HENDERSON</td>
<td>RUSK</td>
<td>8420</td>
</tr>
<tr>
<td>242</td>
<td>ARANSAS PASS</td>
<td>SAN PTOCO</td>
<td>9437</td>
</tr>
<tr>
<td>243</td>
<td>GREGORY</td>
<td>SAN PTOCO</td>
<td>9445</td>
</tr>
<tr>
<td>244</td>
<td>INGLESDIE</td>
<td>SAN PTOCO</td>
<td>9447</td>
</tr>
<tr>
<td>245</td>
<td>PONTIAD</td>
<td>SAN PTOCO</td>
<td>9445</td>
</tr>
<tr>
<td>246</td>
<td>SINTON</td>
<td>SAN PTOCO</td>
<td>9436</td>
</tr>
<tr>
<td>247</td>
<td>ELDELADO</td>
<td>SCHLEICHR</td>
<td>9076</td>
</tr>
<tr>
<td>248</td>
<td>SNYDER</td>
<td>SCURRY</td>
<td>8718</td>
</tr>
<tr>
<td>249</td>
<td>STRATFORD</td>
<td>SHFARMAN</td>
<td>8049</td>
</tr>
<tr>
<td>250</td>
<td>TYLER</td>
<td>SMITH</td>
<td>8817</td>
</tr>
<tr>
<td>251</td>
<td>BREEKNERIDGE</td>
<td>STEPHENS</td>
<td>8562</td>
</tr>
<tr>
<td>252</td>
<td>STERLING CITY</td>
<td>STEFLENG</td>
<td>8900</td>
</tr>
<tr>
<td>253</td>
<td>ASPERMONT</td>
<td>ST. NEVALL</td>
<td>8566</td>
</tr>
<tr>
<td>254</td>
<td>TULIA</td>
<td>SWISHER</td>
<td>8397</td>
</tr>
<tr>
<td>255</td>
<td>ARLINGTON</td>
<td>TARRANT</td>
<td>8472</td>
</tr>
<tr>
<td>256</td>
<td>BEDFORD</td>
<td>TARRANT</td>
<td>8447</td>
</tr>
<tr>
<td>257</td>
<td>COLLEYVILLE</td>
<td>TARRANT</td>
<td>8447</td>
</tr>
<tr>
<td>258</td>
<td>CROWLEY</td>
<td>TARRANT</td>
<td>8518</td>
</tr>
<tr>
<td>259</td>
<td>EULESS</td>
<td>TARRANT</td>
<td>8447</td>
</tr>
<tr>
<td>260</td>
<td>FORRETT HILL</td>
<td>TARRANT</td>
<td>8479</td>
</tr>
<tr>
<td>261</td>
<td>FORT WORTH</td>
<td>TARRANT</td>
<td>8479</td>
</tr>
<tr>
<td>262</td>
<td>GRAPEVINE</td>
<td>TARRANT</td>
<td>8425</td>
</tr>
<tr>
<td>263</td>
<td>HALTON CITY</td>
<td>TARRANT</td>
<td>8470</td>
</tr>
<tr>
<td>264</td>
<td>HURST</td>
<td>TARRANT</td>
<td>8447</td>
</tr>
<tr>
<td>265</td>
<td>LAKE NORTH</td>
<td>TARRANT</td>
<td>8471</td>
</tr>
<tr>
<td>266</td>
<td>NO RICHLAND HLS</td>
<td>TARRANT</td>
<td>8447</td>
</tr>
<tr>
<td>267</td>
<td>RICHLAND HILLS</td>
<td>TARRANT</td>
<td>8479</td>
</tr>
<tr>
<td>268</td>
<td>SOUTH LAKE</td>
<td>TARRANT</td>
<td>8425</td>
</tr>
<tr>
<td>269</td>
<td>WHITE SETTL'MNT</td>
<td>TARRANT</td>
<td>8485</td>
</tr>
<tr>
<td>270</td>
<td>ARLENE</td>
<td>TAYLOR</td>
<td>8699</td>
</tr>
<tr>
<td>271</td>
<td>SANDERSON</td>
<td>TERRELL</td>
<td>9333</td>
</tr>
<tr>
<td>272</td>
<td>BROWNFIELD</td>
<td>TERRY</td>
<td>8705</td>
</tr>
<tr>
<td>273</td>
<td>MT PLEASANT</td>
<td>TITUS</td>
<td>8234</td>
</tr>
<tr>
<td>274</td>
<td>SAN ANGELO</td>
<td>TOM GREEN</td>
<td>8944</td>
</tr>
<tr>
<td>275</td>
<td>AUSTIN</td>
<td>TRAVIS</td>
<td>9005</td>
</tr>
<tr>
<td>276</td>
<td>GILMER</td>
<td>UPSHUR</td>
<td>8317</td>
</tr>
<tr>
<td>277</td>
<td>RANKIN</td>
<td>UPTON</td>
<td>9084</td>
</tr>
<tr>
<td>278</td>
<td>UVALDE</td>
<td>UVALDE</td>
<td>9357</td>
</tr>
<tr>
<td>279</td>
<td>DEL RIO</td>
<td>VAL VERDE</td>
<td>9349</td>
</tr>
<tr>
<td>280</td>
<td>CANTON</td>
<td>VAN ZANDT</td>
<td>8414</td>
</tr>
<tr>
<td>281</td>
<td>VICTORIA</td>
<td>VICTORIA</td>
<td>9245</td>
</tr>
<tr>
<td>282</td>
<td>HUNTSVILLE</td>
<td>WALLER</td>
<td>8732</td>
</tr>
<tr>
<td>283</td>
<td>HEMPSTEAD</td>
<td>WALLER</td>
<td>8823</td>
</tr>
<tr>
<td>284</td>
<td>MONAHANS</td>
<td>WARD</td>
<td>9066</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run (Continuation 5)

<table>
<thead>
<tr>
<th>Location</th>
<th>County</th>
<th>Population</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brenham</td>
<td>Washington</td>
<td>8942</td>
<td>3759</td>
</tr>
<tr>
<td>Laredo</td>
<td>Webb</td>
<td>9681</td>
<td>4099</td>
</tr>
<tr>
<td>Wharton</td>
<td>Wharton</td>
<td>9115</td>
<td>3649</td>
</tr>
<tr>
<td>Quickly</td>
<td>Wheeler</td>
<td>8170</td>
<td>4408</td>
</tr>
<tr>
<td>Waco</td>
<td>Houston</td>
<td>8240</td>
<td>4440</td>
</tr>
<tr>
<td>Wichita Falls</td>
<td>Wichita</td>
<td>8326</td>
<td>4413</td>
</tr>
<tr>
<td>Vernon</td>
<td>Wilbarger</td>
<td>8326</td>
<td>4467</td>
</tr>
<tr>
<td>Raymondville</td>
<td>Willacy</td>
<td>9761</td>
<td>3703</td>
</tr>
<tr>
<td>Floresville</td>
<td>Wilson</td>
<td>9241</td>
<td>3970</td>
</tr>
<tr>
<td>Kermit</td>
<td>Winkler</td>
<td>9024</td>
<td>5060</td>
</tr>
<tr>
<td>Decatur</td>
<td>Wise</td>
<td>8399</td>
<td>4265</td>
</tr>
<tr>
<td>Winnie</td>
<td>Wood</td>
<td>8295</td>
<td>3794</td>
</tr>
<tr>
<td>Denver City</td>
<td>Yoakum</td>
<td>8781</td>
<td>5058</td>
</tr>
<tr>
<td>Graham</td>
<td>Young</td>
<td>8492</td>
<td>4365</td>
</tr>
<tr>
<td>Olney</td>
<td>Young</td>
<td>8450</td>
<td>4414</td>
</tr>
<tr>
<td>Crystal City</td>
<td>Zavala</td>
<td>9466</td>
<td>4246</td>
</tr>
<tr>
<td>Aransas Pass</td>
<td>Aransas</td>
<td>9437</td>
<td>3700</td>
</tr>
<tr>
<td>Pearsall</td>
<td>Atascosa</td>
<td>9320</td>
<td>4027</td>
</tr>
<tr>
<td>Banera</td>
<td>Bandera</td>
<td>9255</td>
<td>4190</td>
</tr>
<tr>
<td>Bastrop</td>
<td>Bastrop</td>
<td>9007</td>
<td>3900</td>
</tr>
<tr>
<td>Marble Falls</td>
<td>Burnet</td>
<td>8960</td>
<td>4115</td>
</tr>
<tr>
<td>Lockhart</td>
<td>Caldwell</td>
<td>9077</td>
<td>3954</td>
</tr>
<tr>
<td>Luling</td>
<td>Caldwell</td>
<td>9117</td>
<td>3933</td>
</tr>
<tr>
<td>Baird</td>
<td>Callahan</td>
<td>8688</td>
<td>4450</td>
</tr>
<tr>
<td>Pittsburg</td>
<td>Camp</td>
<td>8264</td>
<td>3742</td>
</tr>
<tr>
<td>Panhandle</td>
<td>Carson</td>
<td>8210</td>
<td>5004</td>
</tr>
<tr>
<td>Atlanta</td>
<td>Cass</td>
<td>8182</td>
<td>3618</td>
</tr>
<tr>
<td>Rusk</td>
<td>Cherokee</td>
<td>8515</td>
<td>3672</td>
</tr>
<tr>
<td>Henrietta</td>
<td>Clay</td>
<td>8333</td>
<td>4354</td>
</tr>
<tr>
<td>Crosbyton</td>
<td>Crosby</td>
<td>8545</td>
<td>4462</td>
</tr>
<tr>
<td>Carrollton</td>
<td>Dallas</td>
<td>8410</td>
<td>4066</td>
</tr>
<tr>
<td>Cooper</td>
<td>Delta</td>
<td>8241</td>
<td>3866</td>
</tr>
<tr>
<td>Carrizo Springs</td>
<td>Dimmit</td>
<td>9580</td>
<td>4200</td>
</tr>
<tr>
<td>Cicero</td>
<td>Eastland</td>
<td>8662</td>
<td>4377</td>
</tr>
<tr>
<td>La Grange</td>
<td>Fayette</td>
<td>9016</td>
<td>3813</td>
</tr>
<tr>
<td>Mt Vernon</td>
<td>Franklin</td>
<td>8246</td>
<td>3801</td>
</tr>
<tr>
<td>Anderson</td>
<td>Grimes</td>
<td>8876</td>
<td>3708</td>
</tr>
<tr>
<td>Memphis</td>
<td>Hall</td>
<td>8297</td>
<td>4421</td>
</tr>
<tr>
<td>Alamo</td>
<td>Hidalgo</td>
<td>9854</td>
<td>3754</td>
</tr>
<tr>
<td>Granbury</td>
<td>Hood</td>
<td>8572</td>
<td>4178</td>
</tr>
<tr>
<td>Crockett</td>
<td>Houston</td>
<td>8634</td>
<td>3645</td>
</tr>
<tr>
<td>Winnie</td>
<td>Hutchinson</td>
<td>8117</td>
<td>5054</td>
</tr>
<tr>
<td>Jasper</td>
<td>Jasper</td>
<td>8643</td>
<td>3390</td>
</tr>
<tr>
<td>Groves</td>
<td>Jefferson</td>
<td>8789</td>
<td>3316</td>
</tr>
<tr>
<td>Port Neches</td>
<td>Jefferson</td>
<td>8789</td>
<td>3316</td>
</tr>
<tr>
<td>Cotulla</td>
<td>LaSalle</td>
<td>9476</td>
<td>4120</td>
</tr>
<tr>
<td>Giddings</td>
<td>Lef</td>
<td>8968</td>
<td>3848</td>
</tr>
<tr>
<td>Groesbeck</td>
<td>Limestone</td>
<td>8671</td>
<td>3986</td>
</tr>
<tr>
<td>Llano</td>
<td>Llano</td>
<td>8970</td>
<td>4190</td>
</tr>
<tr>
<td>Newton</td>
<td>Newton</td>
<td>8600</td>
<td>3353</td>
</tr>
<tr>
<td>Bridge City</td>
<td>Orangef</td>
<td>8774</td>
<td>3295</td>
</tr>
<tr>
<td>Vinor</td>
<td>Orangef</td>
<td>8761</td>
<td>3334</td>
</tr>
<tr>
<td>Palo Pinto</td>
<td>Plo Pinto</td>
<td>8541</td>
<td>4291</td>
</tr>
<tr>
<td>Livingston</td>
<td>Polk</td>
<td>8716</td>
<td>3543</td>
</tr>
<tr>
<td>Refugio</td>
<td>Refugio</td>
<td>9345</td>
<td>3797</td>
</tr>
<tr>
<td>Franklin</td>
<td>Robertson</td>
<td>8766</td>
<td>3839</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run
(Continuation 6)

<table>
<thead>
<tr>
<th>Location</th>
<th>Code</th>
<th>A</th>
<th>AR</th>
<th>AR</th>
<th>AR</th>
<th>AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEMPHILL</td>
<td>342</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAN AUGUSTINE</td>
<td>343</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLSPRING</td>
<td>344</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAN SABA</td>
<td>345</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATHIS</td>
<td>346</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CENTER</td>
<td>347</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIO GRANDE CITY</td>
<td>348</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BENROOK</td>
<td>349</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVERMAN</td>
<td>350</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIVERAOKS</td>
<td>351</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROVETON</td>
<td>352</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WOODVILLE</td>
<td>353</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EL CAMPO</td>
<td>154</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WHEELER</td>
<td>355</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IOWA PARK</td>
<td>356</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GEORGETOWN</td>
<td>357</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAYLOR</td>
<td>358</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUITMAN</td>
<td>359</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLAINES</td>
<td>360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZAPATA</td>
<td>361</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AZLI MULESHOE PD</td>
<td>362</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **AZKK MORTON**: 0.00
- **AZKW SPUR PD**: 0.00
- **AZKL PLAINVIEW PD**: 0.00
- **AZLR PLAINVIEW**: 0.00
- **AZLQ LEVELLAND PD**: 0.00
- **AZLC LITTLEFIELD PD**: 0.00
- **AZKA LITTLEFIELD**: 0.00
- **AZTI OLTON PD**: 0.00
- **AZGL LURHOCK DPS**: 0.00
- **AZLK LURHOCK PD**: 0.00
- **AZLL LURHOCK**: 0.00
- **AZLP SLATON PD**: 0.00
- **AZLJ TAHOKA PD**: 0.00
- **AZKS TAHOKA**: 0.00
- **AZLF BROWNFIELD PD**: 0.00
- **AZLF BROWNFIELD**: 0.00
<table>
<thead>
<tr>
<th>Line</th>
<th>AZLN DENVER CITY PD</th>
<th>295</th>
<th>7.39</th>
<th>20.87</th>
<th>0.00</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>NAAF MULESHOE S.O.</td>
<td>0.00</td>
<td>0.00</td>
<td>1.66</td>
<td>5.26</td>
<td>0.00</td>
</tr>
<tr>
<td>402</td>
<td>NABG CROSBYTTON S.O.</td>
<td>312</td>
<td>1.66</td>
<td>5.26</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>403</td>
<td>NACG LEVELLAND S.O.</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>404</td>
<td>NAAE MULESFOE 5.0.</td>
<td>157</td>
<td>2.68</td>
<td>8.48</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>405</td>
<td>NAFR PLAINES S.O.</td>
<td>357</td>
<td>1.53</td>
<td>4.46</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>406</td>
<td>AAAUSTIN SWITCHER</td>
<td>272</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 3-2. Input Data for the Example Run
(Continuation 7)
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>455</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>456</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>457</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>458</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>459</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>460</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>461</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>462</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>463</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>464</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>465</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>466</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>467</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>468</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>469</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>470</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>471</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>472</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>473</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>474</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>475</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>476</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>477</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>478</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>479</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>480</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>481</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>482</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>483</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>484</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>485</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>486</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>487</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>488</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>489</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>490</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>491</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>492</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>493</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>494</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>495</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>496</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>497</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>498</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>499</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>500</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>501</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>502</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>503</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>504</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>505</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>506</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>507</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>508</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>509</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>510</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>511</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>512</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
### Table 3-2. Input Data for the Example Run (Continuation 9)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>523</td>
<td>135</td>
<td>135</td>
<td>135</td>
<td>135</td>
</tr>
<tr>
<td>524</td>
<td>135</td>
<td>135</td>
<td>135</td>
<td>135</td>
</tr>
<tr>
<td>525</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>526</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>527</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>528</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>529</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>530</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>531</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>532</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>533</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>534</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>535</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>536</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>537</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>538</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>539</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>540</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>541</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>542</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>543</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>544</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>545</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>546</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>547</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>548</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>549</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>550</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>551</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>552</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>553</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>554</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>555</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>556</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>557</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

#### LINEAR FOR 1200 BAUD LINE

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>558</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>559</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>560</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>561</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>562</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>563</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>564</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>565</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>566</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>567</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>568</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>569</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### LINEAR FOR 2400 BAUD LINE

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>568</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>569</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
</tbody>
</table>

#### LINEAR FOR 4800 BAUD LINE

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>568</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>569</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
</tbody>
</table>

#### LINEAR FOR 7200 BAUD LINE

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>568</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
</tr>
<tr>
<td>569</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
</tbody>
</table>

#### LINEAR FOR 9600 BAUD LINE

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
<tr>
<td>568</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
</tr>
<tr>
<td>569</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
</tr>
</tbody>
</table>

#### LINEAR FOR 14400 BAUD LINE

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
</tr>
<tr>
<td>568</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
</tr>
<tr>
<td>569</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
</tr>
</tbody>
</table>

#### LINEAR FOR 28800 BAUD LINE

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
</tr>
<tr>
<td>568</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>569</td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
</tr>
</tbody>
</table>

#### LINEAR FOR 57600 BAUD LINE

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
</tr>
<tr>
<td>568</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
</tr>
<tr>
<td>569</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
</tr>
</tbody>
</table>

#### LINEAR FOR 115200 BAUD LINE

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
</tr>
<tr>
<td>568</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
</tr>
<tr>
<td>569</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run (Continuation 10)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>570</td>
<td>ING/NL 50 200 .45 .45 0 0 0 0</td>
</tr>
<tr>
<td>571</td>
<td>ADM/NL 300 300 .45 .0 0 0 0 0</td>
</tr>
<tr>
<td>572</td>
<td>NCIC 50 90 .0 9.21 0 0 0 0</td>
</tr>
<tr>
<td>573</td>
<td>DB/DLS 90 50 3.49 1.1 0 0 0 0</td>
</tr>
<tr>
<td>574</td>
<td>DB/ANT 90 50 .45 .45 0 0 0 0</td>
</tr>
<tr>
<td>575</td>
<td>(1) 571 572 573 574</td>
</tr>
<tr>
<td>576</td>
<td>(2) 2 AAAA 1 GSITCHER ASSIGNMENT WITH 11+1X,A4+1S</td>
</tr>
<tr>
<td>577</td>
<td>3 TERMINATE GSITCHER ASSIGNMENT</td>
</tr>
<tr>
<td>578</td>
<td>(1) 2 AAAA QSTATE CENTER</td>
</tr>
<tr>
<td>579</td>
<td>(2) 6.43 2 TOTAL XSAC 8 REG. AT THE AUSTIN SWITCHER WITH F8.5 AI3</td>
</tr>
<tr>
<td>580</td>
<td>(3) 20 7.0 1 GSITCHER LINE,RESP. TIME, MPROC WITH 13,F5.2+12</td>
</tr>
<tr>
<td>581</td>
<td>(1) 1 Q1 FOR PLOT, 10 AND 0 FOR SKIPPING IT WITH 13</td>
</tr>
</tbody>
</table>

CPU: 787 CTP: 0.91 SUPS: 17.904

QBKRKP PRINTS
Each group is then divided into three subgroups of 12 cards, one for each chargeable item. Each subgroup is then divided into 3 units, 4 cards per unit, according to the three types of traffic density combinations: high-high, high-low and low-low. Each specific unit is then divided into two subunits of 2 cards. The first subunit is for installation costs, and the second for recurring costs. The first card of each subunit is for costs under half duplexing mode, and the second for costs under full duplexing mode. The first number of each card is the cost for the initial unit; the second for each additional unit at the same location.

Item 14 indicates that, in Texas, a linear costing function is used for all of the line service charges. The first card gives the installation charge as a function of distance, and the second the monthly recurring charge as a function of distance. Under each line type, the line cost is also given as a function of traffic density mix between two terminals.

Item 15 indicates that an optimization process is requested after a star network is formed.

Item 16 shows the line protocol characteristics for those three line types under consideration by providing data such as no. of polling characters, modem turn-around time, while item 17 and 18 give the message statistics. Item 19 indicates that a 110 milli-second is used as the average transaction service time needed in the central switcher of the system being studied.

Item 20 pre-selects system termination AAAA as the RSC, and item 21 designates AAAA as the system centroid.

The three remaining cards define the total traffic load at the central switcher, the multidrop line constraints, and a request for a CalComp plot at the end of each regional network optimization.

3.4.3 Output

After a normal termination from a STACOM program run, outputs from the printer and the CalComp plotter should contain all data desired. This subsection describes the contents of these outputs obtained from the example run.

3.4.3.1 Printer Output. Data showing results from a normal program execution of the STACOM program are printed on a regular printer. Table 3-3 shows the exact output obtained from running the STACOM program utilizing the set of input data as given in Paragraph 3.4.2.

To facilitate the following discussions, the contents of Table 3-3 are itemized as shown.

Item 1 reminds the user that only one region has been considered in this specific run. Item 2 shows the line protocol for each
Table 3-3. Printer Output from the Example Run

<table>
<thead>
<tr>
<th>Time</th>
<th>Polls</th>
<th>NAKs</th>
<th>Poll O/H</th>
<th>NAK O/H</th>
<th>MSG O/H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1200</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2400</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4800</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Averages:**
- Input MSG with Prio 1 = 147.0 Chars
- Output MSG with Prio 1 = 212.2 Chars
- Overall Avg. MSG = 184.0 Chars

**Traffic Matrix (bps):**

<table>
<thead>
<tr>
<th>Term</th>
<th>AZLI</th>
<th>AZKK</th>
<th>AZKX</th>
<th>AFXZ</th>
<th>AZLD</th>
<th>AZLA</th>
<th>AZLR</th>
<th>AZLD</th>
<th>AZLC</th>
<th>AZKA</th>
<th>AZTL</th>
<th>AZKL</th>
<th>AZLL</th>
<th>APLH</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAFIN</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.0</td>
<td>1.0</td>
<td>1.4</td>
<td>1.2</td>
<td>0.7</td>
<td>3.1</td>
<td>3.3</td>
<td>5.3</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td>TRFOUT</td>
<td>2.6</td>
<td>1.4</td>
<td>2.4</td>
<td>1.0</td>
<td>7.4</td>
<td>2.0</td>
<td>3.7</td>
<td>2.7</td>
<td>2.1</td>
<td>15.1</td>
<td>15.4</td>
<td>7.9</td>
<td>3.2</td>
<td></td>
</tr>
<tr>
<td>TRAFIN</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>TRFOUT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>TRAFIN</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>TRFOUT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run
(Continuation 1)

<table>
<thead>
<tr>
<th></th>
<th>TOTAL TRAFFIC ORIGINATED FROM SYS. TERM. (MTC/SCC)</th>
<th></th>
<th>TOTAL TRAFFIC DESTINATED TO SYS. TERM. (MTC/SCC)</th>
<th></th>
<th>TOTAL TRAFFIC</th>
<th>TOTAL TRAFFIC DESTINATED TO SYS. TERM. (MTC/SCC)</th>
<th>TOTAL TRAFFIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZLI</td>
<td>.060</td>
<td>AZFK</td>
<td>.545</td>
<td>AZKV</td>
<td>.837</td>
<td>AZFC</td>
<td>.617</td>
</tr>
<tr>
<td>AYLA</td>
<td>.564</td>
<td>AYLI</td>
<td>1.625</td>
<td>AYLD</td>
<td>1.893</td>
<td>AYLC</td>
<td>1.145</td>
</tr>
<tr>
<td>AZTI</td>
<td>.115</td>
<td>AZTL</td>
<td>.449</td>
<td>AZTL</td>
<td>2.104</td>
<td>AZTC</td>
<td>.291</td>
</tr>
<tr>
<td>AZLU</td>
<td>.103</td>
<td>AZKS</td>
<td>.547</td>
<td>AZLF</td>
<td>2.370</td>
<td>AZLF</td>
<td>.537</td>
</tr>
<tr>
<td>NAAF</td>
<td>.221</td>
<td>NANO</td>
<td>.221</td>
<td>NAGA</td>
<td>.267</td>
<td>NAAF</td>
<td>.267</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TOTAL TRAFFIC</td>
<td></td>
<td>TOTAL TRAFFIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>41.469</td>
<td></td>
<td>TOTAL TRAFFIC</td>
<td></td>
<td>114.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TOTAL TRAFFIC DESTINATED TO SYS. TERM. (MTC/SCC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AZLI</td>
<td>2.364</td>
<td>AZFK</td>
<td>1.411</td>
<td>AZKV</td>
<td>2.375</td>
<td>AZLF</td>
<td>1.133</td>
</tr>
<tr>
<td>AYLA</td>
<td>7.415</td>
<td>AYLI</td>
<td>1.056</td>
<td>AYLD</td>
<td>3.767</td>
<td>AZFC</td>
<td>2.099</td>
</tr>
<tr>
<td>AZTI</td>
<td>2.135</td>
<td>AZTL</td>
<td>1.144</td>
<td>AZTL</td>
<td>4.020</td>
<td>AZTC</td>
<td>7.060</td>
</tr>
<tr>
<td>AZLU</td>
<td>2.945</td>
<td>AZKS</td>
<td>1.325</td>
<td>AZLF</td>
<td>5.459</td>
<td>AZLF</td>
<td>1.114</td>
</tr>
<tr>
<td>NAAF</td>
<td>.101</td>
<td>NANO</td>
<td>.701</td>
<td>NAGA</td>
<td>1.131</td>
<td>NAAF</td>
<td>.068</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TOTAL TRAFFIC</td>
<td></td>
<td>TOTAL TRAFFIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>114.30</td>
<td></td>
<td>TOTAL SYSTEM TRAFFIC</td>
<td></td>
<td>114.30</td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run
(Continuation 2)

<table>
<thead>
<tr>
<th>TERM</th>
<th>AZLR</th>
<th>ACLL</th>
<th>AZLK</th>
<th>AZLH</th>
<th>AZLI</th>
<th>AZNK</th>
<th>AZLX</th>
<th>AZLO</th>
<th>AZLA</th>
<th>AZLC</th>
<th>AZLE</th>
<th>AZLI</th>
<th>AZLN</th>
<th>AZLL</th>
<th>AZLK</th>
<th>AZLL</th>
<th>AZLE</th>
<th>AZLK</th>
<th>AZLL</th>
<th>AZLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZLI</td>
<td>93</td>
<td>57</td>
<td>67</td>
<td>31</td>
<td>31</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>105</td>
<td>70</td>
<td>37</td>
<td>64</td>
<td>40</td>
<td>40</td>
<td>37</td>
<td>64</td>
<td>70</td>
<td>37</td>
<td>40</td>
</tr>
<tr>
<td>AZNK</td>
<td>64</td>
<td>94</td>
<td>31</td>
<td>94</td>
<td>31</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>40</td>
<td>37</td>
<td>64</td>
<td>40</td>
<td>37</td>
<td>64</td>
<td>70</td>
<td>37</td>
<td>64</td>
<td>70</td>
<td>37</td>
</tr>
<tr>
<td>AZLX</td>
<td>64</td>
<td>94</td>
<td>31</td>
<td>94</td>
<td>31</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>40</td>
<td>37</td>
<td>64</td>
<td>40</td>
<td>37</td>
<td>64</td>
<td>70</td>
<td>37</td>
<td>64</td>
<td>70</td>
<td>37</td>
</tr>
<tr>
<td>AZLO</td>
<td>23</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>74</td>
<td>74</td>
<td>74</td>
<td>74</td>
<td>64</td>
<td>71</td>
<td>71</td>
<td>64</td>
<td>71</td>
<td>71</td>
<td>64</td>
<td>71</td>
<td>71</td>
<td>64</td>
<td>71</td>
</tr>
<tr>
<td>AZLA</td>
<td>52</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td>AZLC</td>
<td>52</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>AZLE</td>
<td>52</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>AZLI</td>
<td>64</td>
<td>94</td>
<td>31</td>
<td>94</td>
<td>31</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>40</td>
<td>37</td>
<td>64</td>
<td>40</td>
<td>37</td>
<td>64</td>
<td>70</td>
<td>37</td>
<td>64</td>
<td>70</td>
<td>37</td>
</tr>
<tr>
<td>AZLN</td>
<td>14</td>
<td>21</td>
<td>31</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>AZLK</td>
<td>14</td>
<td>21</td>
<td>31</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>AZLL</td>
<td>14</td>
<td>21</td>
<td>31</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>AZLR</td>
<td>14</td>
<td>21</td>
<td>31</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
</tbody>
</table>

**Indices for Sys. Term:**
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

RSC = AAAA FOR REGION 1
Table 3-3. Printer Output from the Example Run  
(Continuation 3)

<table>
<thead>
<tr>
<th>SYSTEM TERMINAL</th>
<th>AZL1</th>
<th>AZLM</th>
<th>AZLC</th>
<th>AZLA</th>
<th>AZLD</th>
<th>AZL1</th>
<th>AZLM</th>
<th>AZLC</th>
<th>AZLA</th>
<th>AZLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE UTILIZATION</td>
<td>.004</td>
<td>.003</td>
<td>.003</td>
<td>.002</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
</tr>
<tr>
<td>DISTANCE FROM NAE</td>
<td>358</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
</tr>
<tr>
<td>TRAFFIC</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
</tr>
<tr>
<td>CPU TO CPU</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
</tr>
<tr>
<td>LINE RESPONSE TIME</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INST. COSTS</th>
<th>LINES</th>
<th>SER. 1.</th>
<th>MODEM</th>
<th>DROP</th>
<th>TOTAL COST</th>
<th>INST. COST</th>
<th>ANNUAL RECUH. COST</th>
<th>REGIONAL STA NETWORK AND ITS COSTS- 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>120</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>480</td>
<td>240</td>
<td>900</td>
<td>130</td>
</tr>
</tbody>
</table>

**REGIONAL STA NETWORK AND ITS COSTS- 1**

<table>
<thead>
<tr>
<th>SYSTEM TERMINAL</th>
<th>AZL1</th>
<th>AZLM</th>
<th>AZLC</th>
<th>AZLA</th>
<th>AZLD</th>
<th>AZL1</th>
<th>AZLM</th>
<th>AZLC</th>
<th>AZLA</th>
<th>AZLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE UTILIZATION</td>
<td>.004</td>
<td>.003</td>
<td>.003</td>
<td>.002</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
</tr>
<tr>
<td>DISTANCE FROM NAE</td>
<td>358</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
</tr>
<tr>
<td>TRAFFIC</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
</tr>
<tr>
<td>CPU TO CPU</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
<td>794</td>
</tr>
<tr>
<td>LINE RESPONSE TIME</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
<td>3.27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INST. COSTS</th>
<th>LINES</th>
<th>SER. 1.</th>
<th>MODEM</th>
<th>DROP</th>
<th>TOTAL COST</th>
<th>INST. COST</th>
<th>ANNUAL RECUH. COST</th>
<th>REGIONAL STA NETWORK AND ITS COSTS- 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>120</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>480</td>
<td>240</td>
<td>900</td>
<td>130</td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run  
(Continuation 4)

<table>
<thead>
<tr>
<th>SYSTEM TERMIN.</th>
<th>NAAE</th>
<th>NAGP</th>
<th>NACG</th>
<th>NAFA</th>
<th>NAAW</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. OF LINES REQ.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2KB</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>n</td>
</tr>
<tr>
<td>2.4KB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>4.8KB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>LINE UTILIZATION</td>
<td>.002</td>
<td>.002</td>
<td>.002</td>
<td>.002</td>
<td>.000</td>
</tr>
<tr>
<td>DISTANCE FROM MSC TRAFFIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE TO CPU</td>
<td>.221</td>
<td>.221</td>
<td>.257</td>
<td>.284</td>
<td>.000</td>
</tr>
<tr>
<td>CPU TO LINE</td>
<td>.701</td>
<td>.701</td>
<td>1.131</td>
<td>.648</td>
<td>.000</td>
</tr>
<tr>
<td>LINE RESPONSE TIME</td>
<td>3.273</td>
<td>3.273</td>
<td>3.273</td>
<td>3.273</td>
<td>.000</td>
</tr>
<tr>
<td>INST. COSTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>n</td>
</tr>
<tr>
<td>MODERN</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>n</td>
</tr>
<tr>
<td>DROP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>ANNUAL REPLAC. COST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>14364</td>
<td>11160</td>
<td>12700</td>
<td>12096</td>
<td>n</td>
</tr>
<tr>
<td>MODERN</td>
<td>360</td>
<td>360</td>
<td>360</td>
<td>360</td>
<td>n</td>
</tr>
<tr>
<td>DROP</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>n</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INST. COST</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>n</td>
</tr>
<tr>
<td>RECUR. COST</td>
<td>15492</td>
<td>12288</td>
<td>13904</td>
<td>14124</td>
<td>n</td>
</tr>
<tr>
<td>TOTAL COSTS</td>
<td>16712</td>
<td>13508</td>
<td>15104</td>
<td>15348</td>
<td>n</td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run
(Continuation 5)

<table>
<thead>
<tr>
<th>SUBNET NO.</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGINNING NODE</td>
<td>AZL</td>
<td>AZK</td>
</tr>
<tr>
<td>NO. OF TERM.</td>
<td>19</td>
<td>6</td>
</tr>
<tr>
<td>NO. OF LINES</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1.2KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.4KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.8KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LINE UTILIZATION</td>
<td>0.158</td>
<td>0.817</td>
</tr>
<tr>
<td>TOTAL MILEAGE</td>
<td>617</td>
<td>388</td>
</tr>
<tr>
<td>TRAFFIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE TO CPU</td>
<td>55.691</td>
<td>5.305</td>
</tr>
<tr>
<td>CPU TO LINE</td>
<td>100.777</td>
<td>12.617</td>
</tr>
<tr>
<td>LINE RESPONSE TIME</td>
<td>4.038</td>
<td>3.416</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INST. COSTS</th>
<th>SUM TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>0</td>
</tr>
<tr>
<td>SER.T.</td>
<td>280</td>
</tr>
<tr>
<td>MODEM</td>
<td>1300</td>
</tr>
<tr>
<td>DROP</td>
<td>0</td>
</tr>
<tr>
<td>ANNUAL RECURR. COST</td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>16492</td>
</tr>
<tr>
<td>SER.T.</td>
<td>4640</td>
</tr>
<tr>
<td>MODEM</td>
<td>6064</td>
</tr>
<tr>
<td>DROP</td>
<td>3120</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>31066</td>
</tr>
<tr>
<td>INST. COST</td>
<td>1560</td>
</tr>
<tr>
<td>RECUR. COST</td>
<td>31066</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>32646</td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run
(Continuation 6)

<table>
<thead>
<tr>
<th>Table 3-3. Printer Output from the Example Run (Continuation 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2→ REGIONAL CENTRES: A4AA</td>
</tr>
<tr>
<td>SUBNETWORK BEGINS AT</td>
</tr>
<tr>
<td>AZL  AZL P AZL A AZX Z NAQ G AZKW</td>
</tr>
<tr>
<td>AZT I AZK A NAQ G AZKK</td>
</tr>
<tr>
<td>NAAL AZLD</td>
</tr>
<tr>
<td>AZL C AZL I</td>
</tr>
<tr>
<td>AZK S AZL J AZL F NAF A AZL H</td>
</tr>
<tr>
<td>AZL E AZL I</td>
</tr>
</tbody>
</table>
individual line type under consideration. For example, a modem turn-
around time of 50 milliseconds has been used in the run.

Item 3 shows the traffic characteristic as calculated by the
STACOM program and item 4 prints out the pre-assignment activities. In
this example run, the system termination AAAA is preselected as the
regional switching center; since only one region is under consideration,
all of the remaining system terminations are assigned to region 1.

Item 5 shows a small portion of a traffic matrix from each
system termination to four data bases calculated by the program. Item 6
prints the total incoming/outgoing traffic in bps to/from each individual
system termination. Also included is total incoming/outgoing traffic
to/from the system.

Item 7 gives a short list of point-to-point distances between
system terminations as calculated by the program.

Item 8 gives the system centroid as designated from the input.
Item 9 shows the CPU utilization at the central switcher of the system
being studied.

Item 10 gives the IDs and names of all system terminations in
the region and their internal indices. Item 11 prints the regional
switching center for the region which has been preselected. In this run,
the RSC turns out to be the central switcher.

Item 12 provides the details of the star network developed by
the program. For example, the system termination AZLI is linked to the
regional switching center AAAA by a 1200 bps line. With the traffic as
shown, its line utilization is only .004 and response time 3.279 seconds.
It is 399 miles away from AAAA. Based on the tariff applicable for Texas,
it's installation costs are $20 for service terminal and $100 for modems.
Annual recurring costs are $892 for lines, $360 for service terminals,
$528 for modems and $240 for the drop charges. After the printout for the
star network, the multidrop network (as generated by the STACOM program)
is printed as given by item 13. In this example run, two distinctive
subnetworks have been generated. Both subnetworks require only the 1200
bps lines. In addition to data similar to item 12, it also includes the
total number of terminals on each multidrop line and the total connection
mileage. Summarized costs are also provided.

Finally, the actual structure of the final multidrop network
is printed as item 14. It is printed in a tree-type form, relating each
individual termination to others.

The above described printer output is a copy of the FORTRAN
output alternate file, 100. In addition to this, a regular FORTRAN output
file, 6, is generated by the program. For this example run, Table 3-4 is
the copy of output file, 6. It indicates all of the request messages go
by the program during its input phase. The last two lines are an
indication that the program has been successfully executed.
Table 3-4. Unit 6 Printer Output from the Example Run

<table>
<thead>
<tr>
<th>ASSUME NUMBER OF REGIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTER NR AND STRIKE RETURN KEY</td>
</tr>
<tr>
<td>TYPE IN NO. OF SYS. TERMS; DATA HASES AND CITIES WITH FORMAT 915</td>
</tr>
<tr>
<td>THERE ARE 2 SYS. TERMS; 4 DATA HASES 35 CITIES</td>
</tr>
<tr>
<td>TYPE IN DATA BASE LOCATIONS WITH FORMAT 6(1X,AN)</td>
</tr>
<tr>
<td>4 DATA HASES ARE AT AAAA NNNN SSSS NNNN</td>
</tr>
<tr>
<td>TYPE IN CITY V=H WITH FORMAT 13X(15;2X;15)</td>
</tr>
<tr>
<td>IF IN PIV NO., NAME MAPPING ADR. AND TRAFFIC</td>
</tr>
<tr>
<td>WITH FORMAT 14;1X;4A6,14;6FA;2</td>
</tr>
<tr>
<td>TYPE IN NO. OF RATE STRUCTURES UNDER CONSIDERATION WITH FORMAT 13</td>
</tr>
<tr>
<td>TYPE IN RATE APPLICATION TO EACH COMMON, READ EACH SYS. TERM, WITH FORMAT 1012</td>
</tr>
<tr>
<td>READ IN TRAFFIC DENSITY TYPE AND RATE STRUCTURE FOR EACH CITY WITH FORMAT 0011</td>
</tr>
<tr>
<td>TYPE IN NO. OF LINF TYPES APPLICABLE WITH FORMAT 13</td>
</tr>
<tr>
<td>TYPE IN NAME+ CAPACITY; UTIL, FACTOR AVAIL. FOR EACH LINE TYPE WITH FORMAT 6(1X;4X;1X;2X;21X;1X)</td>
</tr>
<tr>
<td>TYPE IN NO. OF DEVICES AND NAMES FOR EACH LINF TYPE WITH FORMAT 13/101(AD+1Y)</td>
</tr>
<tr>
<td>TYPE IN INST. AND RECUR. COSTS WHT RATE STRUCTURE, LINE TYPE, DEVICE, TRAFFIC DENSITY AND DAINPLEXING MODE WITH FORMAT 2E9.2/2E9.2</td>
</tr>
<tr>
<td>TYPE IN INST. COSTS &quot;IN&quot; LINES WHT RATE, LINF, DENSITY &quot;Y&quot; DAINPLEXING NODE</td>
</tr>
<tr>
<td>TYPE IN INDEX FOR LINEARITY OF LINE RECUR. COST FUNCTION WITH 1 LINEAR AND NONLINEAR OTHERWISE WITH FORMAT 11 FOR EACH LINF TYPE</td>
</tr>
<tr>
<td>TYPE IN RECUR. COSTS WITH FORMAT 4E9.2 IF LINEAR WITH FORMAT 10F9.3/10F9.3 IF NONLINEAR</td>
</tr>
<tr>
<td>IF NONLINEAR, TYPE INF.2</td>
</tr>
<tr>
<td>TYPE IN ACTION INDICES FOR EACH REGION</td>
</tr>
<tr>
<td>1ST ELEMENT: 1 = INSERTION TO THIS PRELOADED REGION IS OK</td>
</tr>
<tr>
<td>2ND ELEMENT: 1 = OPTIMIZATION IS NEEDED</td>
</tr>
<tr>
<td>TYPE IN REGION INDEX AND ACTION NUMBER &quot;NEED&quot;, WITH FORMAT 2I AND END IT WITH A 0</td>
</tr>
<tr>
<td>TYPE IN INFL, MAX, &quot;MEAN&quot; RATIO, &quot;HAPPY&quot; RATIO</td>
</tr>
<tr>
<td>TADA, TAD IN FORMAT (5;14;2F7.5)</td>
</tr>
<tr>
<td>TYPE III: NO. OF MSG TYPES, AND TRAFFIC STATISTICS SUCH AS MSGAM, MSRED, MSRED, RATIO WITH FORMAT 7F/1AD+7(12F;2F;6F;3)</td>
</tr>
<tr>
<td>TYPE IN PRELOADED SYSTEM TERMINALS AND RSF WITH FORMAT 11, 1X, 1X, 1X, AB</td>
</tr>
<tr>
<td>32 RMS DISTANCE IFV'S ARE OVERRUN</td>
</tr>
<tr>
<td>ASSUME A SYSTEM CENTERED</td>
</tr>
<tr>
<td>ENTER CODE FOR HSCC AND STRIKE RETURN KEY</td>
</tr>
<tr>
<td>INPUT TOTAL NO. OF TRANSACTIONS AND NO. OF ACCESS AT THE SWITCHER</td>
</tr>
<tr>
<td>ENTER WITH 40.4 AND 13 UNDER X5AC/SRC</td>
</tr>
<tr>
<td>READ IN LIMITS ON NO. OF SYS. TERMS, ON A LIM</td>
</tr>
<tr>
<td>RESPONSE TIME READ AND NO. OF PROCESSES WITH FORMAT 13;F;2.F;12</td>
</tr>
<tr>
<td>IF PLOTTING IS REQUIRED, TYPE 1 WITH FORMAT 13</td>
</tr>
<tr>
<td>TRYLINK HAS REG Accessed FOR 10052 TIMES</td>
</tr>
<tr>
<td>UPTMA HAS BEEN ACCESSIVE FOR 25 TIMES</td>
</tr>
</tbody>
</table>

QUIT INPUT PRINT
3.4.3.2 CalComp Plot. Figure 3-1 is the actual network graph as plotted by the CalComp plotter. It reflects the network as printed in the last part of printer output. It should be noted that because of the existence of identical V-H coordinates associated with system terminations in the example run, fewer distinctive nodes are shown in the plot. The root node is for the system termination, AAAA, which is the location of the Austin central switcher as used in the example run.

Figure 3-1. CalComp Plot from the Example Run
REFERENCES


APPENDIX A

STACOM PROGRAM LISTING

51928*STACOM(1),MAIN/0777
1 C******************************************************************************C
2 C******************************************************************************C
3 C STACOM TOPOLOGY PROGRAM
4 C******************************************************************************C
5 C JET PROPULSION LABORATORY
6 C 4800 OAK GROVE DRIVE
7 C PASADENA, CALIFORNIA 91103
8 C******************************************************************************C
9 C******************************************************************************C
10 C THIS PROGRAM IS DESIGNED TO PERFORM FORMATIONS OF REGIONS, SELECTIONS
11 C OF REGIONAL SWITCHING CENTERS, FORMATIONS OF INITIAL REGIONAL NETWORKS,
12 C OPTIMIZATION OF REGIONAL NETWORKS USING THE FRANK-WILLIAMS METHOD IF
13 C REQUESTED, AND FINALLY FORMATION OF AN INTERREGION NETWORK AND ITS
14 C OPTIMIZATION
15 C******************************************************************************C
16 C******************************************************************************C
17 C THIS TOPOLOGY PROGRAM CONTAINS ONE 'MAIN PROGRAM' AND ELEVEN SUBPROGRAMS
18 C THEY ARE AS FOLLOWS:
19 C******************************************************************************C
20 C MAIN PROGRAM : MAIN (REGION ASSIGNMENTS OF TERMINAL TERMINATIONS)
21 C SUBPROGRAM-1 : RAGMET (REGIONAL NETWORK FORMATION AND ITS OPTIMIZATION)
22 C SUBPROGRAM-2 : IRNOR (INTERREGION NETWORK OPTIMIZATION)
23 C SUBPROGRAM-3 : ICOSTJ (COSTING FUNCTION)
24 C SUBPROGRAM-4 : RHOFUN (LINE UTILIZATION FUNCTION)
25 C SUBPROGRAM-5 : LINUN (LINE CONF. DEFINITION BASED ON TRAFFIC)
26 C SUBPROGRAM-6 : PACK (STORING OR RETRIEVING DISTANCE DATA)
27 C SUBPROGRAM-7 : DIST (FINDING DISTANCE BETWEEN TWO GIVEN TERMINALS)
28 C SUBPROGRAM-8 : LINK (FINDING COMPRESSED INDEX FOR DIST)
29 C SUBPROGRAM-9 : RECOVT (RECOVERING COMPRESSED DISTANCE DATA)
30 C SUBPROGRAM-10: PLOTPT (PLOTTING EACH DROP ON A MULTIDROP NETWORK)
31 C SUBPROGRAM-11: RSPNSE (ESTIMATING RESPONSE TIME)
32 C******************************************************************************C
33 C******************************************************************************C
34 PARAMETER NWP=4, NWT=100, NLMITEP=1, NPC=102, NPO=1
35 PARAMETER NWP=130, NWP=1, NPO=9, NP=1
36 PARAMETER NWP=61, NPO=2-NPC+1/4+1
37 PARAMETER NWP=4, NPO=1, NPC
38 COMMON A/NIV/SVR (NP1), NRS (NP), JUMP (NP), TRAFI (NP1),
39 * TRAFI (NP1)
40 * /VH/ IVERT(NPC), IHRZ(NPC)
41 * /CONST/ NL=1, N3=1, N4=1, N7=1, N8=1, N9=1
42 * /INP/ IATE (NP1), ITRAN (NP+1), ITRAN (NP+1)
43 * /HCST/ A1STC (NP1), NPC (NP+1), NPO, N7, N8, N9
44 * /NML/ ANGL (NP1), ANGL (NP1), ANGL (NP1)
45 * /
46 * /RST/ RST (NP1), RST (NP1), RST (NP1)
47 * /NPL/ NPL (NP1), NPL (NP1), NPL (NP1)
48 * /NAME/ NAME (NP1), NAME (NP1), NAME (NP1)
49 * /SUM/ ASUM (NP1)
50 * /XMT/ TIMXMT (NP1), WATT (NP1)
51 * /WSL/ AMSL (?)
52 * /RUND/ NTERMS, TERMO, nPROC, MPLOT
53 * /ADV/ ID(XNP1), XCHG, KADD, OKD, OKCHG, OKCHG, FIRST DROP, KADD, JUST FOR LINE
54 * /INTEG/ DSTNCF
55 DIMENSION IACTN(NP1), INDFNT (NP1)
56 DIMENSION NUMRR (NP1), TRAFFC (2), TRAFF (NP7)

A-1
DATA ITRAFC/ITRAFINTPOUT/
DIMENSION TRM(NM,NM), DRM(NM,NM), NUMR(NP1), NUMR(NP1,4)
INTEGER SRV
DIMENSION OUTPR1(NP1)
NMAX=NPO MAXIMUM SIZE FOR OVERFLOW DISTANCE DATA TABLE
C
C SELECT NUMBER OF REGIONS
C
WRITE(6,220)
READ(5,735) NR1
WRITE(10,1011) NR1
ANR1=NP1
C
READ IN TRAFFIC DENSITY INDEX AND RATE STRUCTURE FOR EACH SYSTEM
C TERMINATION IN THE SYSTEM
C
CALL CREADA(N1)
C
READ IN RATE APPLICATION MATRIX
C
CALL CREADB(N2)
C
READ IN NAMES, CAPACITIES, UTILIZATION FACTORS AND AVAILABILITIES
C FOR LINES APPLICABLE IN THE SYSTEM
C
CALL CREADD(N3)
C
READ IN INSTALLATION AND RECURRING COSTS FOR CHARGEABLE ITEMS
C REQUIRED FOR COMMUNICATION LINES
C
CALL CREADD(N4)
C
READ IN INSTALLATION AND RECURRING COSTS FOR LINES
C
CALL CREADV
C
READ IN ACTIONS TO BE PERFORMED ON EACH REGIONAL NETWORK
C 1ST ELEMENT : 1=INSERTIONS TO PRELOADED REGIONS ARE ALLOWED
C 0= SUCH AN ACTION IS NOT ALLOWED
C 2ND ELEMENT : 1=NETWORK OPTIMIZATION IS TO BE PERFORMED
C 0=NO OPTIMIZATION IS NEEDED
C
CALL CREADV
C
READ IN LINE AND LINE PROTOCOL CHARACTERISTICS
C
CALL CREADR
C
CONVERT TRAFFIC FROM CHARACTERS/MIN TO RITS/SFC
C
DO 85 K=1,N
DO 85 L=1,N1
TRAFF(I,K,L)=TRAFF(I,K,L)*R/L.
85 CONTINUE
ISUM=0

A-2
DO 25 I=1+NCITY
116 ISUM=ISUM+1
117 TRAFF(I)=ISUM
118 CONTINUE
119 DO 70 I=1+NI
120 NUMPR(I) = 0 NUM OF SYSTEM TERMINATIONS AT EACH REGION
121 70 CONTINUE
122 WRITE(6,888)
123 805 READ(5,801) NCODE,NSTATE,NREGO
124 WRITE(IWT,804) NCODE, NSTATE, NREGO
125 NSTATE=LOCAL(NSTATE) GOTO CARDINAL INDEX
126 CONTINUE
127 CONTINUE
128 NREGO=NREGO + 1
129 CONTINUE
130 NSTATE=NSTATE + 1
131 GO TO 70
132 CONTINUE
133 CONTINUE
134 TRAFF(I)=0.
135 70 CONTINUE
136 IOVER=1 G0COUNTER FOR OVERSIZED TRAFFIC DATA
137 C CALCULATE DISTANCE DATA BETWEEN SYSTEM TERMINATIONS.
138 C DO 20 J=1+NCITY
139 C 20 K=1+NCITY
140 IF(J<K) 51,30,30
141 51 CONTINUE
142 ISQ=ISQ+1 IVFRT(J)+IVFRT(K)
143 ISQ=ISQ+1 IVRZ(J)+IVRZ(K)
144 IF(ISQ = EQ. 0 GO TO 23
145 SQ=SQ/10.1
146 CONTINUE
147 DIFF=SQ-SQ1
148 IF(DIFF .GT. 0.) SQ1=SQ1+1.
149 DIST=SQRT(SQ1)
150 KDIST=INT(DIST)
151 DIFF=KDIST-KDIST
152 IF(DIFF .GT. 0.) KDIST=KDIST+1
153 GOTO 23
154 CONTINUE
155 KDIST=0
156 22 CONTINUE
157 CONTINUE
158 23 CONTINUE
159 JKL=LINK(J,K)
160 IF(KDIST .LE. 510) GOTO 5
161 CALL OVERFL(JKL,KDIST)
162 GOTO 30
163 CONTINUE
164 CALL PACK(JKL,KDIST,1,PSTNCF)
165 30 CONTINUE
166 CONTINUE
167 IOVER=IOVER+1
168 WRITE(6,3) IOVER
169 C TOTAL INPUT TRAFFIC BY EACH SYS. TERMIN.
77-33, Vol. IV

C
171 TRFALL=0.0
172 TALLIT=0.
173 TALLDN=0.
174 DO 41 L=1,N1
175 TRAFIT(L)= 0.0
176 TRAFDN(L)= 0.0
177 DO 42 J=1,N7
178 TRAFIT(L) = TRAFIT(L) + TRAFDN(L+J)
179 TRAFDN(L) = TRAFDN(L) + TRAFDN(L+1+J)
180 42 CONTINUE
181 TALLDN=TALLDN+TRAFIT(L)
182 TALLDN=TALLDN+TRAFIT(L)
183 41 CONTINUE
184 TRFALL=TALLDN+TALLIT
185 C
186 C PRINT OUT TRAFFIC DATA BETWEEN SYSTEM TERMINATIONS
187 C
188 NTURN=N1/15 + 1
189 NREM=MOD(N1,15)
190 IF(NREM .EQ. 0) NTURN=NTURN-1
191 WRITE(IWT+11)
192 DO 1100 KK=1,1 FOR TEST ONLY
193 KK=KK+15
194 IF(KK .GT. N1) KK2=KK
195 KK=KK+15
196 WRITE(IWT+113) (INDXPT(J), J=KK1,KK2)
197 DO 99 J=1,N7
198 DO 97 JT=1,2
199 IF(NREM .EQ. 0) NTURN=NTURN-1
200 KK=KK+1
201 CONTINUE
202 WRITE(IWT+110) (TRAFC(KT), KT=1,2)
203 WRITE(IWT+1110) (INDXPT(J), J=KK1,KK2)
204 CONTINUE
205 CONTINUE
206 1100 CONTINUE
207 C
208 C PRINT OUT TRAFFIC ORIGINATED FROM EACH SYSTEM TERMINATION
209 C
210 WRITE(IWT+1013)
211 WRITE(IWT+1001) (INDEXPT(J), J=1,N1)
212 WRITE(IWT+74) TALLDN
213 C
214 C PRINT OUT TRAFFIC DESTINATED TO EACH SYSTEM TERMINATION
215 C
216 WRITE(IWT+1014)
217 WRITE(IWT+1001) (INDEXPT(J), J=1,N1)
218 WRITE(IWT+74) TALLIT
219 WRITE(IWT+75) TRFALL
220 C
221 C PRINT OUT DISTANCE DATA BETWEEN SYSTEM TERMINATIONS
222 C
223 NTURN=N1/15+1
224 NREM=MOD(N1,15)
225 IF(NREM .EQ. 0) NTURN=NTURN-1
226 NTURN=1 FOR SHORT OUTPUT
227 DO 101 KK=1, NTURN

A-4
77-53, Vol. IV

228     KK1=(KK-1)*15 + 1
229     KK2=KK*15
230     IF(KK2 > N1) KK2=N1
231     WRITE(IWT+100) (INDXPT(J),J=KK2, KK1,-1)
232     DO 99 J=1, KK2
233     IF(J.GE.KK1) KK1=J+1
234     DO 27 KK=KK2, KK1, -1
235     OUTPUT(R)=DIST(J+KP)
236     CONTINUE
237     WRITE(IWT+112) INDXPT(J), (OUTPUT(K), K=KK2, KK1, -1)
238     98     CONTINUE
239     101     CONTINUE
240     WRITE(6,210)
241     CONTINUE
242     READ(5,734) NSCC1
243     WRITE(IWT+105) NSCC1
244     NSCC1=LOCAL(NSCC1:
245     IF(NSCC1.LE.0) GOTO 4003
246     WRITE(6,4013)
247     GOTO 4005
248     400x     CONTINUE
249     TPR1 = TRAFF
250     WRITE(6,2101)
251     READ(5,2102) X5AC, NRGSW, QNRSKENG OF REQUESTS/TRANS AT SWITCH
252     WRITE(6,2103)
253     READ(5,2104) NTERMS, TMREQ, MPAC
254     WRITE(6,2105)
255     READ(5,2104) MPRINT QMPLT=1 IF PLOT IS NEEDED
256     C
257     C Pre-Calculate CPU Turnaround Time
258     C
259     CALL CWAITC
260     C
261     C Sum up Total Traffic for Preloaded System Terminations in Regions
262     C Which do not allow any insertions of other system Terminations
263     C
264     TPR2=0
265     DO 77 N=1, N1
266     NK=5VR(N)
267     IF(NK .LT. 0) GOTO 77 IWOH PRELOADED
268     IF(TACTN(NK,1).EQ.0) GOTO 77 IWOH PRELOADED
269     TPR2=TRAFTH(N) + TRAFTH(N) + TPRP
270     77     CONTINUE
271     DO 76 L=1, NR1
272     IF(TACTN(L,1).EQ.0, OR, HINPR(L,1).EQ.0) GOTO 76
273     NR1=NR1-1.
274     76     CONTINUE
275     TPR1=TPR1-TPR2
276     IF(NR1 .LT. 0) GOTO 726 None Region Case
277     C
278     C Determine Lower Limit for Average Regional Traffic
279     C
280     ZLAT=1
281     IF(NR1 .LT. 0) GOTO 340
282     TPR=TPR1/NR1
283     GOTO 350
284     340     CONTINUE
TPR=TPR1
CONTINUE

TPRL=TPR*(1.-ZETA)
DO 909 NREG=1:NRI
TRFS=0.
AMAX=0.
I=0
IF(NUMPR(NREG).NE.0) GOTO 5000 IF NREG IS A PRELOADED REGION
C
ASSIGN SYSTEM TERMINATIONS TO A REGION WITHOUT ANY PRELOADING
C
DO 400 NI=1:N1
IF(SVR(NI).NE.0) GOTO 400 IF NI IS PRELOADED
ADIST=ADIST(NSCC1+NI)
IF(ADIST .LE. AMAX) GOTO 400
AMAX=ADIST UPDATE LONGEST DIST. FROM NCC
II=NI UPDATE FARDEST SYS. TFRRN.
CONTINUE
N51=II THE FARDEST SYSTEM TERMINATION
TRFS=TRFS+TRAFD(NS1)+TRAFI(NS1)
SVR(NS1)=NREG
NUMPR(NREG)=NUMPR(NREG)+1
IF(ADIST .GT. TPRL) GOTO 707
GOTO 7021
5000 CONTINUE
IF(IACTN(NREG).EQ.1) GOTO 909 INSERTIONS ARE NOT ALLOWED
C
SUM UP TRAFFIC IN THIS REGION
C
DO 702 I=1:N1
IF(SVR(I).NE. NREG) GOTO 702
TRFS=TRFS+TRAFD(I)+TRAFI(I)
ADIST=ADIST(NSCC1+I)
IF(ADIST .GT. AMAX) II=I
CONTINUE
702 CONTINUE
CALL FINDD(NS1+NS2)
IF(NS2 .EQ. 0) GOTO 909
SVR(NS2)=NREG
NUMPR(NREG)=NUMPR(NREG)+1
TRFS=TRFS+TRAFD(NSP)+TRAFI(NSP)
IF(NREG .EQ. NR1) GOTO 7021
IF(ADIST .GT. TPPL) GOTO 707
GOTO 7021
707 CONTINUE
TPRI=TPR1-TRFS UPDATE REMAINING TRAFFIC
ANR1=ANR1-1
TPR=TPR1/ANR1 UPDATE AVERAGE TRAFFIC DED. REGION
TPPL=TPR*(1.-ZETA) UPDATE LOWER LIMIT
CONTINUE
GOTO 703
726 CONTINUE
C
ONE REGION CASE
342 C
343 DO 727 NN=1,N1
344 SUR(NN) = 1
345 727 CONTINUE
346 NUMBR(I) = N1
347 703 CONTINUE
348 C
349 C SELECT REGIONAL SWITCHING CENTER
350 C
351 DO 500 J=1,N1
352 WCASE = 1.0D12
353 NNBR = 0
354 DO 505 K=1,N1
355 IF(SVR(K) .NE. J) GO TO 505
356 NUMBR = NUMBR + 1
357 NNBR(NNBR) = K
358 NUMBR(NNBR) = INNXPT(K)
359 DO 490 I=1,4
360 NUMR(NNBR,I) = NAMEST(K,I)
361 490 CONTINUE
362 500 CONTINUE
363 C
364 C PRINT OUT PID AND NAMES FOR SYSTEM TERMINATIONS IN THE REGION J
365 C
366 WRITE(IWT,101A) J,(NUMBR(I),(NUMP(I,K),I=1,4),I=1,NMNR)
367 C
368 C PRINT OUT INDICES OF SYSTEM TERMINATIONS IN THE REGION J
369 C
370 WRITE(IWT,102A) (NUMR(I),I=1,NMNR)
371 C
372 IF(NRSC(J) .NE. 0) GO TO 501 QPPE-SELECTED
373 DO 520 K=1,NMNR
374 NN1 = NUMR(K)
375 SUMT = 0.0
376 DO 530 L=1,NMNR
377 NN2 = NUMR(L)
378 SUMT = SUMT + (TRAFFN(NN1) + TRAFFT(NN2)) + DST(NN1,NN2)
379 530 CONTINUE
380 IF(SUMT .LT. WCASE) GO TO 520
381 WCASE = SUMT
382 NRSC(J) = NN1
383 520 CONTINUE
384 CONTINUE
385 501 CONTINUE
386 NN4 = NRSC(J)
387 WRITE(IWT,1003) INDEXT(NN4),J
388 IF(JACTH(J,2) .EQ. 1) OPTIMIZATION IS REQUIRED
389 CALL RGMET(J,NURR,NUMR,JGO,NUMRR)
390 500 CONTINUE
391 C
392 C GENERATE INTER-REGION ORIGIN-DESTINATION MATRIX
393 C
394 C INITIALIZATION
395 C
396 IF(NR1 .LE. 2) GOTO 551
397 DO 902 K1=1,N7
398 KKK = NBASE(K1)
399 902 CONTINUE
399 IF(KKK .EQ. 0) WRITE(6,7777) K1
400 NBASE(K1)=NVR(KKK)
401 CONTINUE
402 DO 609 K1=1,NR1
403 CONTINUE
404 TRM(K1,K2)=0.
405 CONTINUE
406 DO 699 NR1=1,NR1
407 CONTINUE
408 DO 906 J=1,NR1
409 CONTINUE
410 MVRK(K1,J)=MVR(KK)
411 CONTINUE
412 NN2=NBASE(KK)  OOREGIONAL INPUT FOR KK'S DATA PAGE
413 TRM(J,NN2)=TRAFF(K,KK)+TMP(J,NN2)  QUITGOING TRAFFIC
414 CONTINUE
415 DO 915 NR1=1,NR1
416 CONTINUE
417 DO 905 J=1,NR1
418 CONTINUE
419 ORN(J,J1) = DIST(NR1,NR1)
420 CONTINUE
421 CONTINUE
422 IF(ORN(J,J1).LE.0) GOTO 910
423 LL=LL+10
424 LL=LL+10
425 IF(LL.GT.NR1) LL=NR1
426 WRITE(IWT,1030) NR1+NR1, (K,KK=LL,LU)
427 CONTINUE
428 WRITE(IWT,1021) I, (TRM(I,J),J=LL,L)
429 CONTINUE
430 CONTINUE
431 CONTINUE
432 CONTINUE
433 CONTINUE
434 CONTINUE
435 IF(LL.GT.NR1) LL=NR1
436 WRITE(IWT,1021) I, (TRM(I,J),J=LL,L)
437 CONTINUE
438 CONTINUE
439 CONTINUE
440 CONTINUE
441 CALL TRNOP(NR1,NLIMIT,TAM)
442 FORMAT(/'40X* TOTAL TRAFFIC=',F9.2)
443 FORMAT(/'55X* TOTAL SYSTEM TRAFFIC=',F9.2)
444 FORMAT(/'ASSUME NUMBER OF REGIONS*',
445 *) FORMAT(/' ENTER NR AND STRIKE RETURN KEY')
446 FORMAT(13)
447 FORMAT(1X*TYPE IN PRELODED SYSTEM TFRMN. AND PSC WITH*,
448 1 /'1X*FORMAT II+X+A,5*)
449 FORMAT(1X+X+A,5)
450 FORMAT(10X,1I2*X+A,5)
451 FORMAT(1X+X,A,5)
452 FORMAT(1X+X,A,5)
453 FORMAT(10X,1I2*X+A,5)
454 FORMAT(1X+X,A,5)
455 FORMAT(1X+X,A,5)
2101 FORMAT(1X,'INPUT TOTAL NO. OF TRANSACTIONS AND NO. OF ACCESS * *
2102 1 *AT THE SWITCHER*/1X,'ENTER WITH FA,5 AND 13 UNDER XSEC/SEC')
2103 FORMAT(8F5.13)
2104 FORMAT(1X,'READ IN LIMITS ON NO. OF SYS. TERMS. ON A 1 INF*/
2105 1 *RESPONSE TIME REGN AND NO. OF PROCESSES WITH FORMAT */
2106 2 */I5,F5.2,I2*)
2107 FORMAT(13F5.2,12)
2108 FORMAT(1X,'IF PLOTTING IS REQUIRED; TYPE 1 WITH FORMAT IX')
2109 FORMAT(1X,'THE NEXT SYSTEM TERMINATION M WHICH IS CLOSEST TO N
2110 C WHERE M HAS NOT BEEN ASSIGNED TO ANY REGION YET
2111 C ***************
2112 AM=200000,
2113 M=0
2114 DD 70A K=1,N1
2115 IF(SWR(K) .LE. 0) GOTO 70A
2116 ADIST=ADIST(N,K)
2117 TF(ADIST +GF, AMIN) GOTO 70A
2118 AMIN= ADIST
2119 M=K
2120 70A CONTINUE
2121 RETURN
2122 SUBROUTINE FIPIDDIN(M)
2123 C ***************
2124 C READ IN ACTIONS REGARDING INSPECTIONS OF SYSTEM TERMINATIONS
2125 C TO PRELOADED REGIONS AND REGIONAL NETWORK OPTIMIZATIONS
2126 C ***************
2127 WRITE(6,94)
2128 94 FORMAT('TYPE IN ACTION INDICS FOR EACH REGION * *
2129 C 1ST ELEMENT 1= INSPECTION TO THIS PRELOADED REGION IE OK *')
2130 STOP
2131 SUBROUTINE CRFADK
2132 C ***************
2133 C FIND THE NEXT SYSTEM TERMINATION M WHICH IS CLOSEST TO N
2134 C WHERE M HAS NOT BEEN ASSIGNED TO ANY REGION YET
2135 C ***************
77-53, Vol. IV

513 1 /* 2D ELEMENT: 1 = OPTIMIZATION IS NEEDED */
514 DO 200 NN1=1,N1
515 DO 200 NN2=1,2
516 IACT(NN1,NN2)=0
517 200 CONTINUE
518 WRITE(*,206)
519 206 FORMAT(' TYPE IN REGION INDEX AND ACTION NUMBER NEEDED',/)
520 * /* WITH FORMAT 212 AND END IT WITH A 0 0 */
521 250 CONTINUE
522 READ(5,201) NREG, NCODE
523 FORMAT(2I12). IF(NREG .EQ. 0) GOTO 255
524 IF(NREG .GT. N1 OR. NCODE .GT. 2) GOTO 260
525 IACT(NREG,NCODE)=1
526 GOTO 250
527 260 CONTINUE
528 WRITE(*,202)
529 202 FORMAT(' PLEASE RFYPE THE INPUT')
530 GOTO 250
531 265 CONTINUE AND MORE INPUT
532 RETURN
533 SUBROUTINE CRFADA(N1)
534 C ************
535 C
536 C FUNCTIONS OF THIS SUBROUTINE ARE TO
537 C 1. RECEIVE TOTAL NO. OF SYSTEM TERMINATIONS, DATA BASES AND CITIES
538 C 2. RECEIVE CITY LOCATIONS (V & H)
539 C 3. RECEIVE PID NO., SYS. TERM. NAMES, AND OTHER MAPPING AND TRAFFICS
540 C
541 C
542 C
543 WRITE(*,6,41)
544 61 FORMAT(' TYPE IN NO. OF SYS. TERMS, DATA BASES AND CITIES ') * WITH FORMAT 315*]
545 62 READ(5,10) NI,N7,NCITY ' A NUMBER OF SYSTEM TERMINATIONS
546 WRITE(*,7A1) 'I =N1+NCITY
547 78 FORMAT(' THERE ARE '15 SYS. TERMS, '14 DATA BASES, '15
548 CITIES') */ TYPE IN DATA BASE LOCATIONS WITH FORMAT 6(1Y4A4)*)
549 68 READ(5,15) (BASE(I),I=1,N7)
550 15 FORMAT(6(1X,A4))
551 WRITE(*,6,16) N7 (BASE(I),I=1,N7)
552 16 FORMAT(15 A4) DATA BASES ARE AT*6(2Y4A4))
553 WRITE(*,6,16)
554 161 FORMAT(' TYPE IN CITY V-H WITH FORMAT (23Y15,2X,15)')
555 162 READ(5,17) (VERT(I),HORIZ(I),I=1,NCITY)
556 17 FORMAT(23X15,2X,15)
557 WRITE(*,6,16)
558 16 FORMAT(' TYPE IN PID NO., NAME, MAPPING AND TRAFFIC') * WITH FORMAT 14(1X,A4,A4,14,F9.2)
559 60 DO 79 I=1,N1
560 76 FORMAT(' AND TRAFFIC') */ WITH FORMAT 14(1X,A4,A4,14,F9.2)
561 READ(5,80) I,NAMST(I),IAND(I),INAMP(I),ITRAF(I),IK(1,1),I1(I,7)
562 80 FORMAT(A4,1X,1X,1X,1X,1X,1X,A4,1X,1X,1X,1X,1X,1X,1X,1X,1X,1X,1X,1X,A4,F9.2)
563 * K=1+2 I=1(N7)
564 INDEX(I)=I1
565 79 CONTINUE
566 90 FORMAT(A4,1X,1X,A4,1X,1X,1X,1X,A4,1X,1X,1X,1X,1X,1X,1X,1X,1X,1X,1X,A4,F9.2)
567 10 FORMAT(315)
568 RETURN
569 SUBROUTINE CRFADR(N2)
C CREATE A RATE APPLICATION MATRIX IRATEJ[N,J,N2]
C
***************
WRITE(6,83)
83   FORMAT(* TYPE IN NO. OF RATE STRUCTURES UNDER*,
      * / * CONSIDERATION WITH FORMAT I3*)
769  READ (5,50) N2
769  WRITE(6,84)
84   FORMAT(* TYPE IN RATE APPLICATION TO EACH COMM. *,
      * / * WRT EACH SYS. TRM. WITH FORMAT 10I2*)
852  DO 11  IRATE=1,N2
583  READ (5,100) (IRATE(J,IRATE),J=1,N2)
584  11 CONTINUE
585
766  FORMAT(* READ IN TRAFFIC DENSITY TYPE AND RATE STRUCTURE*,
587      * / * FOR EACH CITY WITH FORMAT 40I1*)
588  READ(5,72) ((TRANP(I,J),J=1,2),I=1,NCITY)
589  72 FORMAT((40I1))
590
790  FORMAT (13)
591  100 FORMAT (10I2)
592  RETURN
593  *** SUBROUTINE CREATE(N3)
594  C
595
596  C READ IN NAMES, UTILIZATION FACTORS AND CAPACITY FIGURS
597  C FOR LINES TO BE USED IN THE SYSTEM
598  C
599  C
600  C
601  WRITE(6,85)
602  85   FORMAT(* TYPE IN NO. OF LINE TYPES APPLICABLE WITH FORMAT I3*)
603  READ (5,50) N3
604  WRITE(6,86)
605  86   FORMAT(* TYPE IN NAME: CAPACITY, UTIL. FACTOR AVAILABLE FOR *,
606      * / * EACH LINE TYPE WITH FORMAT A6*1X,16*1X,F8.2,(1X,I1)**)
607  READ(5,72) LINAMF(I),LINCAP(I),UTILIZ(I),LINMIX(I),TDUPLY(I)
608  72 CONTINUE
609  50 FORMAT (13)
610  100 FORMAT(A6*1X,16*1X,F3.2,2*(1X,I1))
611  RETURN
612  *** SUBROUTINE CRFADD(N4)
613  C
614  C
615  C CREATE A MATRIX OF BASIC INSTALLATION AND RECIPRING COSTS
616  C FOR CHARGEABLE ITEMS; ASSUMING COST IS A LINEAR FUNCTION
617  C
618  C
619  WRITE(6,87)
620  87   FORMAT(* TYPE IN NO. OF DEVICES AND NAMES FOR EACH LINE TYPE*,
621      * / * WITH FORMAT I3/10(A6*1X)**)
622  READ (5,50) N4, (NAMEHW(I),I=1,N4)
623  WRITE(6,88)
624  88   FORMAT(* TYPE IN INST. AND RECURR. COSTS WRT *,
625      * / * RATE STRUCTURE, LINE TYPE, DEVICE, TRAFFIC DENSITY *,
626      * / * AND MULTIPLYING MODE WITH FORMAT 2F9.2,2F9.2*)

A-11
DO 14 IRATE=1,N2
DO 14 ILINE=1,N3
DO 14 IDNSTY=1,3
READ(5,100)((ANSTC(IRATE,ILINE,INDX,TD,INSTY,J,K),K=1,2),J=1,2)
READ (5,101)((RECR(IRATE,ILINE,INSTY,T),J,K=1,2),J=1,2)
CONTINUE
13
CONTINUE
100 FORMAT (2F9.2/2F9.2)
50 FORMAT (13/A10(A6.1X))
RETURN
SUBROUTINE CREATE

************

C CREATE A MATRIX OF BASIC INSTALLATION AND RECURRING COSTS FOR
C LINES. COST MAY OR MAY NOT BE A LINEAR FUNCTION OF DISTANCE
C
C
C
WRITE(6,99)

9 FORMAT (1" TYPE IN INST. COSTS FOR LINES W/A INDICES.",1X,2/" RATE, LINE, DENSITY, AND DUPLEXING MODE.",1X,2/" WITH FORMAT 4F9.2")
WRITE(6,99)

90 FORMAT (1" TYPE IN INDEX FOR LINEARITY OF LINE RECUR. COST.",1X,2/" WITH FORMAT 11 FOR EACH LINE TYPE.")
WRITE(6,99)

91 FORMAT (1" TYPE IN RECUR. COSTS WITH FORMAT 4F9.2 IF LINEAR.",1X,2/" WITH FORMAT 10F9.3/10F9.3 IF NONLINEAR.",1X,2/" IF NONLINEAR, USE 10F9.2")
DO 14 IRATE=1,N2
DO 14 ILINE=1,N3
READ(5,200) INDX
IFLAG(IRATE,ILINE)=INDEX QLINE COST LINEARITY INDICATOR
DO 14 IDNSTY=1,3
READ (5,100)((ANSTC(IRATE,ILINE,INDX,TD,INSTY,J,K),K=1,2),J=1,2)
IF (INDX,INTERFACE=1) GO TO 3
C LINEAR COST FUNCTION
READ (5,100)((RECR(IRATE,ILINE,INSTY,T),J,K=1,2),J=1,2)
GO TO 14
CONTINUE
3 CONTINUE A NONLINEAR COST FUNCTION
READ (5,401)((RECR(IRATE,ILINE,INSTY,T),J,K=1,16),J=1,12)
CONTINUE
14 CONTINUE
100 FORMAT (4F9.2)
200 FORMAT (11)
401 FORMAT (10F9.3/10F9.3)
RETURN
FUNCTION LOCAL(NL)
C
C FIND LOCAL INDICES FOR SYSTEM TERMINATION WITH YD NL
C
C
LOCAL=0
IF(NL.EQ.0) RETURN
DO 401 IMM=1,N1
IF(IF_INDEX(YD,NL).EQ.0) GO TO 404
CONTINUE
401 CONTINUE
**LISTING:**

```
77-53, Vol. IV

684     RETURN
685     4002  LOCAL=IN
686     RETURN
687     SUBROUTINE OVERFL(J,K)
688     ***************
689     C
690     C  STORE OVERFLOW ELEMENT (J,K) AT LOCATION TOVR OF TABLE
691     C  LA AND PUT A MARK S11 AT LOCATION J OF TABLE HA (DISTANCE)
692     C
693     C
694     ***************
695     IF(IOVER1 .GE. NMAX) GOTO 4000
696     CALL PACK (J,S11+1,DISTANCE)
697     IVRO(IOVER1+1)=J
698     IVRO(IOVER1+2)=K
699     IOVER1=IOVER1+1
700     RETURN
701 4000  CONTINUE
702     CALL WRITE(6,4001)
703     FORMAT(2X,'THE OVERFLOW TABLE IS FFIIH FULLY LOADED**',
704     *    /'2X','PLEASE INCREASE ITS SIZE*)
705     STOP
706     SUBROUTINE CREATDR
707     C
708     C RECEIVE DATA FOR RESPONSE TIME CALCULATION
709     C
710     C NLSIN(NP)+2=INPUT MSG LENGTH AS A FUNCTION OF TYPE AND PRIORITY
711     C NMSOUT(NP)+2=OUTPUT MSG LENGTH AS A FUNCTION OF TYPE AND PRIORITY
712     C AMSL(T)= AVERAGE MSG LENGTH FOR
713     C 1=POLLING  2=NAK RESPONSE  3=INPUT MSG WITH PRIORITY 1
714     C 4=INPUT MSG  5=OUTPUT MSG WITH PRIORITY 1
715     C 6=OUTPUT MSGS WITH PRIORITY 2  7=ALL MSGS
716     C TMXMT(T+N3)=AVERAGE TRANSMISSION TIME FOR ABOVE ITEMS
717     C RATPR1(NP)+2=OUTPUT MSG DISTRIBUTION AND OUT-GOING MSG RATIO BY PRIORITY
718     C N+1.1 = PERCENT OF OUTPUT MSG SENT WITH PRIORITY 1 IF ITS TYPE IS N
719     C N+1.2 = PERCENT OF OUTPUT MSG WHOSE DESTINATION IS OUTSIDE OF
720     C RATIO(NP)+2=INPUT TRAFFIC DISTRIBUTION AS A FUNCTION OF TYPE AND PRIORITY
721     C RATIO(NP)+2=OUTPUT TRAFFIC DISTRIBUTION AS A FUNCTION OF TYPE AND PRIORITY
722     C
723     C
724     ***************
725     DIMENSION NLSIN(NP)+2,RATIO(NP)+2,RATIO(NP)+2,
726     N+1.1,N+1.2
727     N+1.3
728     WRITE(6,771)
729     FORMAT(14,TYPE IN NPL, NAK, NPLA, NAKOH, MOH++*,
730     *   /+ TADMM, TAD IN FORMAT (514,2F7.5)*)
731     READ(5,771)(NPL1(I),NAK1(I),NPLOH1(I),NAKO1(I),
732     *   MOH1(TADMT1(TAD1(I)),I=1,N3)
733     WRITE(6,733)(LINEP1(I)+NPL1(I),NAK1(I)+NPLOH1(I),NAKO1(I),
734     * MOH1(TADMT1(TAD1(I)),I=1,N3)
735     FORMAT(*2X,15.3X,*POLL CHAP,*,714,** PULL O/H=*
736     * 14/10X,NAK O/H=14,** MSG O/H=14,
737     2,/*X,MPISEM=,FA.3,1MPSEM=,FA.3)
738     77     FORMAT(*514,2F7.5)
739     77     WRITE(6,772)
740     772     FORMAT(*2X,15.3X,*POLL CHAP,*,714,** PULL O/H=*
741     * 14/10X,NAK O/H=14,** MSG O/H=14,
742     2,/*X,MPISEM=,FA.3,1MPSEM=,FA.3)
743     77     FORMAT(*514,2F7.5)
744     77     WRITE(6,772)
745
746
```
such as MSGnam, MSLn1, MSLout, RATIO WITH $^*$

READ (5, 77) NTYP
READ (5, 179) (MSGNAM(I), MSLIN(I), MSLOUT(I), J)
READ (5, 179) (RATIO(I, J), RATIO(I, J), J = 1, NTYP)

179 FORMAT (16.2) (16.2)
READ (5, 81) CPM AVG

FORMAT (F7.4)

C calculate average MSG length
C
DO 61 I = 1, 7
AMSL (1) = 0.
CONTINUE
DO 65 I = 1, 4
ASIM (1) = 0.
CONTINUE
DO 62 I = 1, NTYP
DO 66 J - 1, 2
J1 = J2
J2 = J4
AMSL(J1) = AMSL(J1) + MSLIN(J1) * RATIO(J1)
AMSL(J2) = AMSL(J2) + MSLOUT(J1) * RATIO(J1)
ASUM(J1) = ASUM(J1) + RATIO(J1)
ASUM(J2) = ASUM(J2) + RATIO(J1)
CONTINUE

66 CONTINUE
67 CONTINUE
BSUM = 0.
DO 68 J = 1, 4
J1 = J2
AMSL(J) = AMSL(J) + AMSL(J1)
BSUM = BSUM + ASUM(1)
CONTINUE

AMSL(1) = AMSL(1) / BSUM OVERALL AVG. MSG LENGTH
IF (ASUM(1) .GT. 0.) GOTO 69
AMSL(6) = AMSL(6) / ASUM(1)
CONTINUE

68 CONTINUE
AMSL(5) = AMSL(5) / ASUM(1)
RAVG, MSG LENGTH FOR PRIME1
AMSL(4) = AMSL(4) / ASUM(2) RAVG INPUT MSG LFN.
AMSL(3) = AMSL(3) / ASUM(1)
WRITE (108, 105) (AMSL(I), I = 1, 7)

105 FORMAT (/ "5X" AVG. INPUT MSG WITH PRIO 1 = ", F6.1, "+ CHAR$)

1

2

3

4

OVERALL AVG. MSG LFN. = ", F6.1, "+ CHAR$)

DO 65 K = 1, 3
AMSL(1) = NPL(K)
AMSL(2) = NAK(K)
TINXMT(1, K) = (AMSL(1) + NPLOH(K)) * R. / LINCAP(K) + TAMP(K)
TINXMT(2, K) = (AMSL(2) + NAKH(K)) * R. / LINCAP(K) + TAPP(K)
CONTINUE
63 CONTINUE
64 CONTINUE
BSUM = ASUM(3) + ASUM(4)
RETURN
798 SUBROUTINE CWAITC
799 C
800 C ************
801 C PPF=Calculate CPU Wait Time
802 C ************
803 C
804 RHOCPU=XSAC*CPUAVG/MPROC
805 WRITE(100,R50) RHOCPU
806 R50 FORMAT(* CPU UTILIZATION PER PROCESSOR IS *, F5.3)
807 IF(RHOCPU .LE. .A) GOTO 851
808 WRITE(6,R55)
809 R55 FORMAT(* THE CPU IS OVERLOADED, THEREFORE IT IS NO USE TO *
810 * GO FURTHER.*))
811 STOP
812 851 CONTINUE
813 BETA=RHOCPU
814 IF( MPROC .EQ. 1) GOTO 700
815 RH02=RHOCPU**2
816 BETA=2*RH02/(1+RHOCPU)
817 IF( MPROC .EQ. 2) GOTO 700
818 RH04=RHOCPU**4
819 BETA=256*RH04/(24+72*RHOCPU+96*RH02+64*RH04)
820 CONTINUE
821 700 CONTINUE
822 WAIT(4)=CPUAVG*(BETA/(MPROC*(1.-RHOCPU!!))+1)
823 WAIT(4)=WAIT(4)*NREQSW
824 RETURN
825 END

QPRT STACOM.RHNET/0777
SUBROUTINE RNGNET(JREGN, NREPEN, NUMP, 160, NUMR)

C

**************

C DEVELOP A REGIONAL MULTIDROP NETWORK, STARTING WITH A STAR
C NETWORK AND THEN OPTIMIZE IT AT ESAIL-WILLIAMS METHOD GIVEN
C THE FOLLOWING ARGUMENTS:

C

C JREGN= THE INDEX FOR THE REGION UNDER CONSIDERATION
C NREPEN= THE NUMBER OF SYSTEM TERMINATIONS IN REGION JREGN
C NUMP= AN ARRAY THAT CONTAINS INDICES FOR ALL SYSTEM
C TERMINATIONS IN REGION JREGN
C IGO= 1 IF NETWORK OPTIMIZATION IS TO BE PERFORMED
C

C NOTE: NODE AND SYSTEM TERMINATION ARE EXCHANGEABLE
C

C

**************

PARAMETER NP1=130, NP2=1, NP3=4, NP4=3
PARAMETER NPT=4
PARAMETER NCW=100, MW=4, NPC=360
PARAMETER NP6=1*NPC*NPC//2+NPC+1//4+1
COMMUNITY/CONST/NN1NP2NXNA+7HICITY
* /REF/REDF(NPC), TRAFD(NP1:2+NP2), DSTNCF(NP6), MAPADP(NP1)
* /LINC/ LINCNT(NP1), LINCNT(NP3), UTIL17(NP3)
* /INF/ IRATF(NP2NP2), IRAND(NP2C2), IFLAG(NP2NP3)
* /ETN/ SRT(NP1), RSC(MW), NUMPE(MW), TRAFN(NP1), TRAFIT(NP1)
* /NAME/HAMEST(NP1), LNAME(NP3), NAMEH(NP4)
* /SUM/ ASUM(N), ASUM
* /MUL/ AMUL, AMUL
* /RMD/ NTERMS, TIMPE, MPROC, MPLAYER
* /ADD/ IADD (NP1), KC, KCIH
* /RSC/ ROLIN(N), RSPTIM
DIMENSION COSTEW(NP1NP2NP3), IAPRAY(NP1NP2NP3), APAY(NP1NP2NP3)
DIMENSION TIMRSP(NP1NP2NP3), TRFSUM(NP1NP2NP3), TIMOUT(NP1NP2NP3)
DIMENSION HLINES(NP1NP2NP3), LDUWAY(NP3), NUMR(NP3)
DIMENSION IOST(NP1NP2NP3), LINCHW(NP1NP2NP3), LNP1LN(NP3NP3)
DIMENSION ICSTHW(NP1NP2NP3), ICSTLN(NP1NP2NP3), ITCST(NP1NP2NP3)
DIMENSION ISTR(NP1NP2NP3), ISTR, IPNUM(NP1NP2NP3), NUMIP(NP3)
DIMENSION IBLANK(NP1NP2NP3), ICHAR
DIMENSION NUMR(1)
DIMENSION PHOF(NP1NP1)
EQUIVALENCE (ICSTHW, ICHAR)
DATA JRLANK/* /
RSPTIM=0.
IPOINT=0
INTEG FP COST1, COST2, COST, C.FC.
C
C INITIALIZE COST ARRAY, INDEPENDENT OF INF TYPE
C
DO 399 K1=1NP1
DO 399 K2=1NP2
ICSTLN(K1NP3)=0
DO 399 K4=1NP4
ICSTHW(K1NP3NP3)=0
399 CONTINUE

C

NN1=RSC(JREGN) GLOBAL INDEX FOR RSC

A-16
C FIND THE LOCAL RSC INDEX IN THE REGION ARRAY

C

DO 9A IND=1,NOREG
   IF (NNI .EQ. NUMR(IND)) GOTO 199
9A CONTINUE
189 CONTINUE

IF (IND=1) GOTO 199
C

C BUILD A STAR NETWORK

C

CALL STARNW
C

C PRINT OUT STAR NETWORK

C

CALL SUMPRT(NOREG,1)

C OF DEVELOP A MULTIDROP NETWORK UTILIZING THE
C EGAU-WILLIAMS ALGORITHM

C

MAXSAV=0
MAXM=0
MAXL=0
MAXK=0
MAXKI=0
MAXLIN=0
MAXNOL=0
LINNEW=0
RSPMAX=0
RHOMAX=0
ICHAR=+
ITALY=0
JTALLY=0
KCHG=1
KADO=1
IK=0
INT=0
RETURN

C

C

C FORM THE INITIAL REGIONAL STAR NETWORK, IARRAY, AND FIND ITS
C COST, COSTW

C

INREGION=NUMBER OF SYSTEM TERMINATIONS IN THE REGION

C

***************

C

INTEGRAL COST

DO 110 K3=1,NOREG
   IF (K3=K4) GOTO 111
   IARRAY(K3+K4)=0
110 CONTINUE
111 K4=NUMR(K3)

C

***************
114 IARRAY(K3,1)=IAND(KK)
115 IARRAY(K3,5)=IRSC ! LOCAL INDEX FOR RSC
116 IARRAY(K3,1)=TARRAY(KK)
117 IARRAY(K3,2)=TARRAY(KK)
118 TEMPSP(K3)=0.
119 100 CONTINUE
120 IARRAY(IRSC+1)=NOREGN - 1 . NO. OF NODES UNDER RSC
121 NM=1 ! ASSUMING THE 1ST SUCCESSOR WITH INDEX 1
122 IF(IRSC.NE.1) NM=2 1ST SUCCESSOR IS WITH INDEX 2
123 IARRAY(IRSC+2)=NM
124 IARRAY(IRSC+5)=0
125 C
126 C RELATE ALL OF RSC'S SUCCESSORS
127 C
128 DO 200 K5=1, NOREGN
129 IF(K5.EQ. IRSC) GOTO 200
130 NM=NM + 1
131 C IF(NM .EQ. IRSC) NM=NM + 1
132 C RELATE ALL OF RSC'S SUCCESSORS
133 IARRAY(K3+1)=NM
134 IARRAY(NM+4)=K5
135 200 CONTINUE
136 C
137 C DETERMINE LINE TYPE FOR CENTRAL LINKS TO RSC
138 C
139 DO 550 NONF=1, NOREGN
140 IF(NODE .EQ. IRSC) GOTO 555
141 TFIN=IARRAY(NODE+1)+K5
142 TFOUT=IARRAY(NODE+2)+K5
143 NN2=NUMR(INODE)
144 DSTN=DIST(NN1,NN2)
145 IDST=NODE+1
146 COSTW=NODE+1
147 C
148 C TAKE A FIRST GUESS FOR LINE CONFIGURATION
149 C
150 COST=0
151 RHO=0.
152 MDROP=IARRAY(NODE+1)+1
153 CALL LINNUM(TFIN,TFOUT,LDUMMY,LINOLD,0,RHO)
154 781 CONTINUE
155 C
156 C COMPUTE INITIAL RESPONSE TIME
157 C
158 IKONT=0
159 DO 783 I=1,N3
160 IF(LUMMY(1).NE. 0) IKONT=IKONT+1
161 783 CONTINUE
162 AINTEF=TFIN
163 OUTTRF=TFOUT
164 IF(IKONT .EQ. 1) 11, LINOLD(LINOLD)+1) GOTO 772
165 C
166 C RESPONSE TIME CALCULATION NEEDS MODIFICATION
167 C
168 ACAP=0.
169 DO 771 NL=1,N3
170 ACAP=ACAP+LINCAP(NL)+LDUMMY(NL)
LNUMYM(NL) = 0
CONTINUE
LNUMYM(LINOLD) = 1
AINTRF = TRFIN + LINCAP(LINOLD) / ACAP
MUTRF = TRFOUT + LINCAP(LINOLD) / ACAP
CONTINUE
CALL RSPNSY(AINTRF, MUTRF, LINOLD, NODE, IOK)
IF (IOK = 0) GOTO 773
IF (LINOLD = FQ. N3) GOTO 774
CALL RNFUN(TRFIN, TRFOUT, LNUMYM, LINOLD, PHALT + RHO)
GOTO 781
CALL RHFUN(TRFIN, TRFOUT, LNUMYM + 1, LINOLD, PHALT + RHO)
GOTO 781
CONTINUE
TIMESP(NODE) = RSPTIM
KCHRF = 2
CALL SISMP(TRSC, NONF, 0, COST)
RMOF(NODE) = RHO
COSTW(NODE + 1) = COST
COSTW(NODE + 2) = LINOLD
DO 498 NL = 1, N3
IF (LNUMYM(NL) = 0) GOTO 498
ICSTLN(NODE + NL) = TCSTLN(NODE + NL) + LINCNH(NL + N)
DO 498 NK = 1, N4
ICSTHW(NODE + NK + N) = LNCCHW(NL + NK + N) + TCSTHW(NODE + NK + N)
CONTINUE
JTRAF = TRFIN + TRFOUT
JTRAF = JTRAF / UTILIZ(LINOLD)
COSTW(NODE + 3) = JTRAF / LINCAP(LINOLD) + 1
GOTO 555
CONTINUE
C
ASSUMING TRAFFIC AT TRSF IS TAKEN CARE OF AUTOMATICALLY
COSTFWS(NODE,4)=0
ROF(NODE)=0.
550 CONTINUE
RETURN
SUBROUTINE ISUMUP(L1,L2,LT,IC)

CALCULATE COST BETWEEN NODES L1 AND L2 AND ADD IT TO
TOTAL COST IC WHERE LT=LINE TYPE

LL1=NUMR(L1)
LL2=NUMR(L2)
CALL ICOSTJ(LNUMMY,LL1,LL2,LNKCHW,NKCN)
KK=3
IF(FLT .NE. 0) KK=1
DO 211 LINTYP=1, KK
LTYPE=LINTYP
IF(KK.EQ.1) LTYPE=LT
DO 221 I1=1,2
IC=IC+LNKCHW(LTYPE,II)
220 CONTINUE
211 CONTINUE
RETURN
SUBROUTINE ESSWL

TRY AGAIN TO OPTIMIZE THE NETWORK

K=1
500 CONTINUE
K=ARRAY(IENF+2) OF FIRST SUBNETWORK UNDER RSC
K=1=ARRAY(K+3) OF NEXT SUBNETWORK UNDER RSC
IF(K=1 .EQ. 0) GOTO 599
CONTINUE
560 CONTINUE
570 L=ARRAY(IENF+2) GK=SUBNET IS TO PF LINKED TO L=SUBNET
CONTINUE
570 CONTINUE
IF(L .NE. K) GOTO 575
571 L=ARRAY(L+3)
572 CONTINUE
573 K1=NUMR(K)
574 DIF=DIST(INI,K1)
575 C
C TEST TOTAL NO. OF TERMINALS IF K AND L ARE COMBINED
C
IMTRY=2 INDICATION OF ENTRY TO TRYLINK
LINE=COSTF(K+21)
IF(LINCAP(LINF) .GE. 9600) GOTO 585 AND MULTIDROPPING ON 9600
NOD=ARRAY(K)+1=ARRAY(L+1)+2
585 CONTINUE
IF(INDET .GT. NTERMS) GOTO 585 "TOO MANY TERMINALS
M=L
K=K
285  580   CONTINUE
286     MINSUM(M)
287     DTYR=DISt(K1,M1)/2.
288     IF(DTYR GT DEF) GOTO 140
289     CALL TYPLNK(K+K1+L+M) & M IS THE INSERTION NODE
290     IF(10K EQ. 0) GOTO 585
291     140 CONTINUE
292     NEXTNODE(KM) NEXT NODE UNDER M ON L-SRRNET
293     IF(KM NE. 0) GOTO 580 GHO MORE NODES UNDER M ON L-SRRNET
294     K=NEXTNODE(K+K1) QSTART WITH NEXT NODE ON K-SRRNET
295     IF(K EQ. 0) GOTO 585
296     K=MINK(KI)
297     MEL
298     GOTO 580
299     585 CONTINUE
300     NEXT = ARRAY(L+3) NEXT SUCCESSOR
301     IF(L NE. 0) GOTO 570
302     660 CONTINUE
303     K=ARRAY(L+3)
304     IF(K NE. 0) GOTO 560 QNOT AN END YET, REPEAT THE SEARCH
305     CALL ALL POSSIBLE COMBINATIONS HAVE BEEN TRIED
306     CALL IF(MAXSAV LE. 0) GOTO 599 QNO NEED TO GO FURTHER
307     CALL
308     UPDATE NETWORK BASED ON UP-TO-DATE MAXIMUM COST SAVING
309     CALL PARAMETERS
310     JTALLY=JTALLY+1
311     CALL UPNETWORK
312     CALL REINITIALIZATION
313     CALL RSPMAX=0.
314     MAXSAV=0
315     MAXK=0
316     MAXL=0
317     MAX=0
318     MAX=0
319     MAX=0
320     MAX=0
321     MAX=0
322     MAX=0
323     MAX=0
324     MAX=0
325     LINNEW=0
326     MAX=0
327     RHOMAX=0
328     GOTO 5500
329     599 CONTINUE
330     CALL PRINT OUT COSTS FOR THE OPTIMIZED MULTIDROP NETWORK
331     CALL CALL MULTIDROP
332     CALL PRINT OUT THE OPTIMIZED MULTIDROP NETWORK
333     CALL NFTPR
334     IF(NPLOT EQ. 1) GOTO 50
335     CALL CALPLT
336     50 CONTINUE
337     RETURN
SUBROUTINE TRYLNK(KL,KIL,LL,ML)

***************
C TRY TO ELIMINATE CENTRAL LINK KL AND LINK IT TO THE SUBNETWORK
C LL THROUGH SYSTEM TERMINATIONS KIL AND ML.
C
C ***************
INTEGER COSTKL,COST
ITALLY=ITALLY+1
IF(LINTRY .EQ. 1) GOTO 719
TFIN=ARRAY(KL)+ARRAY(ML)+0.5
TFOUT=ARRAY(KL)+ARRAY(ML)+0.5
C
C FIND THE LINE WITH THE ENOUGH CAPACITY TO HANDLE
C THE TOTAL TRAFFIC ON THE PROPOSED SUBNETWORK LL
C
CALL LINNWK(TFIN,TFOUT,DUMMY,LINNEW,RNH)
IF(LINCAP(LINNEW) .EQ. 9600) GOTO 132
LINUP=LINNEW-1
IF(LINUP .EQ. 0) GOTO 712
DO 711 NL=LINUP+1
     IF(LDUMMY(NL) .EQ. 0) GOTO 711
714 GOTO 132
711 CONTINUE
712 CONTINUE
NLNEW=DUMMY(LINNEW)
IF(LINNEW .EQ. 0) GOTO 132 MORE THAN 1 LINE NOT ALLOWED
C
C TEST RESPONSE TIME; IF NOT SATISFIED, INCREASE LINE CAPACITY
C
CALL RSPT3(TKL,LL,LINNEW,INX)
IF(INX .EQ. 2) GOTO 3001
C
C IF LINE TYPE IS THE HIGHEST, NO NEED TO GO FURTHER
C
IF(LINNEW .EQ. N3) GOTO 132
LINNEW=LINNEW+1
IF(LINCAP(LINNEW) .EQ. 9600) GOTO 132
LDUMMY(LINNEW)=1
NLNEW=1
CALL RHOFUN(TFIN,TFOUT,DUMMY,LINNEW,RHOLIN,RHO)
GOTO 3000
3001 CONTINUE
IF(LINNEW .EQ. LINOLD,AND,NLOLD .EQ. 1) GOTO 131
CALL LCOSTK(IRSC,LL,1,MCOST) NEW COST FOR SUBNET UNDER LL
131 CONTINUE
LINOLD=MCOST(KL)
MCOST=MCOST(KL)
NLOLD=MCOST(KL)
IF(LINNEW .EQ. LINOLD,AND,NLOLD .EQ. 1) GOTO 133
CALL LCOSTK(IRSC,KL,0,MCOST) NEW COST FOR SUBNET UNDER KL
INTEGER TCO5T
TCOST=0
KCHG=2
CALL ISUMUP(1,NA,NA,T0ST)
C START COMPUTING SUBNET COST
C J5AN=1ARRAY(NA+2)  © FIRST SUCCESSOR
443 IF (JSAN=1, EQ, 0) GOTO 400
444 CONTINUE
445 JPA=ARRAY(J5AN+5)
446 CALL ISUMUP(JPA,J5AN,NA,TCOST)
447 JSAN=XTNODI(NA,J5AN)
448 IF (JSAN+EQ, 0) GO TO 400  © CALL IT AN END
449 GO TO 300
450 CONTINUE
451 IF (NB+EQ, 1) RETURN
452 ITEMP=0
453 KADD=0
454 CALL ISUMUP(1,NA,NA,T0ST)
455 TCOST=TCOST+ITEMP
FUNCTION NXTNOD(L1,M1)

**************

C FIND THE NEXT NODE IN THE SUBNET L1 WHICH M1 BELONGS TO.
C IN THE PROCESS; IF THE NEXT NODE IS L1, 0 IS RETURNED
C OTHERWISE THE NEXT NODE IS RETURNED.

C **************

NXTNOD=0
M1=M1
KSON=IARRAY(MM+2)
IF(KSON.EQ.0.AND. MM.EQ. L1) RETURN 0A SINGLE NODE
IF (KSON.EQ. 0) GO TO 1 0 NO SUCCESSOR
NXTNOD=KSON
RETURN
1 CONTINUE
C LOOK FOR HIS NEXT BROTHER
KBRD=IARRAY(MM+3)
IF (KBRD.EQ. 0) GO TO 2 0 NO MORE SUCCESSORS WITH SAME PREDCESSOR
NXTNOD=KBRD
RETURN
2 CONTINUE
C GO TO HIS FATHER
MM=IARRAY(MM+5)
IF (MM.NE. L1) GO TO 1 0 BACK TO THE BEGINNING
RETURN
SIRoutine UPNETW

**************

C UPDATE IARRAY AND COSTEW BASED ON MAXIMUM SAVING
C PARAMETERS OBTAINED
C UPDATE TRAFFIC AND NO. OF TERMINALS FOR L-SUBNET
C

**************

NOK=IARRAY(MAXK+1)+1 0 NO. OF NODES BELOW MAXK
IARRAY(MAXK+1)=IARRAY(MAXK+1)+NOK
ARRAY(MAX1+1)=ARRAY(MAX1+1)+ARRAY(MAXK,1)
ARRAY(MAX1+2)=ARRAY(MAX1+2)+ARRAY(MAXK+2)

C UPDATE THE COSTEW
C
C COSTEW(MAX1+1)=COSTEW(MAX1+1)+COSTF(MAX1+1)-MAXSAV
C COSTEW(MAX1+2)=MAXLIN
C COSTEW(MAX1+3)=MAXNOL
C COSTEW(MAXK+1)=0
C COSTEW(MAXK+2)=0
C COSTEW(MAXK+3)=0
MAXK=NUMR(MAXK)
MAXK=NUMR(MAXK)
MAXK=NUMR(MAXK)
MAXK=NUMR(MAXK)

C COSTEW(MAX1+4)=COSTEW(MAX1+4)+COSTF(MAX1+4)+DIST(MAXK+1,MAXK+1)

A-24
513  * -DIST(MAXN+NN1)
514  RHOF(MAXL)=Rhomax
515  COSTW(MAXK+4)=0
516  C
517  C UPDATE MULTIDROPPED-LINE RESPONSE TIME
518  C
519  TIPRESP(MAXL)=RESPMAX
520  CONTINUE
521  KIPA=ARRAY(MAXK+5) KREWHEN K1'S PREDECESSOR
522  MSON=ARRAY(MAXK+2) KM'S 1ST SUCCESSOR
523  CALL LNKOFF(MAXK) GEND KI AS A SUCCESSOR OF KIPA
524  ARRAY(MAXK+2)=MAXK
525  ARRAY(MAXK+5)=MAXK
526  ARRAY(MAXK+3)=MSON
527  IF(MSON .NE. 0) ARRAY(MSON+4)=MAXK
528  ARRAY(MAXK+4)=0
529  MAXK=MAXK
530  MAXK=KIPA
531  IF(MAXK .NE. MAXK) GOTO 91
532  RETURN
533  FUNCTION JCOSTA(NKREF)
534  C ***************
535  C
536  C FIND PARTIAL SUM FOR ICST1N
537  C
538  C ***************
539  JCOSTA=0
540  DO 777 KI=1,NKREF
541  JCOSTA=JCOSTA+ICSTLN(KI+1)
542  777 CONTINUE
543  RETURN
544  FUNCTION JCOSTR(NM,KREF)
545  C ***************
546  C
547  C FIND PARTIAL SUM FOR ICSTHW
548  C
549  C ***************
550  JCOSTR=0
551  DO 77A KK=1,NKREF
552  JCOSTR=JCOSTR+ICSTHW(KK+1)
553  77A CONTINUE
554  RETURN
555  SUBROUTINE NETPRT
556  C ***************
557  C
558  C PRINT OUT CONFIGURATION OF THE MULTIDROP NETWORK
559  C
560  C ***************
561  DO 196 KK=1,NP1
562  IBLANK(KK)=JBLANK
563  196 CONTINUE
564  NIP=SUMMAR(IRSC)
565  WRITE(NINT,197) N1,N2
566  197 FORMAT(1HL,4,TPNATIONAL CENTER=**A4///**X,*SUBNETWORK***+**X)
567  *BEGIN'S AT///)
568  KP=1
569  ISON=ARRAY(IRSC+2)
77-53, Vol. IV

570 IPOINT=ISON
571 ISON=NUMR(ISON)
572 WRITE(192) IRLANK(I),I=1,KP),ISONR
573 C
574 C LOOK FOR ITS FIRST SUCCESSOR
575 C
576 190 CONTINUE
577 ISON=ARRAY(IPOINT+2) CURRENT NORMAL INDEX
578 IF(IPOINT+EQ.0) GOTO 191 AND MORE SON
579 KP=KP+1 GA LEVEL DEPHER
580 ISON=NUMR(ISON)
581 WRITE(192) IRLANK(I),I=1,KP),ISONR
582 192 FORMAT(1X,N2A6)
583 IPOINT=ISON
584 GOTO 190
585 191 CONTINUE
586 C
587 C LOOK FOR NEXT SUCCESSOR WITH THE SAME PREDECESSOR
588 C
589 IRRO=ARRAY(IPOINT+3)
590 IF(IRRO+EQ.0) GOTO 193
591 IFRO=NUMR(IRRO)
592 WRITE(192) IRLANK(I),I=1,KP),IRRO
593 IPOINT=IRRO
594 GOTO 190
595 192 CONTINUE
596 CONTINUE
597 C NEXT LEVEL UP
598 C
599 KP=KP-1
600 IPOINT=ARRAY(IPOINT+4)
601 IF(KP+EQ.0) GOTO 194 AND NFEN TO GO FURTHER
602 GOTO 191
603 194 CONTINUE
604 RETURN
605 SUBROUTINE CONVRT(ICOST)
606 ***************
607 C
608 C CONVERT A NUMBER INTO ITS FIELD EQUIVALENT
609 C
610 ***************
611 JCCHAR(1)=JRLANK
612 JCCHAR(2)=JRLANK
613 IF(ICOST+EQ.0) GOTO 916
614 ENCOE(198,JCCHAR) ICOST
615 198 FORMAT(1B)
616 916 CONTINUE
617 RETURN
618 SUBROUTINE SUMPR(INREF,NN)
619 ***************
620 C
621 C SUM UP COSTS AND PRINTS
622 C
623 ***************
624 TCOST1=0
625 TCOST2=0
626 DO 779 K=1,INREF

A-26
**77-53, Vol. IV**

627 \[ \text{ITCOST}(k,1) = \text{ICSTLN}(k+1) \]
628 \[ \text{ITCOST}(k,2) = \text{ICSTLN}(k+2) \]
629 \[ \text{DO } 7791 \text{ \texttt{K}=1+\texttt{N}} \]
630 \[ \text{ITCOST}(k,1) = \text{ITCOST}(k,1) + \text{ICSTW}(k,k+1) \]
631 \[ \text{ITCOST}(k,2) = \text{ITCOST}(k,2) + \text{ICSTW}(k,k+2) \]
632 **7791 CONTINUE**
633 \[ \text{TCOST1} = \text{ITCOST1} + \text{ITCOST}(k,1) \]
634 \[ \text{TCOST2} = \text{ITCOST2} + \text{ITCOST}(k,2) \]
635 **779 CONTINUE**
636 \[ \text{KCOST} = \text{TCOST1} + \text{TCOST2} \]
637 *C*
638 **C PRINT OUT COST**
639 *C*
640 \[ \text{NTURN} = \text{NREF}/10+1 \]
641 \[ \text{IREM} = \text{MOD}(\text{NREF}, 10) \]
642 \[ \text{IF}(\text{IREM} \neq 0) \text{ \texttt{NTURN} = NTURN-1} \]
643 \[ \text{LPAGE} = 1 \]
644 \[ \text{DO } 919 \texttt{KW} = 1, \text{NTURN} \]
645 \[ \texttt{KW} = \text{ID}+1 \]
646 \[ \text{KW} = \text{ID}+\texttt{KW}+1 \]
647 \[ \text{KW} = 10+K \]
648 \[ \text{IF}(\text{ME} = \text{GT}(\text{NREF}) \texttt{KW} = \text{NREF} \]
649 \[ \text{IF}(\text{N} \neq 1) \text{ GOTO 979} \]
650 \[ \text{IF}(\text{LPAGE} \neq 1) \text{ GOTO 9033} \]
651 **9031 FORMAT(1100X, 'REGIONAL STAR NETWORK AND ITS COSTS =', I0)**
652 \[ \text{GOTO 9035} \]
653 **9033 CONTINUE**
654 \[ \text{WRITE(IINT,9031) KW} \]
655 **9034 FORMAT(1100X, 'REGIONAL STAR NETWORK AND ITS COSTS =', I0)**
656 **9035 CONTINUE**
657 \[ \text{WRITE(IINT,9032) (NUMRR(I), I=KWL,KWU)} \]
658 **9032 FORMAT(1/IX*, 'SYSTEM TFRMN=', 13, 10(4X, A9, 1X))**
659 \[ \text{WRITE(IINT,9033) KW} \]
660 **903 FORMAT(1100X, 'NO. OF LINES REGION:')**
661 \[ \text{DO } 903 \texttt{NJ} = 1+3 \]
662 \[ \text{IF}(\text{LNMIX(NJ) EQ} 0) \text{ GOTO 190A} \]
663 \[ \text{WRITE(IINT,904) LNAME(NJ), (NRLINES(K,NJ), K=KWL,KWU)} \]
664 **904 FORMAT(1100X, 'LINE:')**
665 \[ \text{WRITE(IINT,9036) (RHOF(NJ), NJ=KWL,KWU)} \]
666 **9036 FORMAT(1100X, 'LINE UTILIZATION=', 11, 10(PR,3.1))**
667 \[ \text{WRITE(IINT,906) (DIST(N), N=KWL,KWU)} \]
668 **906 FORMAT(1100X, 'DISTANCE FROM RSC=', 11, 10(4X, A9, 1X))**
669 \[ \text{GO TO 806} \]
670 **879 CONTINUE**
671 \[ \text{IF}(\text{LPAGE} \neq 1) \text{ GOTO 8033} \]
672 \[ \text{WRITE(IINT,8031) KW} \]
673 **8031 FORMAT(1100X, 'FINAL MULTIDROP NETWORK AND ITS COSTS =', I0)**
674 \[ \text{GOTO 8035} \]
675 **8033 CONTINUE**
676 \[ \text{WRITE(IINT,8034) KW} \]
677 **8034 FORMAT(1100X, 'FINAL MULTIDROP NETWORK AND ITS COSTS =', I0)**
678 **8035 CONTINUE**
679 \[ \text{WRITE(IINT,8031) (I, I=KWL,KWU)} \]
680 **803 FORMAT(1100X, 'SUMMARY NO.=', 16, 10(4X, A9, 1X))**
681 \[ \text{DO } 1803 \texttt{NSUR} = KWL,KWU \]
682 \[ \text{NSUR} \]
683 **A-27**
77-53, Vol. IV

684     MSUB(N)=NUMAR(N)
685     LSUB(N)=IARRAY(ID,N)
686 1804 CONTINUE
687     WRITE(IWT,1806) (MSUB(N),N=KWL,KWU)
688 1806 FORMAT(1X*BEGINNING NODE,1X*10(X,A6,1X))
689     WRITE(IWT,1807) (LSUB(N),N=KWL,KWU)
690 1807 FORMAT(3X*NO. OF TERMS,12X*10(IN,1X))
691     WRITE(IWT,6811)
692 6811 FORMAT(3X*NO. OF LINES)
693     DO 1808 NI=1,N3
694     IF(LINIMX(NI).EQ.0) GOTO 1808
695     WRITE(IWT,6804) LINAME(NI) *(NLINES(NI,NJ),K=KWL,KWU)
696 1808 CONTINUE
697     WRITE(IWT,8036) (RHOF(NJ),NJ=KWL,KWU)
698 8036 FORMAT(3X*LINE UTILIZATION,9X*10(F8.3,1X))
699     WRITE(IWT,808) (IDST(N),N=KWL,KWU)
700 808 FORMAT(3X*TOTAL MILEAGE,12X*10(IN,1X))
701 806 CONTINUE
702     DO 1101 N=KWL,KWU
703     ID=N
704     IF(NN+F0.0) INNSUB(N)
705     TRFSUM(N,1)=ARRAY(ID,1)
706     TRFSUM(N,2)=ARRAY(ID,2)
707     TIMOUT(N)=TIMSP(ID)
708 1101 CONTINUE
709     WRITE(IWT,1102) (TRFSUM(N,1),N=KWL,KWU)
710 1102 FORMAT(3X*TRAFFIC,9X*LINE TO CPU,11X*10(F8.3,1X))
711     WRITE(IWT,1103) (TRFSUM(N,2),N=KWL,KWU)
712 1103 FORMAT(3X*CPU TO LINE,9X*10(F8.3,1X))
713     WRITE(IWT,1104) (TIMOUT(N),N=KWL,KWU)
714 1104 FORMAT(3X*LINE RESPONSE TIME,7X*10(F8.3,1X))
715     WRITE(IWT,907)
716 907 FORMAT(3X*SUBTOTAL,9X*INST. COSTS)
717     COST=JCOSTA(I,NREF)
718     IF(KW.NE.1) COST=0
719     CALL CONVRT(COST)
720     WRITE(IWT,908) (JCHAR(L),L=1,2), (ICSTLN(NODE,L),NODE=KWL,KWU)
721 908 FORMAT(5X*LINES,8X,A6,A2,1X,10(IN,1X))
722     DO 1909 K=1,N4
723     COST=JCOSTB(K,NREF)
724     IF(KW.NE.1) COST=0
725     CALL CONVRT(COST)
726     WRITE(IWT,909) NAMEHW(K),(JCHAR(L),L=1,2), (ICSTHW(NODE,K),NODE=KWL,KWU)
727     * NODE=KWL,KWU
728 909 FORMAT(5X*A6,7X*A6,A2,1X,10(IN,1X))
729 1909 CONTINUE
730 1909 CONTINUE
731 910 FORMAT(3X*ANNUAL RECURR. COST)
732     COST=JCOSTA(2,NREF)
733     IF(KW.NE.1) COST=0
734     CALL CONVRT(COST)
735     WRITE(IWT,908) (JCHAR(L),L=1,2), (ICSTLN(NODE,K),NODE=KWL,KWU)
736     DO 1911 K=1,N4
737     COST=JCOSTB(K,NREF)
738     IF(KW.NE.1) COST=0
739     CALL CONVRT(COST)
740     WRITE(IWT,909) NAMEHW(K),(JCHAR(L),L=1,2), (ICSTHW(NODE,K),NODE=KWL,KWU)

A-28
77-53, Vol. IV

741 * NODE=KW,KW
742 191 CONTINUE
743 WRITE(IWT,912)
744 912 FORMAT(1X,TXT)
745 IF(KW.NE.1) TCOST=0
746 CALL CONVRT(TCOST)
747 WRITE(IWT,913) (JCHAR(L),L=1,2),(ITCOST(KI),K=KW,KW)
748 IF(KW.NE.1) TCOST=2
749 CALL CONVRT(TCOST)
750 WRITE(IWT,914) (JCHAR(L),L=1,2),(ITCOST(KI),K=KW,KW)
751 913 FORMAT(4X,TXT)
752 914 FORMAT(4X,RECUT, COST, 3X, A6, A2, 1X, 10(I9,1X))
753 LPAGE=LPAGE+1
754 LPAGE=MOD(LPAGE,2)
755 919 CONTINUE
756 WRITE(IWT,695) K
757 695 FORMAT(/2X, 15X, TOTAL COST=, I8)
758 RETURN
759 ** SUBROUTINE MUTDRP **
760 C
761 C PRINT OUT FINAL MULTIDROP NETWORK WITH ITS COSTS
762 C
763 696 CONTINUE
764 697 DO 590 NL=1,N3
765 698 LDUMMY(NL)=0
766 699 CONTINUE
767 690 CONTINUE
768 IBRO=IARRAY(IRO, 2) @FIRST SUCCESSOR
769 K=1
770 699 CONTINUE
771 IF(IBRO.EQ. 0) GOTO 698
772 NK=NUMR(IBRO)
773 NK1=NN1
774 NS1=IARRAY(K1)=IBRO
775 L1=ICOSTE(IBRO, 2)
776 LDUMMY(L1)=ICOSTE(IBRO, 3)
777 JS1=IARRAY(IBRO, 2)
778 IF(JS1.EQ.0) GOTO 694
779 DO 592 NK=1,12
780 ICOSTL(K1,NK)=0
781 DO 592 NK=1,N4
782 ICOSTH(K1,NK)=0
783 592 CONTINUE
784 DO 596 NL=1,N3
785 NLINES(K1,NL)=LDUMMY(NL)
786 596 CONTINUE
787 KCHG=2
788 312 CONTINUE
789 CALL IOST(J(LDUMMY,K1,NK, NLKNCH+1 NLKL))
790 DO 595 NL=1,N3
791 DO 595 NM=1,2
792 ICOSTL(NK1,NM)=ICOSTL(K1,NM)+L1KCL(NL,NM)
793 DO 595 NK=1,N4
794 ICOSTH(K1,NK)=ICOSTH(K1,NK)+L1KCL(NL,NK)
795 595 CONTINUE
796 IF(K1.EQ.0) GOTO 311
797 NK2=NUMR(K1) @GLOBAL INDEX FOR NEXT NODE

A-29
77-53, Vol. IV

798     NK1=IARRAY(JSON+5) QPREDCELERSOR
799     NK1=NUM(NK1) @GLOBAL INDEX FOR PREDCELERSOR
800     JSON=INDEX(IRO+JSON)
801     GOTO 312
802     311 CONTINUE
803     LDUMMY(LINE)=0
804     GOTO 591
805     694 CONTINUE
806     C
807     C USE PREVIOUS DATA
808     C
809     DO 597 NL=1,N3
810        NLINES(K1+NL)=NLINES(IRO+NL)
811     597 CONTINUE
812     DO 598 NM=1,2
813        ICSTLN(K1+NM)=ICSTLN(IRO+NM)
814     598 CONTINUE
815     ICSTHN(K1+NM)=ICSTHN(IRO+NM)
816     598 CONTINUE
817     LDUMMY(LINE)=0
818     591 CONTINUE
819     KI=K1+1
820     GOTO 699
821     CONTINUE
824     DO 598 NL=2,N3
825        NOSUB=K1-1
826        CALL SYNPRAT(NOSUB,0)
827        RETURN
828        SUBROUTINE CALPLT
829        ***************
830        C
831        C PLOT A MULTIDROP NETWORK
832        C
833        C ***************
834        KP=1
835        IPOINT=IRSC
836        CALL TRANSFORM(2)
837        ISON=IARRAY(IRO+2) @FIRST SUCCESSOR
838        IPOINT=ISON
839        CALL TRANSFORM(1)
840        C
841        C LOOK FOR ITS FIRST SUCCESSOR
842        C
843        190 CONTINUE
844        ISON=IARRAY(IPOINT+2) @FIRST SUCCESSOR
845        IF(IPOINT .EQ. 0) GOTO 191
846        KP=KP+1
847        IPOINT=ISON
848        CALL TRANSFORM(1)
849        GOTO 190
850        191 CONTINUE
851        C
852        C LOOK FOR ITS NEXT SUCCESSOR
853        C
854        IRO=IARRAY(IPOINT+3)
855  IPOINTERIALP(IPOINTER+5) NOW ITS PREDECESSOR
856  CALL TPSFP(2)
857  IF(IBRO .EQ. 0) GOTO 193
858  IPOINTERIALBRO
859  CALL TNSFRM(1)
860  GOTO 190
861  CONTINUE
862  C
863  C GO BACK TO ITS PREDECESSOR
864  C
865  KP=KP+1
866  IF(KP .EQ. 0) GOTO 194
867  GOTO 191
868  CONTINUE
869  CALL TNSFRM(3)
870  RETURN
871  SUBROUTINE TRSFRM(LK)
872  **********************
873  C
874  C FIND GLOBAL MADADR INDEX FOR V-H COORDINATES AND PID NO.
875  C
876  **********************
877  DATA IP/0/
878  IF(LK .EQ. 3) GOTO 666
879  LKI=NUMR(IPOINTER) GLOBAL INDEX
880  IDD=MAPADR(LKI1) MAPADR INDEX FOR LK1
881  IF(IOD .EQ. IP) RETURN
882  IP=IDD
883  CONTINUE
884  CALL PLOTPT(IDD,LK)
885  RETURN
886  SUBROUTINE LNKOFF(MP)
887  **********************
888  C
889  C DELETE MP AS A SUCCESSOR OF NONE PA
890  C
891  **********************
892  IFRONT=IARRAY(MP+4) THE SUCCESSOR REFORF MP
893  IRAK=IARRAY(MP+3) THE SUCCESSOR AFTER MP
894  IF(IFRONT .NE. 0) GOTO 92
895  MP=IARRAY(MP+5)
896  IRAK=IARRAY(MP+2) IRAK 1ST SUCCESSOR UNDER NEW MPA
897  GOTO 99
898  CONTINUE
899  IARRAY(IFRONT+3)=IRAK
900  CONTINUE
901  IF(IIRAK .EQ. 0) RETURN
902  IARRAY(IRACK+4)=IFRONT
903  RETURN
904  SUBROUTINE RSPTST(KKK,LLL,LINMAX,IOK)
905  **************************
906  C
907  C TEST RESPONSE TIME SATISFIED WHEN IOK=1
908  C
909  **************************
910  MDROP=IARRAY(LLL+1)+IARRAY(KKK,1)+2
911  TRFIN=IARRAY(LLL+1)+ARRAY(KKK,1)
912  TRFOUT=IARRAY(LLL+2)+ARRAY(KKK,P)
913  CALL RSPTSE(TRFIN,TRFOUT,LINMAX,MDROP,IOK)
914  RETURN
915  END

QPRST STACOM.1RNO.0/777

A-31
**SUBROUTINE IRNOP(NR, LIMIT, TRM)**

*****************

**SUPROROGAM FOR THE INTER-REGION NETWORK OPTIMIZATION**

**LIMIT: MINIMAL NUMBER OF PATHS NEEDED PER REGIONAL SWITCHING CENTER**

*****************

PARAMETER NP1=130, NP2=1, NP3=4, NP4=3
PARAMETER NPC=360, NP6=(NPC+NP3)/2
PARAMETER NP7=100, MW=4
COMMON/CONST/NP1,NP2,NP3,NP4,NP5,NP6,NP7
COMMON/LINCA/IN[X1P3, IN[1XPT(NP1), NAME(NP3)]
COMMON/LINCA/IN[1XPT(NP1), NAME(NP3)]
COMMON/LINCA/IN[1XPT(NP1), NAME(NP3)]
COMMON/LINCA/IN[1XPT(NP1), NAME(NP3)]

**DIMENSION NETSUM(NP3,2), ORINET(MW, MW, NP3)**

**DIMENSION NLINK(NP3), LNCH(NP3+NP4, 4)**

**DIMENSION RHOF2(MW, MW), TRRM(NR)**

**EQUIVALENCE (LINEQ1, LINEQ1) (LINEQ2, LINEQ2)**

**C RESET UTILIZATION FACTOR TO .5**

**DO 70 NN1=1, N3**

**UTILIZ(NN1)=.5**

**70 CONTINUE**

**C COMPUTE ORINET(MW, MW, N3) FOR INITIAL TOPOLOGY WHERE N3 IS**

**THE NUMBER OF CHARGEABLE ITEMS**

**C ORICT=0**

**ORICS=0**

**ORICS2=0**

**NR=NR-1**

**DO 203 NN1=1, N3**

**DO 203 NN2=1, 2**

**203 CONTINUE**

**C MODIFY DUPLING MODE FROM HALF TO FULL DUPLIFY**

**DO 667 K1=1, N3**

**IDUPLY(K1)=2**

**667 CONTINUE**

**DO 101 I=1, NR1**
NLINK(I) = NR1  OR1 LINKS AT THE BEGINNING
I1 = I + 1
DO 102 J = I1, NR
II = NRS(1)
JUNRS(J)
ATMAX = MAX1(1R(I, J), 1R(J, I))  DASIMMING FULL DUPLEX
CALL LINNUM (ATMAX, O, LINFO, LINUP, O, RHO)
RHOF(I, J) = RHO
RHOF(J, I) = RHO
CALL TCOST(J)(LINEQ(I, J), LNKCHW(L, DLM))
DO 104 NN = 1, N3
ORINET(I, J, NN) = LINFO(11N)
ORINET(I, J, NN) = LINFO(II)
DO 105 NN = 1, N3
NLIN = INLKCHW(L, INN)
NLIN = INLCHW(L, INN)
DO 106 NK = 1, N4  OR HARDWARE COSTS
NLIN = INLKCHW(L, INN)
NLIN = INLCHW(L, INN)
104 CONTINUE
105 CONTINUE
106 CONTINUE
107 CONTINUE
CALL OUTPR1(I)
CALL OUTPR1(I)
999 CONTINUE
MAXSAV = 0
DO 777 II = 1, NR
IF (NLINK(I) .LE. LIMIT) GO TO 777
II = I + 1
DO 788 JII = I1, NR
IF (NLINK(J) .LE. LIMIT) GO TO 788
IN = INLKCHW(L, II, J)
IF (IN .EQ. 0) GO TO 788  AND LINK TO BE DELETED
788 CONTINUE
DO 137 NN = 1, N3
IF (L .EQ. N) GO TO 137
IN = INLKCHW(L, N)
IF (IN .EQ. 1) GO TO 138
137 CONTINUE
DO 139 L = 1, NR
L1 = L + 1
DO 138 N = 1, N3
IN = INLKCHW(L, N)
IF (IN .EQ. 1) GO TO 138
138 CONTINUE
DO 139 NR = 1, NR
DO 138 N = 1, N3
IN = INLKCHW(N, NR)
DO 138 N = 1, N3
DO 139 NR = 1, NR
IF (IN .EQ. 0) GO TO 137
C DETERMINE WHETHER THERE IS A LINK CONNECTED BY AT MOST ONE INDIRECT
C ROUTE BETWEEN ANY TWO REGIONS IN THE NETWORK WHEN THE DIRECT LINK
C BETWEEN I AND J IS ELIMINATED. THE INDIRECT LINK ONLY GOES THROUGH
C ONE INTERMEDIATE RSC.
C
DO 139 L = 1, NR
L1 = L + 1
DO 138 N = 1, N3
IF (NY .EQ. 1) GO TO 138
IN = INLKCHW(N, L)
IF (IN .EQ. 1) GO TO 138
139 CONTINUE
138 CONTINUE
DO 139 NR = 1, NR
DO 138 N = 1, N3
C NEXT STEP NOT TO BE TESTED
C
C NEXT STEP NOT TO BE TESTED
114 IY=IY + 100
115 IF(IY .EQ. 1) GOTO 120
116 IF(IY .EQ. 1) GOTO 120
117 IF (IY .EQ. 1) GOTO 120
118 120 CONTINUE
119 GO TO 788
120 120 CONTINUE
121 121 CONTINUE
122 122 CONTINUE
123 CALL TRFDIV(IFLOP)
124 IF (IFLOP .EQ. 1) GOTO 201
125 CALL MINAD(I1AD,MINCST)
126 GO TO 202
127 201 CONTINUE
128 CALL NETWKC(MINCST)
129 202 CONTINUE
130 ISAV=ORICST-MINCST
131 IF (MAXSAV .GE. ISAV) GOTO 78A
132 MAXSAV=ISAV
133 IFLOP=IFLOP + Global Indicator
134 IF (IFLOP .EQ. 1) GOTO 204
135 IMAX=I
136 JMAX=J
137 DO 666 NN=1,N3
138 ILINAD(NN)=LINAD(TT,NN) CHANGE OF LINE RFG.
139 JLINAD(NN)=LINADJ(NN)
140 666 CONTINUE
141 DO 331 K1=1, NR
142 DO 331 K2=1, NR
143 331 CONTINUE
144 331 CONTINUE
145 TRRM(K1,K2)=TRR(K1+K2)
146 CALL NETUP(IFLIP,MAXSAV,IMAXON,MAXSAV)
147 ITALL=ITALL+1
148 GO TO 999
149 999 CONTINUE
150 IF (MAXSAV .LE. 0) GOTO 999
151 CALL NETUP(IFLIP,IMAX,IMAX,MAXSAV)
152 ITALL=ITALL+1
153 GO TO 999
154 999 CONTINUE
155 109 FORMAT(6x,'THIS NETWORK HAS BEEN UPDATED FOR ',ITIMES,' TIMES')
156 WRITE(6,109) ITALLY
157 DO 81 I=1,N3
158 DO 81 J=1,N3
159 NETSUM(I,J)=0
160 81 CONTINUE
161 DO 91 I=1,NR1
162 K=I+1
163 DO 92 J=K,NR1
164 DO 93 K=1,N3
165 LINEO(K1)=ORINET(I,J,K1)
166 93 CONTINUE
167 93 CONTINUE
168 CALL ICOSTJ(LINEO,II,JJ,LNKCHW,LNKCHN)
169 DO 94 K1=1,N3
170 94 CONTINUE
**Subroutine OUTPRT**

```
SUBROUTINE OUTPRT
** *************s*** B**
C PRINT OUT INTERREGIONAL NETWORK CONFIGUATION AND ITS COSTS
C ** ***************
C
DO 110 I=1,N3
ORICSI=ORICSI+NETSUM(I+1)
ORICSS=ORICSS+NETSUM(I+2)
110 CONTINUE
NTURN=NR/10+1
DO 200 I=1,NTURN
LL=(NTURN-1)*10 + 1
LU=NTURN*10
IF(LU .GE. NR) LU=NR
IF(N .GT. 3) GOTO 2100
WRITE(LWT,2021) (J,J=LL=LU)
2002 FORMAT(11,10X,INITIAL INTERREGIONAL NETWORK CONFIGUATION)
2003 GOTO 2101
2100 CONTINUE
2102 FORMAT(11,10X,FINAL OPTIMAL INTERREGIONAL NETWORK CONFIGUATION)
2103 GOTO 2101
2101 CONTINUE
DO 203 I=1,NR
WRITE(LWT,2004) I
2004 FORMAT(15 REGION 144)
DO 210 M=1,N3
WRITE(LWT,2006) LINAME(M),ORIC(S)(I,K+M),K=LL=LU
2006 CONTINUE
210 A CONTINUE
210 B FORMAT(15X,REGION 144)
DO 220 K=1,N3
WRITE(LWT,2201) (RHOF(I,J),J=LL=LU)
2201 FORMAT(15X,LINE UTILIZATION),I=144
220 CONTINUE
220 A CONTINUE
220 B FORMAT(15X,REGION 144)
DO 230 K=1,N3
WRITE(LWT,2307) K,LINAME(K),NETSUM(K),I=1,K+P
2307 FORMAT(15X,REGION 144)
230 CONTINUE
230 A CONTINUE
230 B FORMAT(15X,REGION 144)
DO 240 K=1,N3
WRITE(LWT,2401) (K,LINAME(K),I=1,K+P)
2401 FORMAT(15X,REGION 144)
240 CONTINUE
240 A CONTINUE
240 B FORMAT(15X,REGION 144)
DO 250 K=1,N3
WRITE(LWT,2507) K,LINAME(K),I=1,K+P
2507 FORMAT(15X,REGION 144)
250 CONTINUE
250 A CONTINUE
250 B FORMAT(15X,REGION 144)
PRINT COST=ORICSI+ORICSS
```

**END**
function TEST(I,J,K,L)

***************

IF (I .EQ. K .AND. J .EQ. L) TEST = 1
IF (I .EQ. L .AND. J .EQ. K) TEST = 1
RETURN

***************

DIMENSION NET(MW,MW,NP3)

DO 103 II = 1, NR
IF (NET(I,J,II) .GT. 0) GO TO 10A
103 CONTINUE
NET = 0  \! NO CONNECTION
RETURN
10A TEST = 1
RETURN \! YES, THERE IS A CONNECTION

SUBROUTINE TRFDIV(I,FLOP)

***************

C DIVERT TRAFFIC BETWEEN I AND J THROUGH OTHER RCS.

C ***************

C IT RETURNS WITH FLOP=1 WHEN SUCCESSFUL, OTHERWISE FLOP=0.
C IT ALSO CREATES TEMPORARY MATRICES TAR AND NETCF.

C

SNDIVT = 0.  TOTAL TRAFFIC DIVERTED (I TO J)
SDIVTJ = 0.  TOTAL TRAFFIC DIVERTED (J TO I)

DO 205 K = 1, NR
DIVRI(K) = 0.  \! TRAFFIC DIVERTED THROUGH REGION K (I TO J)
DIVRJ(K) = 0.  \! TRAFFIC DIVERTED THROUGH REGION K (J TO I)

CONTINUE

DO 220 II = 1, NR
IF (II .EQ. I .OR. II .EQ. J) GO TO 220
IC1 = TEST(ORINET; II)
IC2 = TEST(ORINET; II)
IF (IC1 .EQ. 0 .OR. IC2 .EQ. 0) GOTO 220

C DIVERT I TO J TRAFFIC THROUGH II

DIVRI(II) = 0.
DIVRJ(II) = 0.

CALL LINTRF(I,II)

DELTP = A-TPR(I,II)
IF (DELTP .LE. 0.0) GO TO 160

C
285 DIVTR=DELTR
286 CALL LINTRF(II,J+1)
287 DFLTR=A-TRRI,II)
288 IF (DFLTR.LE.0.0) GO TO 160
289 DIVTR(J) = AMIN (DELTR+DIVTR)
290 IF ((DIVTR(J) + SDIVT(J),GT,TR(J,J)) GO TO 140
291 SDIVT(J) = SDIVT(J) + DIVTR(J)
292 GO TO 160
293 C 140 DIVTR(J) = TR(J,J) - SDIVT(J)
295 SDIVT(J) = TR(J,J)
296 C DIVERT J TO I TRAFFIC THRU II
297 C
298 C 160 CONTINUE
299 CALL LINTRF(J,J+1)
301 DELTR = A-TRR(J,J)
302 C IF (DFLTR.LE.0.0) GO TO 220
303 C
304 C DIVTR=DELTR
305 CALL LINTRF(J,J+1)
306 DFLTR=A-TRR(J,J)
307 IF (DFLTR.LE.0.0) GO TO 220
308 DIVTR(J) = AMIN (DELTR+DIVTR)
309 IF ((DIVTR(J) + SDIVT(J),GT,TR(J,J)) GO TO 140
310 SDIVT(J) = SDIVT(J) + DIVTR(J)
311 GO TO 220
312 C 180 DIVTR(J) = TR(J,J) - SDIVT(J)
314 SDIVT(J) = TR(J,J)
315 C
316 200 CONTINUE
317 IF ((SDIVT(J),EQ,TR(J,J)) AND (SDIVT(J),EQ,TR(J,J)) GO TO 340
318 220 CONTINUE
319 IFLOP=0
320 GO TO 360
321 340 IFLOP=1
322 360 CONTINUE
323 C C CREATE A NEW TRAFFIC MATRIX WHICH ELIMINATES THE TRAFFIC BETWEEN C NOTES I AND J AND A TEMPORARY NETWORK NETCNF FOR THE PURPOSE C OF COST EVALUATION
324 C
325 DO 191 K1=1,1NR
326 DO 191 K2=1,1NR
327 TRR(K1,K2) = TR(K1,K2)
328 DO 191 K3=1,1NR
329 NETCNF(K1,K2,K3) = ORINT(K1,K2,K3)
330 191 CONTINUE
331 DO 190 K1=1,1NR
332 NETCNF(J,J,K1) = 0
333 C
334 NETCNF(J,J,K1) = 0
335 C
336 NETCNF(J,J,K1) = 0
337 C
338 NETCNF(J,J,K1) = 0
339 190 CONTINUE
340 TRR(J,J) = 0
341 TR(J,J) = 0
DO 380 IK=1, NR
   IF (IC.EQ. IK * OR. J.EQ. IK) GO TO 380
   TRR(IK+1) = TR(IK+1) + DIVTR(IK)
   TRR(IK+2) = TR(IK+2) + DIVTR(IK)
   TRR(IK+3) = TR(IK+3) + DIVTR(IK)
   ATRMAX = MAX(1,TRR(IK+1),TRR(IK+2),TRR(IK+3))
   INRNSC(IK) = INRNSC(IK) + 1
   CALL LINNUM(ATRMAX,OR, LINFUP, RH)
   RHOF2(IK) = RH
   RHOF2(IK+1) = RH
   DO 430 J=1, NR
      NETCNF(IK,J) = LINETO(N)
      NETCNF(IK,J+1) = LINETO(N)
      NETCNF(IK,J+2) = LINETO(N)
      NETCNF(IK,J+3) = LINETO(N)
   CONTINUE
   380 CONTINUE
   RETURN
   SUBROUTINE LINTRF(I,J)
   **************
   C
   C CONVERT LINES INTO TRAFFIC CAPACITIES BETWEEN NODES I AND J.
   C
   **************
   A20
   DO 100 IR=1, NR
      A=1+ ORINET(I,J,IR)*LINCAP(IR)+UTILI(IR)
   CONTINUE
   RETURN
   SUBROUTINE NETWC(SUMCST)
   **************
   C
   C FIND TOTAL INTERREGIONAL NETWORK COST, SUMCST, BASED ON SPECIFIC
   C CONFIGURATION NETCNF
   C
   **************
   INTEGER SUMCST
   SUMCST = 0
   DO 420 IR=1, NR
   IR=IR+1
   DO 440 IK=IR, NR
      IC=IRFST(NETCNF, IR, IK)
      IF (IC.EQ.0) GO TO 440
      I2=INRNSC(IK)
      J3=INRNSC(IK)
      150 CONTINUE
      CALL TCOST(J,LNEQU, J, LNKCHW, LNKCLN)
   DO 501 J3=1, NR
   J3=J3+2
   DO 502 J3=1, NR
   A-38
SUMCST = SUMCST + LNKCHW(J1, J3, J2)
CONTINUE
SUMCST = SUMCST + LNKCHI(J1, J2)
CONTINUE
SUMCST = SUMCST + LNKCI(N, J1, J2)
CONTINUE
CONTINUE
RETURN
SIROUTINF MINAD(IAAD, MINCST)
***************
C
C CAPACITY INCREASE IS REQUIRED WHEN IFLOP=0, AND THE CAPACITY AT
C MAXIMUM COST SAVINGS.
C
C
DIMENSION LINDI(NP3), LINDJ(NP3)
MINCST = 0
RTFJ = TR(I, J) - SDIVT; CREATING TRAFFIC FROM I TO J
RTFJ = TR(J, I) - SDIVT; CREATING TRAFFIC FROM J TO I
DO 500 II = 1, NR
IF (I0.EQ.1 OR I0.EQ.J) GO TO 500
IF (TRR(I, J) .OR. TRR(J, I) .OR. TRR(I, J) .OR. TRR(J, I)) GO TO 500
RETURN
END
C
C DETERMINE DELTA COST FOR INCREASED CAPACITY IN ALTERNATE ROUTES
C
C
AII = TRR(I, J) + RTFJ
AJI = TRR(J, I) + RTFJ
AM = AMAX1(AII, AJI)
IRD1 = NRSC(I)
IRD2 = NRSC(J)
CALL LINNUM(A0, A, LINDJ, LINDJ, 0, IRO)
DO 311 NN = 1, NR
LINDI(IN) = LINEOA(NN) - NETCF(I, J, NN)
LINDJ(JN) = LINEOA(NN)
NFNETCF(I, J, NN) = LINEOA(NN)
CONTINUE
C
C
LINK(I, J)
C
RII = TRR(I, J) + RTFJ
BJI = TRR(J, I) + RTFJ
AMJ = AMAX1(RII, BJI)
IRD1 = NRSC(J)
CALL LINNUM(A0, A, LINDJ, LINDJ, 0, IRO)
DO 311 NN = 1, NR
LINDJ(JN) = LINEOB(NN) - NETCF(J, I, NN)
LINDI(IN) = LINEOB(NN)
NFNETCF(J, I, NN) = LINEOB(NN)
CONTINUE
C
C
LINK(I, J)
C
C
CALL NETWKC(SUMCST)
IF (SUMCST.SGT.MINCST) GO TO 120
DO 207 NN = 1, NR
LINADI(NN) = LINDI(NN)
LINAJ(NN) = LINDJ(NN)
CONTINUE
C
C
110 AD = 1
MINCST = SUMCST

CONTINUE

C RESET TO INITIAL NETWORK CONFIGURATION FOR NEXT TRY

DO 250 NN= 1, N3

NETCHF(I+1+NN) = NETCHF(I+1+NN) + LINM(I+NN)

NETCHF(J+1+NN) = NETCHF(J+1+NN) + LIMD(J+NN)

NETCHF(J+1+NN) = NETCHF(J+1+NN) - LINM(J+NN)

CONTINUE

DO 460 TRP(I+IAD) = TRP(I+IAD) + RTPFI

TRR(I+IAD,J) = TRR(I+IAD,J) + RTPFJ

TRR(I+IAD,J) = TRR(I+IAD,J) - RTPFJ

RETURN

SUBROUTINE NETUP(IFFLIP*IAD,I,J)

***************

C UPDATE THE INTERREGIONAL NETWORK WHEN THERE IS SOME SAVINGS

***************

IF (IFFLIP = 1) GO TO 700

UPDATE THE OPTIMAL INTERREGIONAL NETWORK

DO 99 NN= 1, N3

ORINET(I+IAD+NN) = ORINET(I+IAD+NN) + LINM(I+NN)

ORINET(I+IAD+NN) = ORINET(I+IAD+NN) + LIMD(I+NN)

ORINET(J+IAD+NN) = ORINET(J+IAD+NN) + JLINM(J+NN)

ORINET(J+IAD+NN) = ORINET(J+IAD+NN) - JLIMD(J+NN)

CONTINUE

DO 701 NN= 1, N3

ORINET(I+J+NN) = 0

ORINET(J+I+NN) = 0

CONTINUE

RESET TRAFFIC MATRIX TR(NR*NR)

DO 980 IR= 1, NR

DO 990 IK= 1, NR

TR(IR, IK) = TRR(IR, IK)

CONTINUE

CONTINUE

UPDATE TOTAL COST FOR OVERALL NETWORK

ORICST = ORICST - MAXSAV

UPDATE NLINK MATRIX

NLINK(I) = NLINK(I) - 1

NLINK(J) = NLINK(J) - 1

RETURN

END

APRT STACOM. ICO$TJ/7777
SUBROUTINE ICOSTJ(INF11,J1,LNKCH1,LNKCLM)

***************

C CALCULATE INSTALLATION ANNUAL RECURRING COSTS NEEDED FOR
C COMMUNICATION LINK BETWEEN NODES I AND J. LNKCH1 = OTHERS
C LNKCLM = LINES: I AND J ARE GLOBAL TUPLES FOR SYSTEM TERMINATIONS
C UNDER CONSIDERATION. LINES = LINE CONFIGURATION BETWEEN T AND J
C
C ***************

PARAMETER NP1=130+NP2=1+NP3=4+NP4=3+NP5=360
PARAMETER NP7=4
DIMENSION LINFQ(INP3), LNKC(LNP3,NP4+2), LNKCLM(NP5+2)
COMMON/LINCH/LINMAX(INP3)+LINCAP(INP3)+ITL,IZ(NP3)
/COST/NL,N2,N44,N7,N6
/COST/AINSTC(NP2,NP3,NP4+3,2,2),RFPCP(NP2,NP3,NP4+3,2,2,2)
/AMSTLN(NP2,NP3,NP4+3,2)+RECRLN(NP2,NP4+3,2,16)+IMPLXY(NP3)
/INF/AINSTC(NP2,NP3),IRAND(INP3),IRAMP(INP3),IRAFL(NP2,NP3)
/ADV/AINSTC(INP1),KCHG,KAND* TERMINALS WITH SAME W =
/REF/REF(INP3),TRAFD(INP2+2,NP7),DISTCCE(NP6),MAPOD(N1)

C INITIALIZATION

II=MAPA(I)
IJE=MAPA(J)
IANDTN=IAND(IJ)
DO 100 NL=1+NL
DO 100 NM=1+NM
LNKCLM(NL,NM)=0
DO 100 NK=1+NN
LNKCH(NL,NK,NM)=0
CONTINUE

30 KRATEI=IRAND(I)+100 KRATF=IRAND(J)+100
KRENST=IRAND(I)+2 KRENSTJ=IRAND(J)+2
KK=KRENST+KRENSTJ
K=KRENST+KRENSTJ
K=KK
DST=DIST(I,J)
DISTNRFWEN(NODES I AND J)
INSTDST
ITIP=1 GPRIME COST FOR H/W UNIT
ITIP=2,3,4 GPRIME COST FOR ADDITIONAL UNIT
KP=IRATF(KRATEI,KRATEJ) A ACTUAL RATE STRUCTURE TO BE USED
DO 10 IL=1+IN3
IMPX=IMPLXY(IL) A DUPLEXING MODE I=H AND 2=I
40 NOV=LINES(IL) A NUMBER OF LINES REQUIRED
NOV=NOV+IANDTN+KAND
C
C CALCULATE COSTS FOR NON-LINE TYPE CHARGES
IF (NOV.EQ.0) GO TO 1 A NO LINS ARE REQUIRED
DO 2 IV=1+IV NO HIGH DENSITY RATE
C
C INSTALLATION COSTS FOR NON-LINE TYPE CHARGES
C
LNKCH(NL,IV+1)=AINSTC(KP,IL,IV,KK,IMPX,ITIP)*KCHG*NOV

A-41
1 + ANSTLC(1RIL,IV,KK,1DPX,2)*AN*NDV

C ANNUAL RECURRING COSTS FOR NON-LINE TYPE CHARGES

C LNKCLN(IL,IV)= (RECRLN(1RIL,IV,KK,1DPX,ITID)*KCM8*NDV
2 + RECRLN(1RIL,IV,KK,1DPX,2)*AN*NDV)*12.

C CONTINUE

C CALCULATE LINE COSTS

C LIN= IFLAG(KR,IL) = LINEAR IF 1 AND NONLINEAR OTHERWISE

C ANNUAL LINE INSTALLATION COST

C AN=1,

C LNKCLN(IL,1)= ANSTLC(1RIL,KK,1DPX,2)*AN*NDV

C IF (LIN.EQ.1) GO TO 41

C LINEAR LINE RECURRING COST FUNCTION

C BNL=ST/RECRLN(KR,IL,KK,1DPX,1)

C LNKCLN(IL,2)= RECRLN(1RIL,IL,KK,1DPX,2)*AN*NDV*12.

C GO TO 32

C CONTINUE

C NONLINEAR LINE RECURRING FUNCTION

C DO 10 NON1=1,8

C NON2=2*NON

C NON1=NON2=1

C COST=RECRLN(KR,IL,1,1DPX,NON2)

C DT=RECRLN(KP,KL,1DPX,1)

C IF (NST+DT) GO TO 51

C LNKCLN(IL,2)= COST*NST*NDV*12+LNKCLN(IL,2)

C GO TO 32

C CONTINUE

C 51 CONTINUE

C LNKCLN(IL,2)= COST*NST*NDV*12+LNKCLN(IL,2)

C DST=NST+DT

C 10 CONTINUE

C 32 CONTINUE

C 47 1 CONTINUE

C KCM8=1

C KAPD=1

C RETURN

END
**SUBROUTINE RHOFUN(T1,T2,LINEQ,LNLMT,RHOLIN,RH0)**

**CALCULATE LINE UTILIZATION**

**T1**: LINT TO SWITCHER TRAFFIC  
**T2**: SWITCHER TO LINT TRAFFIC

**LNLMT**: HIGHEST LINE TYPE  
**LINEQ**: LINE CONFIGURATION

**PARAMETER NP3=4**

**COMMON/LINCHR/LINFIX(NP3),LINCAP(NP3),UTILIZ(NP3)**

**/CONST/ N1,N2,N3,N4,N7,N7NCITY**

**/SUM/ASUM(4),RSUM**

**/XMT/TIMXMT(7,NP3),WAIT(6)**

**/MSLA/AMSL(7)**

**DIMENSION LINEQ(1),RHOLIN(1)**

**RHO=0.**

**CAP=0.**

**DO 8 N=1,N3**

**CAP=CAP+LINEQ(N)*LINCAP(N)**

**CONTINUE**

**CN=LINCAP(LNLMT)/CAP **

**NORMALIZATION FACTOR**

**KSAC1=CN*T2*ASUM(3)/(RSUM*AMSL(5)*R**

**OUTPUT WITH PRIO 1**

**XSAC2=0.**

**IF(AMSL(6)-EN.0.) GOTO 101**

**IF(AMSL(6)-EN.0.) GOTO 101**

**CONTINUE**

**XSAC3=CN*T1/(AMSL(4)*R**

**INPUT TRAFFIC IN TRANS**

**RHOLIN(1)=XSAC1+TIMXMT(5,LNLMT)**

**RHOLIN(2)=XSAC2+TIMXMT(6,LNLMT)**

**RHOLIN(3)=XSAC3+TIMXMT(4,LNLMT)**

**RHO=RHOLIN(1)+RHOLIN(2)+RHOLIN(3)**

**RETURN**

**END**

**QPRT STACOM.LINNUM/0777**
SUBROUTINE LINNUM(T1,T2,LINQ,TNLMT,FLAG,RHO)

C FIND LINE CONFIGURATION BASED ON THE GIVEN TRAFFIC AND
C APPLICABLE LINE TYPE
C
C JFLAG= 1 FOR MULTIDROP LINE CASE
C T1= LINE TO SWITCHER TRAFFIC
C T2= SWITCHER TO LINE TRAFFIC
C
C
PARAMETER NPS=4
COMMON/LINCHR/ LINMIX(NPS),LINCAP(NPS),UTILIZ(NPS)
/CONST/ N1,N2,N3,N4,N7,N10,CITY
/MSLA/ AMSL(7)
INTEGER TRAF
DIMENSION LINEQU(1),LINMT(3)
TRAF=T1+T2
DO 1 I=1,N3
LINEQU(I)=0
1 CONTINUE
LINMT=0
CALL REFER
LINMT=LINMT+1
CALL REFER
LINMT=LINMT+1
C SET UP INITIAL LINE CONFIGURATION
C
C IF(JFLAG .EQ. 1) GOTO 10
C CONTINUE
C LINEQU(LINMT)=LINEQU(LINMT)+1
C LCAP=LINCAP(LINMT)+UTILIZ(LINMT)
C IF(TRAF .LT. LCAP) GOTO 7
C TRAF=TRAF-LCAP
C CALL REFER
C GOTO 3
C CONTINUE
C LINEQU(LINMT)=TRAF/(LINCAP(LINMT)+UTILIZ(LINMT))+1
C CONTINUE
C CALL RHOFLUNIT(T1,T2,LINQ,TNLMT,RHOLIN,RHO)
C IF(RHO .LT. UTILIZ(LINMT)) GOTO 150
C IF(LINMT .NE. N3) GOTO 72
C IF(JFLAG .NE. 1) GOTO 73
C LINEQU(N3)=LINEQU(N3)+1 0 NEEDED TO BE MODIFIED
C GOTO 70
C CONTINUE
C DO 2 N=1,N3
C IF(LINEQU(N) .NE. 0) GOTO 20
C CONTINUE
C CONTINUE
C CONTINUE
C CONTINUE
C CONTINUE
C CONTINUE
C CONTINUE
C CONTINUE
C CONTINUE
C CONTINUE
C IF(NL .EQ. N3) GOTO 74
C LINEQU(NL)=0
C CONTINUE
C NL=NL+1
C IF(LINMIX(NL) .EQ. 0) GOTO 22
C LINEQU(NL)=LINEQU(NL)+1
C
57 74  GOTO 70
58 74  CONTINUE
59  LINEGU(1)=1
60  GOTO 70
61 72  CONTINUE
62  LINEGU(LNLMT(1))=0
63  LNLMT=NLMT+1
64  IF(LINMIX(LNLMT),FG,0) GOTO 70
65  LINEGU(LNLMT(1))=1
66  GOTO 70
67 150  CONTINUE
68  LNLMT=NLMT
69  RETURN
70  SUBROUTINE REFER
71  C
72  C  FIND THE UPPER LIMIT OF LINE TYPE ALLOWED
73  C
74  DO 14  NN=1,N3
75  LTRAF=TRAF/UTILIZ(NN)+0.5
76  IF(LINMIX(NN)+FG,0) GOTO 14
77  LNLMT=NN
78  IF(LINCAP(NN)+GT, LTRAF) GOTO 15
79 14  CONTINUE
80 15  CONTINUE
81  RETURN
82  END

OPRT STACON.PACK/0777
51928*STACOM(1).PACK/0777
1 COMPILER (FLD=ABS)
2 SUBROUTINE PACK(I,K,L,IA)
3 C
4 C
5 C RETRIEVE/STORE DATA FROM/INTO ARRAY IA
6 C L=1 FOR STORING AND L=2 FOR RETRIEVAL
7 C K= DISTANCE DATA CONCERNED
8 C
9 C
10 DIMENSION IA(1)
11 IQ=(I-1)/4 THE WORD LOCATION
12 IR=I-IQ*4 THE QUARTER CONCERNED
13 IQ=IQ+1
14 IS=(IR-1)*4
15 IF(L.EQ.1) GOTO 10
16 C
17 C RETRIEVE 19 BITS BEGINNING AT IS-TH BIT OF THE IQ-TH WORD
18 C
19 K=FLD(15+9,IA(IQ))
20 RETURN
21 C
22 C STORE IT19 BITS BEGINNING AT IS-TH BIT OF THE IQ-TH WORD
23 C
24 CONTINUE
25 FLD(15+9,IA(IQ))=K
26 RETURN
27 END

END STACOM.DIST/0777
FUNCTION DIST(I,J)

C ***************
C FIND DISTANCE BETWEEN I AND J
C
C ***************
PARAMETER NP1=130,NPC=360
PARAMETER NP6=(NPC*NPC/2-NPC*1)/4+1
PARAMETER NP7=4
COMMON /REF/REF(NPC),TRAFF(NP1+2,NP7),
* DSTMCE(NP6),HAPADR(NP1)
INTEGER DSTMCE
DST=0.
IF(I.EQ.J) RETURN
II=MAPADR(I) ACTUAL CITY INDEX
JJ=MAPADR(J)
IF(J.EQ.JJ) RETURN
I,J=LINK(I,J,J)
CALL PACK(I,JL,IDIST+2,DSTMCE)
DIST=IDIST
IF(DIST .GT. 511) RETURN
DST=RCVR(I,JL)
RETURN
END

519200$TACOM0).DIST/0777

A-47
FUNCTION LINK(J,K)

C ***************

C FIND THE RELATIVE LOCATION FOR (J,K) COMBINATION

C WHICH IS THEN USED FOR FINDING DISTANCE BETWEEN SYSTEM

C TERMINATIONS J AND K

C ***************

PARAMETER NP1=130*NPC+360
PARAMETER NP6=(NPC+NP2-1)*4+1
PARAMETER NP7=4*NP3=4
COMMON /CONST/ N1,N2,N3,N4,N7,NCITY
1 /REF/IREF(NP1)+TRAF(NP1)+NPCF(NP1)+MAP(NP1)
INTEGER DSTNC
LINK=0
IF(J.GT.NCITY.OR.K.GT.NCITY.OR.K.FO.J) RETURN
JJ=J
K=K
IF(J.LT.K) GOTO 1
J=K
K=J
1 CONTINUE
CONTINUE
LINK=1+NCITY + K - IREF(JJ)
RETURN
END

A-48
5128*STACOM(I),RECOVR/O777
1     FUNCTION RECOVR(I)
2     *****************************************
3     C
4     C RETRIEVE OVERFLOW DISTANCE DATA FROM IVRD
5     C
6     *****************************************
7     PARAMETER NPC=360,NPO=10*NPC
8     COMMON /OVER/ IVRD(NPO+2),IOVR1
9     DO 10 I=1,IOVR1
10    IF(I.EQ. IVRD(I)) GOTO 20
11    CONTINUE
12    WRITE(6,99) I
13    99 FORMAT('IX+1 NO OVERFLOW DATA HAS BEEN FOUND ','
14     * FOR LOCAL INDEX',216)
15    STOP
16    20 CONTINUE
17    RECOVR=IVRD(NP+2)
18    RETURN
19    END

&PRV L:PLOTPT/O777
SUBROUTINE PLOTPT(L1, L3)

C SUBROUTINE FOR MOVING CALCOMP PEN WITH OR WITHOUT PEN DOWN
C L3=1 FOR MOVING WITH PEN DOWN
C =2 FOR MOVING WITH PEN UP
C =3 FOR CLOSING THE PLOTTING

PARAMETER NPC=360
COMMON/VH/VERT(NPC), HORIZ(NPC)
DIMENSION IBUF(1000)
DATA IF/0/ GFLAG FOR PLOTS CALL
DATA X/1.2566/
IF(IP .NE. 0) GOTO 50
CALL PLOTS
50 CONTINUE
IF(L3 .EQ. 3) GOTO 100 GPPLOTTING IS TO BE CLO~ED
AV=VERT(L1)
AH=HORIZ(L1)
BV=AV*COS(X)+AH*SIN(X)
BH=AV*SIN(X)-AH*COS(X)
BV=(BV-5500)*301.
IF(L .EQ. 2) GOTO 80
CALL SYMBOL(BH,BV,.025,.025,.025).
80 CONTINUE
CALL PLOT(RH,BV,.3) OPEN IS UP
IP=1 GPPLOTTING CALL IS NOT NEEDED ANY MORE
RETURN
100 CONTINUE
CALL PLOT(10,.0,.999)
RETURN
END

GPR L.RSPNSE/0777
SUBROUTINE RSPNSE(T1, T2, LINTYP, N, ICK)

CALCULATE MEAN RESPONSE TIME FOR THE PROPOSED MULTIDROP LINE

WAIT(6)=TFMIZED DELAYS DUE TO

1=WAIT FOR POLLING  2=WAIT FOR I/O  3=INPUT XMT TIME
4=CPU TURNAROUND  5=OUTPUT QUEUE IF WAIT 6=OUTPUT XMT TIME

PARAMETER NPS=4
COMMON/RSP/RHOLIN(4),RSPTIM
1 /XMT/TIMXMT(T,NPS), WAIT(6)
2 /BOUND/ TERMS,TIMED,O,PROC,PLOT
3 /CONST/N1,N2,N3,N4,N7,NFTY
DIMENSION LDUMMY(NP3)
DO 10 N=1,N3
LDUMMY(N)=0
CONTINUE
LDUMMY(LINTYP)=1
IK=0
CALL RHOFUN(T1,T2,LDUMMY,LINTYP,RHOLIN,PHN)
RHOLIN(4)=1.-RHO
IF(RHOLIN(4) .LE. 0.) RETURN
WAIT(1)=TIMXMT(1,LINTYP)+TIMXMT(2,LINTYP)/(N-1/2)
WAIT(2)=(1.-RHOLIN(4))*TIMXMT(1,LINTYP)/(1.-RHOLIN(1)*)
WAIT(3)=TIMXMT(3,LINTYP)
WAIT(5)=(1.-RHOLIN(4))*TIMXMT(5,LINTYP)/(1.-RHOLIN(1)*)
WAIT(6)=TIMXMT(5,LINTYP)
RSPTIM=0.
DO 11 J=1,6
RSPTIM=WAIT(J)+RSPTIM
CONTINUE
IF(RSPTIM .GT. TIMREQ) RETURN
IK=1
RETURN
END
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT&amp;T</td>
<td>American Telephone and Telegraph Company</td>
</tr>
<tr>
<td>@RUN, etc.</td>
<td>Control statements under EXEC 8 system (of the UNIVAC computer system)</td>
</tr>
<tr>
<td>BPS</td>
<td>Bits per second</td>
</tr>
<tr>
<td>CalComp</td>
<td>CALifornia COMputer Products</td>
</tr>
<tr>
<td>Central Link</td>
<td>The direct link between a computer and a remote terminal</td>
</tr>
<tr>
<td>Centroid</td>
<td>The geographical center of a set of system terminations</td>
</tr>
<tr>
<td>Communication Network</td>
<td>A network with several terminals connected by a set of communication channels</td>
</tr>
<tr>
<td>Communication Protocol</td>
<td>The system used for performing interfacing (hand-shaking) between a computer and a remote terminal</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>Data Base</td>
<td>A collection of cross-referenced set of files which allows systematic data filing and retrieval by a digital computer</td>
</tr>
<tr>
<td>D Bank</td>
<td>Storage area for data under EXEC-8 system of the UNIVAC Computer System.</td>
</tr>
<tr>
<td>Drop</td>
<td>A chargeable item associated with each terminal on a multidrop line</td>
</tr>
<tr>
<td>EXEC-8</td>
<td>UNIVAC 1100 series executive system</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>FORMula TRANslator</td>
</tr>
<tr>
<td>FORTRAN V</td>
<td>A FORTRAN type of high level language which is only applicable in UNIVAC computers</td>
</tr>
<tr>
<td>I Bank</td>
<td>Storage area for program instructions under EXEC 8 system of the UNIVAC Computer System</td>
</tr>
<tr>
<td>ID</td>
<td>IDentification</td>
</tr>
<tr>
<td>Line Utilization</td>
<td>The ratio of traffic on a line to the line capacity</td>
</tr>
</tbody>
</table>

B-1
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPL</td>
<td>Multischedule Private Line, one of the interstate tariffs used by AT&amp;T</td>
</tr>
<tr>
<td>Multidrop Line</td>
<td>A communication line which has more than one terminal and is connected to a data processing system</td>
</tr>
<tr>
<td>Multidrop Network</td>
<td>A communication network where one or more lines are multidrop lines</td>
</tr>
<tr>
<td>PUNCH$, etc.</td>
<td>System designated file name for punch card output, etc.</td>
</tr>
<tr>
<td>Regional Network</td>
<td>A network which connects all terminals in a given region</td>
</tr>
<tr>
<td>Regional Switching Center (RSC)</td>
<td>A regional data processing center which is used to provide the message switching capability for all terminals in the region</td>
</tr>
<tr>
<td>STACOM</td>
<td>STAte Criminal Justice COMmunication Project</td>
</tr>
<tr>
<td>Star Network</td>
<td>A communication network where each system termination is directly connected to the central data processing system</td>
</tr>
<tr>
<td>SUP-Time</td>
<td>A run time estimate by the EXEC-8 accounting subsystem which accounts for the amount of time spent by a run on usage of CPU, I/O processing and execution of system control statements and executive requests</td>
</tr>
<tr>
<td>System Termination</td>
<td>A logical node in the communication system under the STACOM program, which consists of one or more physical terminals</td>
</tr>
<tr>
<td>TELPAK</td>
<td>A specific tariff for a telecommunication network</td>
</tr>
<tr>
<td>Terminal</td>
<td>A device that allows users of a data processing system to gain access to that system in a more convenient manner than the input/ output devices local to that system</td>
</tr>
<tr>
<td>Terminal Response Time</td>
<td>The duration from the time a user initiates a request for network service at the terminal to the time he receives a complete response</td>
</tr>
<tr>
<td>Tree</td>
<td>A graph which has a root node without any predecessors and other nodes have unique predecessors</td>
</tr>
</tbody>
</table>
UNIVAC

UNIVersal Automatic Computer, a computer trade name by Sperry Rand Corporation

Vertical Horizontal (V-H) Coordinates

A pair of numbers which are designated by AT&T for cities and used for the purpose of calculating distance between any two cities.