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:

- **State Criminal Justice Telecommunications (STACOM) Final Report - Volume I:**
  - Executive Summary

- **State Criminal Justice Telecommunications (STACOM) Final Report - Volume II:**
  - Requirements Analysis and Design of Ohio Criminal Justice Telecommunications Network

- **State Criminal Justice Telecommunications (STACOM) Final Report - Volume III:**
  - Requirements Analysis and Design of Texas Criminal Justice Telecommunications Network

- **State Criminal Justice Telecommunications (STACOM) Final Report - Volume IV:**
  - Network Design Software Users' Guide

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>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1</td>
<td>Data Requirements</td>
</tr>
<tr>
<td>2.4</td>
<td>PROCESSING LOGIC</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Traffic Calculation</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Distance Calculation</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Formation of Regions</td>
</tr>
<tr>
<td>2.4.4</td>
<td>Selection of Regional Switchers</td>
</tr>
<tr>
<td>2.4.5</td>
<td>Formation of Regional Star Networks</td>
</tr>
<tr>
<td>2.4.6</td>
<td>Optimization of Regional Networks</td>
</tr>
<tr>
<td>2.4.7</td>
<td>Formation of an Interregional Network</td>
</tr>
<tr>
<td>2.4.8</td>
<td>Optimization of an Interregional Network</td>
</tr>
<tr>
<td>2.5</td>
<td>OUTPUT</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Printer</td>
</tr>
<tr>
<td>2.5.2</td>
<td>CalComp Plot</td>
</tr>
<tr>
<td>2.6</td>
<td>SYSTEM CONFIGURATION</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Hardware</td>
</tr>
<tr>
<td>3</td>
<td>PROGRAM OPERATIONS</td>
</tr>
<tr>
<td>3.1</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>3.2</td>
<td>ENVIRONMENT</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Hardware</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Software</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Functional Limitations</td>
</tr>
<tr>
<td>3.3</td>
<td>RUN DESCRIPTION</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Initialization and Setup</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Run Options</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Control Instruction and Sequences</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Program Listing</td>
</tr>
<tr>
<td>3.4</td>
<td>SAMPLE RUN</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Run Stream</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Input</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Output</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
</tr>
<tr>
<td>APPENDIXES</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>STACOM PROGRAM LISTING</td>
</tr>
<tr>
<td>B</td>
<td>GLOSSARY</td>
</tr>
</tbody>
</table>

**Figures**

2-1 Example of a Digital Communication Network --- 2-2
2-2 Basic Communication Network Configurations ------- 2-3
2-3 Example of Initial Regional Networks and an Initial Interregional Network 2-6
2-4 Example of Optimized Regional Networks and an Optimized Interregional Network 2-7
2-5 A Tree with A as its Root ------------------------ 2-8
2-6 Internal Representation of the Tree in Figure 2-5 ---- 2-9
2-7 STACOM Program Structure ------------------------ 2-10
2-8 Flow Chart for Formation of Regions -------------- 2-22
2-9 Example of Region Network Formation and Regional Switcher Selection 2-25
2-10 Flow Chart for RSC Selection ------------------ 2-26
2-11 Flow Chart for Regional Star Network Formation --------- 2-28
2-12 Typical Star Network ------------------------- 2-30
2-13 Typical Multidrop Network of Optimization --------- 2-30
2-14 Flow Chart for Subroutine ESSWIL ----------------- 2-32
2-15 Relationship among K, L, KI, and M Parameters --------- 2-33
2-16 Flow Chart for Subroutine TRYLNK ----------------- 2-35
77-53, Vol. IV

2-17  Flow Chart for Subroutine RSPNSE  ------------------------------------------  2-36
2-18  Flow Chart for Intraregional Line Selection  ---------------------------  2-38
2-19  Basic Topology of Line Elimination  ------------------------------------------  2-39
2-20  Flow Chart for Interregional Network Optimization  -----------------  2-40
3-1   CalComp Plot from the Example Run  ------------------------------------------  3-34

Tables

2-1   Examples of Line Configurations obtained by Subroutine LINNUM  ------------------------------------------  2-29
3-1   Formats for Input Data  ------------------------------------------  3-5
3-2   Input Data for the Example Run  ------------------------------------------  3-13
3-3   Printer Output from the Example Run  ------------------------------------------  3-25
3-4   Unit 6 Printer Output from the Example Run  ------------------------------------------  3-33
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:
(1) 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$,99PLTF will direct the punch card images to a CalComp plotter designated with 99PLTF.

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 @@@ 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.
SECTION 2

THE STACOM PROGRAM

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.

![Network Configurations Diagram](image)

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 database 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:
The line utilization factor does not exceed a specific number.

The average terminal-response time is less than a preselected unit of time.

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).

![Diagram](image)

**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:

\[
\begin{align*}
\text{IA} & \quad f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \\
\end{align*}
\]

Let \( c(f_1) \) = content of \( i \)-th field in a storage cell \( \text{IA} \), where \( \text{IA} \) is an internal index for a system termination \( A \) (data), then

- \( c(f_1) \) = the number of system terminations under \( A \)
- \( c(f_2) \) = a pointer which points to the first successor of \( A \)
- \( c(f_3) \) = a pointer which points to the next system termination whose predecessor is the same as \( A \)'s
- \( c(f_4) \) = a pointer which points back to the previous system termination whose predecessor is the same as \( A \)'s
- \( c(f_5) \) = a pointer which points to \( A \)'s predecessor

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 \( \text{IA}, \text{IB}, \text{IC}, \text{IP}, \) and \( \text{IP} \) which are internal cardinal numbers for system terminations \( A \), \( B \), \( C \), \( D \), and \( E \).

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

![Figure 2-5. A Tree with A as its Root](2-8)
Figure 2-6. Internal Representation of the Tree in Figure 2-5

In the case of IC, ID is the next successor of IA and the previous successor of IA is IC. 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.
Figure 2-7. STACOM Program Structure

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 terminals, 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.

2-14
The @SYM,P command directs the resulting plot card images to a CalComp 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.
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.

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.

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.

Traffic Calculation

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.

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

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

and

\[
\text{TRAFDN}(i) = \sum_{j=1}^{N7} \text{TRAFD}(i, 1, j)
\]
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 DSTNCE or IVRD.

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 DSTNCE 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 DSTNCE, 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
   \[ L1 = \left[ \frac{(L-1)}{4} \right] + 1 \]
   \[ S1 = (L-1) \mod 4 + 1 \]
where \( \lfloor x \rfloor \) = the integer part of \( x \) and
\[
D1 = D \text{ if } D < 511 \\
511 \text{ if } D \geq 511
\]
(5) Store \( D1 \) in segment \( S1 \) of entry \( L1 \) of table \( DSTNCE \).
(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 \( L, L1 \) and \( S1 \) as described above.
2. Retrieve the content \( D1 \) in segment \( S1 \) of entry \( L1 \) of table \( DSTNCE \). If \( D1 < 511 \), it is the distance.
3. If \( D1 = 511 \), 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.,
\[
TPR = \frac{TPR1}{ANR1}
\]

with
\[
TPR1 = \sum_{i=1}^{N1} \left( TRAFIT(i) + TRAFDN(i) \right)
\]

where
\[
I = \text{the set of system terminations in preloaded regions which do not allow other system terminations to be inserted to them}
\]
\[
ANRI = NR1 - \text{[number of preloaded regions which do not allow any insertions]}
\]
START

ESTIMATE
1. TPR
2. TRPL

IS NRI = 17

ASSIGN ALL SYSTEM TERMINATIONS TO REGION 1

TERMINATE

NREG = 1

TRFS = 0

IS NREG A PRELOADED REGION?

Y

1. FIND NS1
2. ADD ITS TRAFFIC TO TRFS

IS TRFS > TRPL?

N

1. FIND NS2
2. ADD ITS TRAFFIC TO TRFS

IS TRFS > TRPL?

N

Y

RE-ESTIMATE
1. TPR
2. TRPL

N

INCREMENT NREG BY 1

ASSIGN REMAINING SYS. TERMINATIONS TO LAST REGION

TERMINATE

N

IS NREG = NRI?

Y

IS NS1 SELECTED?

Y

1. ADD ITS TRAFFIC TO TRFS
2. FIND NS1

IS TRFS > TRPL?

N

N

1. FARTHEST UNASSIGNED SYSTEM TERMINATION FROM SYSTEM CENTROID NSCCI
2. NEAREST UNASSIGNED SYSTEM TERMINATION FROM NS1
3. REGIONAL SWITCHING CENTER

Figure 2-8. Flow Chart for Formations of Regions

TPR = TRAFFIC PER REGION
TPRL = 0.8 TPR
TRFS = PARTIAL TRAFFIC SUM
NS1 = FARTHEST UNASSIGNED SYSTEM TERMINATION FROM SYSTEM CENTROID NSCCI
NS2 = NEAREST UNASSIGNED SYSTEM TERMINATION FROM NS1
RSC = REGIONAL SWITCHING CENTER
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:

\[ TPR1 = TPR1 - TPFS \]
\[ TPR = \frac{TPR1}{(ANR1 - 1.)} \]
\[ TPRL = 0.9 \times TPR \]

On the other hand, if TPFS 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

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

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

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

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 NMBR_i \times (TRAFFI_i + TRAFFO_i) \times DIST(i, K)
NMBR = NUMBER OF SYSTEM TERMINATIONS IN THE REGION
DIST(i, K) = DISTANCE BETWEEN SYSTEM TERMINATIONS i AND K

WCASE = 1 \times 10^{12}

IS RSC FOR THIS REGION PRESELECTED?

K = 1

INCREMENT K BY 1
CALCULATE SUMT

IS SUMT > WCASE?

WCASE + SUMT
NRSC = K

IS K = NMBR?

TERMINATE

Figure 2-10. Flow Chart for RSC Selection
After the above processing has been repeated \( \text{NMBR} \) times, the resulting \( \text{NRSC} \) is the index for the selected \( \text{RSC} \) and \( \text{WCASE} \) 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.

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.

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

The derivation of line configuration \( \text{LDUMMY} \) by subroutine \( \text{LINNUM} \) and the associated cost, \( \text{COST} \), deserves more explanation. The program assumes that the duplexing mode for all line types under consideration to be half-duplexed. Therefore, subroutine \( \text{LINNUM} \) will sum up \( \text{TRFIN} \) and \( \text{TRFOUT} \) 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 \( \text{LDUMMY} \) 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>
<thead>
<tr>
<th>Traffic (bps)</th>
<th>Line Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>300 bps</td>
<td>1200 bps</td>
</tr>
<tr>
<td>200</td>
<td>1</td>
</tr>
<tr>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>850</td>
<td>0</td>
</tr>
<tr>
<td>1300</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
</tr>
<tr>
<td>3500</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 Lj. If we express it as a formula, then

$$K = i \text{ such that } S_i = \max\{S_j\}$$

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

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 NTk,RMS which constrains the number of system terminations on a multidrop line. If this is true, it skips the process of calling on subroutine TRYLNK, because it is not possible to integrate both sub-networks without violating the said constraint. Otherwise, it continues to the distance test.

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

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 $KI$ 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 $KI$ 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 $K$, $L$, $KI$, and $M$. A detailed description of functions performed by subroutine TRYLNK is given in Paragraph 2.4.6.3. After all possible combinations for $K$, $L$, $KI$, 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 $K$, $KI$, $L$, 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.
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 switches, 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.

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 LINEQ between RSCs I and J is obtained by calling subroutine LINNUM.

Cost of line configuration LINEQ 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 Interregion 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 tri... 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 preloadings 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 drpc. 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.

2-42
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 MSCs, 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, NPL is a parameter variable which is used to make the number of system terminations allowed in a system a variable. A choice of NPL 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 @ASG,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$,G9PLTF 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 \([X]\) 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>
</tbody>
</table>
| 5.       | ID, name, city index, additional no. of terminals and traffic to/from each data base for each system termination. | INDEXPT(N1), NAMEST(N1), IADD(N1), MAPADR(N1), TRAFD(N1,2,N7) | a. N1 if N7≤2  
b. 2N1 if N7>2 | (A4,1X,3A6,A4, 12,4F10.2/6F10.2) |
<p>| 6.       | No. of rate structures                                                            | N2                                 | 1                      | (I3)    |
| 7.       | Rate application matrix                                                           | IRATEJ(N2,N2)                      | N2                    | (10I2)  |
| 8.       | Traffic density index and applicable rate structure for each city                | IRAND(NCITY,2)                     | [NCITY/40]            | (80I1)  |
| 9.       | No. of applicable line types                                                      | N3                                 | 1                      | (I3)    |</p>
<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>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>(4F9.2/I1/10F8.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b. 2xN2(2+N3x3x2)</td>
<td>(4F9.2/I1/10F8.3/3x2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>if non-linear</td>
<td>10F8.3</td>
</tr>
<tr>
<td>15.</td>
<td>Action indices for regions</td>
<td>NRGF, NCODE for IACTN (NR1,2)</td>
<td>Variable</td>
<td>(2I)</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), NPLOH (N3), NAKOH(N3), MOH (N3), TAMOH(N3), TAPD (N3)</td>
<td>N3</td>
<td>(5I4,2F7.5)</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>(A6,2I4,2F6.3)</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,IX,A4,A5)</td>
</tr>
<tr>
<td>21</td>
<td>System Center</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 "EXIT after interrupting the line communication by pressing the BREAK key will terminate a program execution at any time. On the other hand, the EXEC-$ 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 J6G3YL,51928,20,90/1000
@ASG,UP 100
@SYM,P PUNCH$,G9PLTF
@XOT 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 J6G3YL, identifies its account number as 51928, assigns project ID as 20, 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 p.in': 3d
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

<table>
<thead>
<tr>
<th>City</th>
<th>County</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>PALESTINE</td>
<td>ANDERSON</td>
<td>8550</td>
</tr>
<tr>
<td>ANDREWS</td>
<td>ANDREWS</td>
<td>8847</td>
</tr>
<tr>
<td>LUFKIN</td>
<td>ANGELINA</td>
<td>8575</td>
</tr>
<tr>
<td>ROCKPORT</td>
<td>ARANSA S</td>
<td>9405</td>
</tr>
<tr>
<td>ARCHER CITY</td>
<td>ARCHER</td>
<td>8396</td>
</tr>
<tr>
<td>JOHNDANTON</td>
<td>ATASCOSA</td>
<td>9332</td>
</tr>
<tr>
<td>BELVILLE</td>
<td>AUSTIN</td>
<td>8964</td>
</tr>
<tr>
<td>MULESHOE</td>
<td>BAILEY</td>
<td>8518</td>
</tr>
<tr>
<td>SEYMOUR</td>
<td>BAYLOR</td>
<td>8437</td>
</tr>
<tr>
<td>BEEVILLE</td>
<td>BFF</td>
<td>9378</td>
</tr>
<tr>
<td>BELTON</td>
<td>BELL</td>
<td>8027</td>
</tr>
<tr>
<td>PORT HOOD</td>
<td>BFLL</td>
<td>8832</td>
</tr>
<tr>
<td>MARKER HEIGHTS</td>
<td>BFLL</td>
<td>8832</td>
</tr>
<tr>
<td>KILLEEN</td>
<td>BELL</td>
<td>8032</td>
</tr>
<tr>
<td>NOLANVILLE</td>
<td>BELL</td>
<td>8032</td>
</tr>
<tr>
<td>TEMPLE</td>
<td>BELL</td>
<td>8032</td>
</tr>
<tr>
<td>ALAMO HEIGHTS</td>
<td>BEXAR</td>
<td>9225</td>
</tr>
<tr>
<td>FT SAN HOUSTON</td>
<td>BEXAR</td>
<td>9225</td>
</tr>
<tr>
<td>LEON VALLEY</td>
<td>BEXAR</td>
<td>9225</td>
</tr>
<tr>
<td>SAN ANTONIO</td>
<td>BEXAR</td>
<td>9225</td>
</tr>
<tr>
<td>UNIVERSAL CITY</td>
<td>BEXAR</td>
<td>9225</td>
</tr>
<tr>
<td>CLIFTON</td>
<td>BOSQUF</td>
<td>8640</td>
</tr>
<tr>
<td>MERIDIAN</td>
<td>BOSQUF</td>
<td>8640</td>
</tr>
<tr>
<td>TEXARKANA</td>
<td>ROWIE</td>
<td>8111</td>
</tr>
<tr>
<td>ALVIN</td>
<td>BRAZORIA</td>
<td>8996</td>
</tr>
<tr>
<td>ANGLETON</td>
<td>BRAZORIA</td>
<td>9099</td>
</tr>
<tr>
<td>CLUTE</td>
<td>BRAZORIA</td>
<td>9091</td>
</tr>
<tr>
<td>FREESTOWN</td>
<td>BRAZORIA</td>
<td>9093</td>
</tr>
<tr>
<td>LAKE JACKSON</td>
<td>BRAZORIA</td>
<td>9091</td>
</tr>
<tr>
<td>PEARLIND</td>
<td>BRAZORIA</td>
<td>9091</td>
</tr>
<tr>
<td>BRYAN</td>
<td>BRAZOS</td>
<td>8927</td>
</tr>
<tr>
<td>COLLEGE STATION</td>
<td>BRAZOS</td>
<td>8927</td>
</tr>
<tr>
<td>ARLINNE</td>
<td>BREWER</td>
<td>9304</td>
</tr>
<tr>
<td>FAUJAN</td>
<td>BROOKS</td>
<td>9645</td>
</tr>
<tr>
<td>BROWNSWOOD</td>
<td>BROUN</td>
<td>8797</td>
</tr>
<tr>
<td>CALDERWELL</td>
<td>BURLESON</td>
<td>8840</td>
</tr>
<tr>
<td>PORT LAVACA</td>
<td>CALHOUN</td>
<td>9248</td>
</tr>
<tr>
<td>BROWNSVILLE</td>
<td>CAMERON</td>
<td>9861</td>
</tr>
<tr>
<td>HARLINGEN</td>
<td>CAMERON</td>
<td>9820</td>
</tr>
<tr>
<td>PORT ISABEL</td>
<td>CAMERON</td>
<td>9807</td>
</tr>
<tr>
<td>SAN BENITO</td>
<td>CAMERON</td>
<td>9826</td>
</tr>
<tr>
<td>LINDEN</td>
<td>CASS</td>
<td>8217</td>
</tr>
<tr>
<td>DIMMIT</td>
<td>CASTRO</td>
<td>8427</td>
</tr>
<tr>
<td>ANAHUAC</td>
<td>CHAMPAE</td>
<td>8884</td>
</tr>
<tr>
<td>JACKSONVILLE</td>
<td>CHEROKEE</td>
<td>8492</td>
</tr>
<tr>
<td>CHILDRESS</td>
<td>CHILDRESS</td>
<td>8328</td>
</tr>
<tr>
<td>MORTON</td>
<td>COCHRAN</td>
<td>8622</td>
</tr>
<tr>
<td>ROBERT LEE</td>
<td>COKE</td>
<td>8857</td>
</tr>
<tr>
<td>COLEMAN</td>
<td>COLEMAN</td>
<td>8804</td>
</tr>
<tr>
<td>FRISCO</td>
<td>COLLIN</td>
<td>8364</td>
</tr>
<tr>
<td>MCKINNEY</td>
<td>COLLIN</td>
<td>8340</td>
</tr>
<tr>
<td>PLANO</td>
<td>COLLIN</td>
<td>8343</td>
</tr>
<tr>
<td>WELLINGTON</td>
<td>COLLINASWH</td>
<td>8240</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run
(Continuation 1)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>COLUMBUS</td>
<td>COLORADO</td>
<td>9032</td>
<td>3744</td>
</tr>
<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>8289</td>
<td>4162</td>
</tr>
<tr>
<td>61</td>
<td>COPPERAS COVE</td>
<td>CORRY</td>
<td>8844</td>
<td>4099</td>
</tr>
<tr>
<td>62</td>
<td>GATESVILLE</td>
<td>CORRY</td>
<td>8771</td>
<td>4090</td>
</tr>
<tr>
<td>63</td>
<td>CRANE</td>
<td>CRANETTE</td>
<td>9144</td>
<td>4642</td>
</tr>
<tr>
<td>64</td>
<td>OZONA</td>
<td>DALLAS</td>
<td>8129</td>
<td>4249</td>
</tr>
<tr>
<td>65</td>
<td>DALHART</td>
<td>DALLAS</td>
<td>8494</td>
<td>4048</td>
</tr>
<tr>
<td>66</td>
<td>ADDISON</td>
<td>DALLAS</td>
<td>8495</td>
<td>4047</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>DESEO</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>4090</td>
</tr>
<tr>
<td>79</td>
<td>SEAGOVILLE</td>
<td>DALLAS</td>
<td>8353</td>
<td>4034</td>
</tr>
<tr>
<td>80</td>
<td>UNIVERSITY PARK</td>
<td>DALLAS</td>
<td>8379</td>
<td>4049</td>
</tr>
<tr>
<td>81</td>
<td>LAURA</td>
<td>DAWSON</td>
<td>8279</td>
<td>4019</td>
</tr>
<tr>
<td>82</td>
<td>HEREFORD</td>
<td>DEAF SMTH</td>
<td>8372</td>
<td>4127</td>
</tr>
<tr>
<td>83</td>
<td>DENTON</td>
<td>DENTON</td>
<td>8398</td>
<td>4093</td>
</tr>
<tr>
<td>84</td>
<td>LEWISVILLE</td>
<td>DENTON</td>
<td>8398</td>
<td>4093</td>
</tr>
<tr>
<td>85</td>
<td>CUPRO</td>
<td>DEWITT</td>
<td>8293</td>
<td>4032</td>
</tr>
<tr>
<td>86</td>
<td>SPOON</td>
<td>DICKENS</td>
<td>8560</td>
<td>4784</td>
</tr>
<tr>
<td>87</td>
<td>SAN DIEGO</td>
<td>DUVAL</td>
<td>9542</td>
<td>4388</td>
</tr>
<tr>
<td>88</td>
<td>EASTLAND</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</td>
</tr>
<tr>
<td>89</td>
<td>ODESSA</td>
<td>ECTOR</td>
<td>8982</td>
<td>4930</td>
</tr>
<tr>
<td>90</td>
<td>ENNIS</td>
<td>ELLIS</td>
<td>8514</td>
<td>3970</td>
</tr>
<tr>
<td>91</td>
<td>WAXAHACHIE</td>
<td>ELLIS</td>
<td>8517</td>
<td>4011</td>
</tr>
<tr>
<td>92</td>
<td>EL PASO</td>
<td>EL PASO</td>
<td>9231</td>
<td>5655</td>
</tr>
<tr>
<td>93</td>
<td>STEPHENVILLE</td>
<td>ERIATH</td>
<td>8645</td>
<td>4232</td>
</tr>
<tr>
<td>94</td>
<td>MARLIN</td>
<td>FALLS</td>
<td>8739</td>
<td>3931</td>
</tr>
<tr>
<td>95</td>
<td>BONHAM</td>
<td>FANNIN</td>
<td>8234</td>
<td>3966</td>
</tr>
<tr>
<td>96</td>
<td>RORY</td>
<td>FISHER</td>
<td>8679</td>
<td>4646</td>
</tr>
<tr>
<td>97</td>
<td>FLOYDADA</td>
<td>FLOYD</td>
<td>8486</td>
<td>4002</td>
</tr>
<tr>
<td>98</td>
<td>FAIRFIELD</td>
<td>FREESTONE</td>
<td>8602</td>
<td>3850</td>
</tr>
<tr>
<td>99</td>
<td>PEARSALL</td>
<td>FRIO</td>
<td>9374</td>
<td>4127</td>
</tr>
<tr>
<td>100</td>
<td>RICHMOND</td>
<td>FT BEND</td>
<td>9009</td>
<td>3508</td>
</tr>
<tr>
<td>101</td>
<td>ROSENBERG</td>
<td>FT BEND</td>
<td>9009</td>
<td>3508</td>
</tr>
<tr>
<td>102</td>
<td>SEMINOLE</td>
<td>CAINES</td>
<td>8822</td>
<td>6040</td>
</tr>
<tr>
<td>103</td>
<td>FRIENDSWOOD</td>
<td>GALVESTON</td>
<td>8946</td>
<td>3490</td>
</tr>
<tr>
<td>104</td>
<td>GALVESTON</td>
<td>GALVESTON</td>
<td>8985</td>
<td>3397</td>
</tr>
<tr>
<td>105</td>
<td>HITCHCOCK</td>
<td>GALVESTON</td>
<td>8927</td>
<td>3441</td>
</tr>
<tr>
<td>106</td>
<td>LA MARQUE</td>
<td>GALVESTON</td>
<td>8975</td>
<td>3424</td>
</tr>
<tr>
<td>107</td>
<td>LEAGUE CITY</td>
<td>GALVESTON</td>
<td>8967</td>
<td>3468</td>
</tr>
<tr>
<td>108</td>
<td>TEXAS CITY</td>
<td>GALVESTON</td>
<td>8975</td>
<td>3424</td>
</tr>
<tr>
<td>109</td>
<td>POST</td>
<td>GARZA</td>
<td>8650</td>
<td>4854</td>
</tr>
<tr>
<td>110</td>
<td>FREDERICKSBURG</td>
<td>GILLESPIE</td>
<td>9079</td>
<td>4194</td>
</tr>
<tr>
<td>111</td>
<td>GOLIAD</td>
<td>GOLIAD</td>
<td>9331</td>
<td>3847</td>
</tr>
<tr>
<td>112</td>
<td>GONZALES</td>
<td>GONZALES</td>
<td>9137</td>
<td>3844</td>
</tr>
<tr>
<td>113</td>
<td>PANAMA</td>
<td>GRAY</td>
<td>8148</td>
<td>4852</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run (Continuation 2)

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 114 | DENISON | GRAYSON | 3225 | 4069 |
| 115 | SHERMAN | GRAYSON | 8235 | 4072 |
| 116 | CLEMORE | GREGG | 3345 | 3679 |
| 117 | KILGORE | GREGG | 3372 | 3679 |
| 118 | LONGVIEW | GREGG | 3348 | 3660 |
| 119 | NAVASOTA | GRIMES | 8665 | 3715 |
| 120 | SEGUN | GUADALUPE | 9161 | 3981 |
| 121 | PLAINVIEW | HALE | 8465 | 4091 |
| 122 | HAMILTON | HAMILTON | 8744 | 4177 |
| 123 | SPEARMAN | HAMIS | 8026 | 5067 |
| 124 | QUANAH | HAMILTON | 8324 | 4654 |
| 125 | KOUNTZE | HARRIN | 8735 | 3405 |
| 126 | SILSBEE | HARRIS | 8730 | 3360 |
| 127 | BAYTOWN | HARRIS | 8916 | 3491 |
| 128 | BELLAIRE | HARRIS | 8938 | 3536 |
| 129 | DEEP PARK | HARRIS | 8929 | 3491 |
| 130 | GALENA PARK | HARRIS | 8938 | 3536 |
| 131 | HOUSTON | HARRIS | 8938 | 3536 |
| 132 | HUMBLE | HARRIS | 8881 | 3540 |
| 133 | JARINTO CITY | HARRIS | 8938 | 3536 |
| 134 | JERSEY VILLAGE | HARRIS | 8925 | 3581 |
| 135 | KATY | HARRIS | 8965 | 3614 |
| 136 | LA PORTE | HARRIS | 8892 | 3670 |
| 137 | PASADENA | HARRIS | 8938 | 3536 |
| 138 | SASA | HARRIS | 8945 | 3612 |
| 139 | SOUTH HOUSTON | HARRIS | 8938 | 3536 |
| 140 | SOUTHSIDE PLACE | HARRIS | 8938 | 3536 |
| 141 | SPRING VALLEY | HARRIS | 8938 | 3536 |
| 142 | TOMBALL | HARRIS | 8889 | 3409 |
| 143 | VILLAGE | HARRIS | 8938 | 3536 |
| 144 | WEBSTER | HARRIS | 8967 | 3564 |
| 145 | WEST UNIV PL | HARRIS | 8938 | 3536 |
| 146 | MARSHALL | HARRIS | 8311 | 3602 |
| 147 | HASKELL | HASKELL | 8554 | 4687 |
| 148 | SAN MARCOS | HAYS | 9006 | 4001 |
| 149 | CANADIAN | HEMPHILL | 8036 | 4822 |
| 150 | ATHENS | HENDERSON | 8644 | 3626 |
| 151 | DONNA | HIDALGO | 8649 | 3724 |
| 152 | EDINBURG | HIDALGO | 8980 | 3756 |
| 153 | Hidalgo | HIDALGO | 8980 | 3756 |
| 154 | MCALLEN | HIDALGO | 8966 | 3764 |
| 155 | MERCEDES | HIDALGO | 8965 | 3701 |
| 156 | MISSION | HIDALGO | 8961 | 3781 |
| 157 | PHARR | HIDALGO | 8964 | 3758 |
| 158 | KESLACO | HIDALGO | 8967 | 3716 |
| 159 | HILLSBORO | HILL | 8612 | 4026 |
| 160 | LEVELLAND | MOCKLEY | 8629 | 5053 |
| 161 | SULPHUR SPRINGS | HOPKINS | 8641 | 3861 |
| 162 | BIG SPRING | HOWARD | 8667 | 4000 |
| 163 | COMMERCE | HUNT | 8280 | 3921 |
| 164 | GRENVILLE | HUNT | 8317 | 3949 |
| 165 | BORGER | HUTCHINSON | 8146 | 5033 |
| 166 | JACKSON | JACKSON | 9186 | 3698 |
| 167 | EDNA | JEFFERSON | 8777 | 3444 |
| 168 | BEAUMONT | JEFFERSON | 8749 | 3316 |
| 169 | NENEDLAND | JEFFERSON | 8749 | 3316 |
| 170 | PORT ARTHUR | JEFFERSON | 8406 | 3298 |
Table 3-2. Input Date for the Example Run
(Continuation 3)

| 171  | ALICE JTM WELLS | 9539 | 3454 |
| 172  | BURLESON JOHNSON | 8522 | 4103 |
| 173  | CLEBURNE JOHNSON | 8563 | 4132 |
| 174  | ANSON JONES | 8647 | 4563 |
| 175  | STAMFORD JONES | 8603 | 4562 |
| 176  | KARNES CITY KARNES | 9264 | 3615 |
| 177  | KAUFMAN KAUFMAN | 8442 | 3936 |
| 178  | TERRELL KAUFMAN | 8410 | 3943 |
| 179  | BOFRINE KENDALL | 9168 | 4133 |
| 180  | JAYTON KENT | 8589 | 4718 |
| 181  | KEMPVILLE KERR | 9143 | 4226 |
| 182  | JUNCTION KIMBLE | 9097 | 4373 |
| 183  | KINGSVILLE KLEBERG | 9566 | 3801 |
| 184  | BENJAMIN KNOX | 8472 | 4610 |
| 185  | PARTS LAMAR | 8173 | 3997 |
| 186  | LITTLEFIELD LAMB | 8558 | 5049 |
| 187  | OLTON LAM | 8490 | 6044 |
| 188  | LAMPASAS LAMPASAS | 8675 | 4137 |
| 189  | HALLETTSVILLE LAVACA | 9114 | 3780 |
| 190  | YOAKUM LAVACA | 9157 | 3914 |
| 191  | CENTERVILLE LEON | 8662 | 3768 |
| 192  | CLYFELD LIFERTY | 8801 | 3540 |
| 193  | LIBERTY LIBERTY | 8435 | 3463 |
| 194  | MEYERS LIMESTONE | 8635 | 3888 |
| 195  | GEORGE WEST LIVE OAK | 9419 | 3910 |
| 196  | LIBROCK LUBBOCK | 8590 | 4962 |
| 197  | SLATON LUBBOCK | 8616 | 4916 |
| 198  | TAHOE LYNN | 8680 | 4924 |
| 199  | MADISONVILLE MADISON | 8740 | 3733 |
| 200  | JEFFERSON MARION | 8267 | 3619 |
| 201  | BAY CITY MATAGORDA | 9135 | 3378 |
| 202  | EAALE PASS MAVERICK | 9505 | 4370 |
| 203  | BRADY MCCULLOCH | 8938 | 4344 |
| 204  | BELLMEAD MCLENNAN | 8706 | 3903 |
| 205  | BEVERLY HILLS MCLENNAN | 8706 | 3903 |
| 206  | KACO MCLENNAN | 8706 | 3903 |
| 207  | WOODWAY MCLENNAN | 8706 | 3903 |
| 208  | HONDO MEDINA | 9265 | 4174 |
| 209  | MENARD MENARD | 9111 | 4417 |
| 210  | MISSION MIDLAND | 8949 | 4899 |
| 211  | CAMPAH MILAM | 8335 | 3910 |
| 212  | ROCKDALE MILAM | 8877 | 3808 |
| 213  | COLORADO CITY MITCHELL | 8781 | 4706 |
| 214  | BOWIE MONTAGUE | 8351 | 4275 |
| 215  | MONTAGUE MONTAGUE | 8323 | 4261 |
| 216  | CONROE MONTGOMERY | 8832 | 4600 |
| 217  | DUNAS MOORE | 8141 | 5144 |
| 218  | DANGERFIELD MORRIS | 8240 | 3704 |
| 219  | NACOGDOCHES NACOGDOCHES | 8515 | 3569 |
| 220  | CONSCINCANA NAVARRO | 8553 | 3921 |
| 221  | SWEETWATER NOLAN | 8777 | 4632 |
| 222  | CORPUS CHRISTI NUECES | 9475 | 3739 |
| 223  | RUSKSTOWN NUECES | 9496 | 3786 |
| 224  | PEPPRINT OCHILTREE | 7962 | 4987 |
| 225  | VEGA OLDHAM | 8292 | 5177 |
| 226  | ORANGE ORANGE | 8746 | 3281 |
| 227  | MINERAL WELLS PARKER | 8520 | 4261 |
Table 3-2. Input Data for the Example Run (Continuation 4)

<table>
<thead>
<tr>
<th>Code</th>
<th>City</th>
<th>County</th>
<th>Pop</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>228</td>
<td>CARTHAGE</td>
<td>PANOLA</td>
<td>8385</td>
<td>3564</td>
</tr>
<tr>
<td>229</td>
<td>WEATHERFORD</td>
<td>PARKER</td>
<td>8589</td>
<td>4206</td>
</tr>
<tr>
<td>230</td>
<td>FARNELL</td>
<td>PARKER</td>
<td>8500</td>
<td>5271</td>
</tr>
<tr>
<td>231</td>
<td>FRIONA</td>
<td>PARKER</td>
<td>8432</td>
<td>5165</td>
</tr>
<tr>
<td>232</td>
<td>FORT STOCKTON</td>
<td>PECOS</td>
<td>9207</td>
<td>4054</td>
</tr>
<tr>
<td>233</td>
<td>APARILLO</td>
<td>POTTER</td>
<td>8266</td>
<td>5876</td>
</tr>
<tr>
<td>234</td>
<td>CANYON</td>
<td>RANDALL</td>
<td>8517</td>
<td>5875</td>
</tr>
<tr>
<td>235</td>
<td>CLARKSVILLE</td>
<td>RED RIVER</td>
<td>9147</td>
<td>3899</td>
</tr>
<tr>
<td>236</td>
<td>PECOS</td>
<td>REEVES</td>
<td>9136</td>
<td>5101</td>
</tr>
<tr>
<td>237</td>
<td>BIG LAKE</td>
<td>REGAN</td>
<td>9062</td>
<td>4704</td>
</tr>
<tr>
<td>238</td>
<td>HEARNE</td>
<td>ROBERTSON</td>
<td>8802</td>
<td>3846</td>
</tr>
<tr>
<td>239</td>
<td>ROCKWALL</td>
<td>ROCKWALL</td>
<td>8391</td>
<td>3991</td>
</tr>
<tr>
<td>240</td>
<td>BALLINGER</td>
<td>RUNNELS</td>
<td>8655</td>
<td>4292</td>
</tr>
<tr>
<td>241</td>
<td>HENDERSON</td>
<td>RUSK</td>
<td>8420</td>
<td>3640</td>
</tr>
<tr>
<td>242</td>
<td>ARANSAS PASS</td>
<td>SAN PTD</td>
<td>9417</td>
<td>3700</td>
</tr>
<tr>
<td>243</td>
<td>GREGORY</td>
<td>SAN PTD</td>
<td>9435</td>
<td>3731</td>
</tr>
<tr>
<td>244</td>
<td>ILLINGSIDE</td>
<td>SAN PTD</td>
<td>9447</td>
<td>3711</td>
</tr>
<tr>
<td>245</td>
<td>PORTLAND</td>
<td>SAN PTD</td>
<td>9455</td>
<td>3731</td>
</tr>
<tr>
<td>246</td>
<td>STINSON</td>
<td>SAN PTD</td>
<td>9436</td>
<td>3777</td>
</tr>
<tr>
<td>247</td>
<td>EL PASO</td>
<td>SCHLEICHR</td>
<td>9076</td>
<td>4541</td>
</tr>
<tr>
<td>248</td>
<td>SNYDER</td>
<td>SCURRY</td>
<td>8718</td>
<td>4737</td>
</tr>
<tr>
<td>249</td>
<td>STRATFORD</td>
<td>SHERRIM</td>
<td>8049</td>
<td>4194</td>
</tr>
<tr>
<td>250</td>
<td>TYLER</td>
<td>SMITH</td>
<td>8417</td>
<td>3784</td>
</tr>
<tr>
<td>251</td>
<td>BRECKENRIDGE</td>
<td>STEPHENS</td>
<td>8582</td>
<td>4344</td>
</tr>
<tr>
<td>252</td>
<td>STERLING CITY</td>
<td>STEFELING</td>
<td>8900</td>
<td>4646</td>
</tr>
<tr>
<td>253</td>
<td>ASPERMONT</td>
<td>ST. NEWALL</td>
<td>8569</td>
<td>4650</td>
</tr>
<tr>
<td>254</td>
<td>TULIA</td>
<td>SWISHER</td>
<td>8397</td>
<td>5116</td>
</tr>
<tr>
<td>255</td>
<td>ARLINGTON</td>
<td>TARRANT</td>
<td>8472</td>
<td>4025</td>
</tr>
<tr>
<td>256</td>
<td>BEDFORD</td>
<td>TARRANT</td>
<td>8447</td>
<td>4002</td>
</tr>
<tr>
<td>257</td>
<td>COLEYVILLE</td>
<td>TARRANT</td>
<td>8447</td>
<td>4117</td>
</tr>
<tr>
<td>258</td>
<td>CROWLEY</td>
<td>TARRANT</td>
<td>8510</td>
<td>4118</td>
</tr>
<tr>
<td>259</td>
<td>EULESS</td>
<td>TARRANT</td>
<td>8447</td>
<td>4092</td>
</tr>
<tr>
<td>260</td>
<td>FORREST HILL</td>
<td>TARRANT</td>
<td>8479</td>
<td>4122</td>
</tr>
<tr>
<td>261</td>
<td>FORT WORTH</td>
<td>TARRANT</td>
<td>8479</td>
<td>4122</td>
</tr>
<tr>
<td>262</td>
<td>GRAPEVINE</td>
<td>TARRANT</td>
<td>8425</td>
<td>4004</td>
</tr>
<tr>
<td>263</td>
<td>HALTOM CITY</td>
<td>TARRANT</td>
<td>8470</td>
<td>4122</td>
</tr>
<tr>
<td>264</td>
<td>HURST</td>
<td>TARRANT</td>
<td>8447</td>
<td>4117</td>
</tr>
<tr>
<td>265</td>
<td>LAKE WORTH</td>
<td>TARRANT</td>
<td>8471</td>
<td>4158</td>
</tr>
<tr>
<td>266</td>
<td>N. RICHLAND HLS</td>
<td>TARRANT</td>
<td>8447</td>
<td>4117</td>
</tr>
<tr>
<td>267</td>
<td>RICHLAND HILLS</td>
<td>TARRANT</td>
<td>8479</td>
<td>4122</td>
</tr>
<tr>
<td>268</td>
<td>SOUTH LAKE</td>
<td>TARRANT</td>
<td>8425</td>
<td>4094</td>
</tr>
<tr>
<td>269</td>
<td>WHITE SETTLEMENT</td>
<td>TARRANT</td>
<td>8485</td>
<td>4153</td>
</tr>
<tr>
<td>270</td>
<td>ARLENE</td>
<td>TAYLOR</td>
<td>8698</td>
<td>4513</td>
</tr>
<tr>
<td>271</td>
<td>SANDERSON</td>
<td>TERRILL</td>
<td>9333</td>
<td>4816</td>
</tr>
<tr>
<td>272</td>
<td>BROWNFIELD</td>
<td>TERRY</td>
<td>8705</td>
<td>5007</td>
</tr>
<tr>
<td>273</td>
<td>MT PLEASANT</td>
<td>TITUS</td>
<td>8234</td>
<td>3755</td>
</tr>
<tr>
<td>274</td>
<td>SAN ANGELO</td>
<td>TOM GREEN</td>
<td>8994</td>
<td>4563</td>
</tr>
<tr>
<td>275</td>
<td>AUSTIN</td>
<td>TRAVIS</td>
<td>9005</td>
<td>4106</td>
</tr>
<tr>
<td>276</td>
<td>GILMER</td>
<td>UPSHUR</td>
<td>8317</td>
<td>3716</td>
</tr>
<tr>
<td>277</td>
<td>RANDIN</td>
<td>UPTON</td>
<td>9084</td>
<td>4811</td>
</tr>
<tr>
<td>278</td>
<td>UVALDE</td>
<td>UVALDE</td>
<td>9557</td>
<td>4279</td>
</tr>
<tr>
<td>279</td>
<td>DEL RIO</td>
<td>VAL VERDE</td>
<td>9397</td>
<td>4490</td>
</tr>
<tr>
<td>280</td>
<td>CANTON</td>
<td>VAN ZANDT</td>
<td>8415</td>
<td>3858</td>
</tr>
<tr>
<td>281</td>
<td>VICTORIA</td>
<td>VICTORIA</td>
<td>9265</td>
<td>3748</td>
</tr>
<tr>
<td>282</td>
<td>HUNTSVILLE</td>
<td>WALKER</td>
<td>8758</td>
<td>3652</td>
</tr>
<tr>
<td>283</td>
<td>HEMPSTEAD</td>
<td>WALLER</td>
<td>8923</td>
<td>3691</td>
</tr>
<tr>
<td>284</td>
<td>MONAHANS</td>
<td>WARD</td>
<td>9066</td>
<td>5065</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>285</td>
<td>BRENHAM</td>
<td>WASHINGTON</td>
<td>8942</td>
<td>3759</td>
</tr>
<tr>
<td>286</td>
<td>LAREDO</td>
<td>WEBB</td>
<td>9681</td>
<td>4099</td>
</tr>
<tr>
<td>287</td>
<td>PIERCE</td>
<td>WHARTON</td>
<td>9115</td>
<td>3649</td>
</tr>
<tr>
<td>288</td>
<td>WHARTON</td>
<td>WHARTON</td>
<td>9078</td>
<td>3670</td>
</tr>
<tr>
<td>289</td>
<td>SHARROCK</td>
<td>WHEELER</td>
<td>8170</td>
<td>4408</td>
</tr>
<tr>
<td>290</td>
<td>BURKBURNETT</td>
<td>WICHITA</td>
<td>8240</td>
<td>4440</td>
</tr>
<tr>
<td>291</td>
<td>WICHITA FALLS</td>
<td>WICHITA</td>
<td>8326</td>
<td>4413</td>
</tr>
<tr>
<td>292</td>
<td>VERNON</td>
<td>WILBARGER</td>
<td>8326</td>
<td>4567</td>
</tr>
<tr>
<td>293</td>
<td>RAYMONDVILLE</td>
<td>WILLACY</td>
<td>9768</td>
<td>3703</td>
</tr>
<tr>
<td>294</td>
<td>FLORESVILLE</td>
<td>WILSON</td>
<td>9261</td>
<td>3979</td>
</tr>
<tr>
<td>295</td>
<td>KERMIT</td>
<td>WINKLER</td>
<td>9024</td>
<td>5060</td>
</tr>
<tr>
<td>296</td>
<td>DECATHER</td>
<td>WISE</td>
<td>8399</td>
<td>4265</td>
</tr>
<tr>
<td>297</td>
<td>WINNSBORO</td>
<td>WOOD</td>
<td>8295</td>
<td>3794</td>
</tr>
<tr>
<td>298</td>
<td>DENVER CITY</td>
<td>YOAKUM</td>
<td>8781</td>
<td>5078</td>
</tr>
<tr>
<td>299</td>
<td>GRAHAM</td>
<td>YOUNG</td>
<td>8492</td>
<td>4365</td>
</tr>
<tr>
<td>300</td>
<td>OLNEY</td>
<td>YOUNG</td>
<td>8450</td>
<td>4414</td>
</tr>
<tr>
<td>301</td>
<td>CRYSTAL CITY</td>
<td>ZAVALA</td>
<td>9646</td>
<td>4246</td>
</tr>
<tr>
<td>302</td>
<td>ARANSAS PASS</td>
<td>ARANSAS</td>
<td>9437</td>
<td>3700</td>
</tr>
<tr>
<td>303</td>
<td>PLEASANTON</td>
<td>ATASCOSA</td>
<td>9320</td>
<td>4027</td>
</tr>
<tr>
<td>304</td>
<td>BANCERA</td>
<td>BANDERA</td>
<td>9255</td>
<td>4190</td>
</tr>
<tr>
<td>305</td>
<td>BASTROP</td>
<td>BASTROP</td>
<td>9077</td>
<td>3026</td>
</tr>
<tr>
<td>306</td>
<td>MARBLE FALLS</td>
<td>BURNET</td>
<td>8960</td>
<td>4115</td>
</tr>
<tr>
<td>307</td>
<td>LOCKHART</td>
<td>CALDWELL</td>
<td>9077</td>
<td>3954</td>
</tr>
<tr>
<td>308</td>
<td>LULING</td>
<td>CALDWELL</td>
<td>9117</td>
<td>3933</td>
</tr>
<tr>
<td>309</td>
<td>BARD</td>
<td>CALLAHAN</td>
<td>8684</td>
<td>4450</td>
</tr>
<tr>
<td>310</td>
<td>PITTSBURG</td>
<td>CAMP</td>
<td>8264</td>
<td>3742</td>
</tr>
<tr>
<td>311</td>
<td>P AHANDLE</td>
<td>CARSON</td>
<td>8210</td>
<td>5009</td>
</tr>
<tr>
<td>312</td>
<td>ATLANTA</td>
<td>CASS</td>
<td>8182</td>
<td>3618</td>
</tr>
<tr>
<td>313</td>
<td>RUSK</td>
<td>CHEROKEE</td>
<td>8515</td>
<td>3672</td>
</tr>
<tr>
<td>314</td>
<td>HENRIETTA</td>
<td>CLAY</td>
<td>8323</td>
<td>4354</td>
</tr>
<tr>
<td>315</td>
<td>CROSBYTON</td>
<td>CROSBY</td>
<td>8548</td>
<td>3462</td>
</tr>
<tr>
<td>316</td>
<td>CAPROLLTON</td>
<td>DALLAS</td>
<td>8410</td>
<td>4066</td>
</tr>
<tr>
<td>317</td>
<td>COOPER</td>
<td>DELTA</td>
<td>8241</td>
<td>3806</td>
</tr>
<tr>
<td>318</td>
<td>CARRIZO SPRS.</td>
<td>DURMIT</td>
<td>9540</td>
<td>4290</td>
</tr>
<tr>
<td>319</td>
<td>C C</td>
<td>EASTLAND</td>
<td>8662</td>
<td>4377</td>
</tr>
<tr>
<td>320</td>
<td>LA GRAMNE</td>
<td>FAYETTE</td>
<td>9016</td>
<td>3813</td>
</tr>
<tr>
<td>321</td>
<td>MT VERNON</td>
<td>FRANKLIN</td>
<td>8246</td>
<td>3801</td>
</tr>
<tr>
<td>322</td>
<td>ANDERSON</td>
<td>GRIMES</td>
<td>8816</td>
<td>3708</td>
</tr>
<tr>
<td>323</td>
<td>MEMPHIS</td>
<td>HALL</td>
<td>8287</td>
<td>4021</td>
</tr>
<tr>
<td>324</td>
<td>ALAMO</td>
<td>HITALO</td>
<td>9054</td>
<td>3754</td>
</tr>
<tr>
<td>325</td>
<td>GRANBURY</td>
<td>HOOD</td>
<td>8572</td>
<td>4178</td>
</tr>
<tr>
<td>326</td>
<td>CROCKETT</td>
<td>HOUSTON</td>
<td>8634</td>
<td>3645</td>
</tr>
<tr>
<td>327</td>
<td>STINNETT</td>
<td>HUTCHINSON</td>
<td>8117</td>
<td>5054</td>
</tr>
<tr>
<td>328</td>
<td>JASPER</td>
<td>JASPER</td>
<td>8643</td>
<td>3349</td>
</tr>
<tr>
<td>329</td>
<td>GROVES</td>
<td>JEFFERSON</td>
<td>8789</td>
<td>3316</td>
</tr>
<tr>
<td>330</td>
<td>PORT NECHES</td>
<td>JEFFERSON</td>
<td>8789</td>
<td>3316</td>
</tr>
<tr>
<td>331</td>
<td>COTULLA</td>
<td>LASALLE</td>
<td>9476</td>
<td>4120</td>
</tr>
<tr>
<td>332</td>
<td>GININGS</td>
<td>LEE</td>
<td>8968</td>
<td>3488</td>
</tr>
<tr>
<td>333</td>
<td>GROSFBECK</td>
<td>LIMESTORNE</td>
<td>8671</td>
<td>3686</td>
</tr>
<tr>
<td>334</td>
<td>LLANO</td>
<td>LLANO</td>
<td>8970</td>
<td>4199</td>
</tr>
<tr>
<td>335</td>
<td>NEWTON</td>
<td>NEWTON</td>
<td>8660</td>
<td>3353</td>
</tr>
<tr>
<td>336</td>
<td>BRIDGE CITY</td>
<td>ORANGF</td>
<td>8774</td>
<td>3295</td>
</tr>
<tr>
<td>337</td>
<td>VIOR</td>
<td>ORANGF</td>
<td>8761</td>
<td>3334</td>
</tr>
<tr>
<td>338</td>
<td>PALO PINTO</td>
<td>PLO PINTO</td>
<td>8541</td>
<td>4291</td>
</tr>
<tr>
<td>339</td>
<td>LIVINGSTON</td>
<td>POLK</td>
<td>8716</td>
<td>3543</td>
</tr>
<tr>
<td>340</td>
<td>REFUGIO</td>
<td>RIFFUGIO</td>
<td>9345</td>
<td>3747</td>
</tr>
<tr>
<td>341</td>
<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>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEMPHILL</td>
<td>342</td>
<td>8511</td>
<td>3413</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAN AUGUSTINE</td>
<td>343</td>
<td>8491</td>
<td>3471</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLDSPRING</td>
<td>344</td>
<td>8754</td>
<td>3567</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAN SABA</td>
<td>345</td>
<td>8886</td>
<td>4242</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATHIS</td>
<td>346</td>
<td>9448</td>
<td>3444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CENTER</td>
<td>347</td>
<td>9443</td>
<td>3505</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIO GRANDE CY</td>
<td>348</td>
<td>9861</td>
<td>3887</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERNROOK</td>
<td>349</td>
<td>8499</td>
<td>4140</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVERMAN</td>
<td>350</td>
<td>8565</td>
<td>4116</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIVEROAKS</td>
<td>351</td>
<td>8479</td>
<td>4122</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROVETON</td>
<td>352</td>
<td>8661</td>
<td>3685</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WOODVILLE</td>
<td>353</td>
<td>9664</td>
<td>3858</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EL CAMPO</td>
<td>354</td>
<td>9115</td>
<td>3849</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WHEELER</td>
<td>355</td>
<td>8126</td>
<td>4020</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IOWA PARK</td>
<td>356</td>
<td>8327</td>
<td>4345</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GEORGETOWN</td>
<td>357</td>
<td>8927</td>
<td>4014</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAYLOR</td>
<td>358</td>
<td>8922</td>
<td>3962</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUITMAN</td>
<td>359</td>
<td>8340</td>
<td>3806</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLAINES</td>
<td>360</td>
<td>8735</td>
<td>5105</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZAPATA</td>
<td>361</td>
<td>9746</td>
<td>4089</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AZLI MULESHOE PD</td>
<td>362</td>
<td>7.35</td>
<td>19.23</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>A0K MORTON PD</td>
<td>363</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZKW SPUR PD</td>
<td>365</td>
<td>47</td>
<td>4.38</td>
<td>10.58</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZK POST SO</td>
<td>366</td>
<td>83</td>
<td>6.28</td>
<td>17.40</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL PLAINVIEW PD</td>
<td>368</td>
<td>94</td>
<td>5.60</td>
<td>11.91</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL PLAINVIEW SO</td>
<td>370</td>
<td>106</td>
<td>2.86</td>
<td>8.16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AZL LEVELLAND PD</td>
<td>371</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>372</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>373</td>
<td>118</td>
<td>25.27</td>
<td>55.61</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL PLAINVIEW PD</td>
<td>374</td>
<td>118</td>
<td>7.67</td>
<td>14.67</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL PLAINVIEW SO</td>
<td>375</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>376</td>
<td>157</td>
<td>10.45</td>
<td>27.80</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>377</td>
<td>183</td>
<td>8.47</td>
<td>21.04</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>378</td>
<td>183</td>
<td>4.57</td>
<td>11.05</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>379</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>380</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>381</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>382</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>383</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>384</td>
<td>193</td>
<td>31.12</td>
<td>94.55</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>385</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>386</td>
<td>193</td>
<td>242.28</td>
<td>340.68</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>387</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>388</td>
<td>193</td>
<td>39.61</td>
<td>59.13</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>389</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>390</td>
<td>194</td>
<td>10.01</td>
<td>23.71</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>391</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>392</td>
<td>195</td>
<td>5.72</td>
<td>17.21</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>393</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>394</td>
<td>195</td>
<td>4.03</td>
<td>9.94</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>395</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>396</td>
<td>269</td>
<td>17.09</td>
<td>40.40</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AZL LITTLEFIELD SO</td>
<td>397</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AZL LITTLEFIELD PD</td>
<td>398</td>
<td>269</td>
<td>4.03</td>
<td>8.36</td>
<td>0.00</td>
<td>0.00</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>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>399</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>400</td>
<td>AZLN DENVER CITY PD</td>
<td>295</td>
<td>7.39</td>
</tr>
<tr>
<td>401</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>402</td>
<td>NAAE MULESHOE S.O.</td>
<td></td>
<td>8.0</td>
</tr>
<tr>
<td>403</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>404</td>
<td>NABG CROSBYTON S.O.</td>
<td>312</td>
<td>1.66</td>
</tr>
<tr>
<td>405</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>406</td>
<td>NACG LEVELLAND S.O.</td>
<td></td>
<td>157</td>
</tr>
<tr>
<td>407</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>408</td>
<td>NAFA PLAINES S.O.</td>
<td></td>
<td>357</td>
</tr>
<tr>
<td>409</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>410</td>
<td>AAA, DUSTIN SWITCHER</td>
<td>272</td>
<td></td>
</tr>
</tbody>
</table>

3-20
Table 3-2. Input Data for the Example Run
(Continuation 8)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>456</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>457</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>458</td>
<td>0</td>
<td>0</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>10</td>
<td>10</td>
</tr>
<tr>
<td>464</td>
<td>10</td>
<td>10</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>10</td>
<td>10</td>
</tr>
<tr>
<td>474</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>475</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>476</td>
<td>15</td>
<td>15</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>10</td>
<td>10</td>
</tr>
<tr>
<td>492</td>
<td>10</td>
<td>10</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>10</td>
<td>10</td>
</tr>
<tr>
<td>496</td>
<td>10</td>
<td>10</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>10</td>
<td>10</td>
</tr>
<tr>
<td>500</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>501</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>502</td>
<td>10</td>
<td>10</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>10</td>
<td>10</td>
</tr>
<tr>
<td>510</td>
<td>10</td>
<td>10</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)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>513</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>514</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>515</td>
<td>135</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>516</td>
<td>135</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>517</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>519</td>
<td>135</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>520</td>
<td>135</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>521</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>522</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>523</td>
<td>135</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>524</td>
<td>135</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>525</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>526</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>527</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>528</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>529</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>530</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>531</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>532</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>533</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>534</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>535</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>536</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>537</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>538</td>
<td></td>
<td></td>
<td></td>
<td>1200 BAUD LINE</td>
<td></td>
</tr>
<tr>
<td>539</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>540</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>541</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>542</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>543</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>544</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>545</td>
<td></td>
<td></td>
<td></td>
<td>2400 BAUD LINE</td>
<td></td>
</tr>
<tr>
<td>546</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>547</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>548</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>549</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>550</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>551</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>552</td>
<td></td>
<td></td>
<td></td>
<td>4800 BAUD LINE</td>
<td></td>
</tr>
<tr>
<td>553</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>554</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>555</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>556</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>557</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>558</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>559</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>560</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>561</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>562</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>563</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>564</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>565</td>
<td>TCIC</td>
<td>60</td>
<td>8615.25</td>
<td>14.47</td>
<td>0</td>
</tr>
<tr>
<td>566</td>
<td>ADM</td>
<td>500</td>
<td>500</td>
<td>3.62</td>
<td>3.62</td>
</tr>
<tr>
<td>567</td>
<td>G-CODE</td>
<td>300</td>
<td>300</td>
<td>1.13</td>
<td>8.43</td>
</tr>
<tr>
<td>568</td>
<td>CCH</td>
<td>426</td>
<td>459</td>
<td>5.70</td>
<td>5.70</td>
</tr>
<tr>
<td>569</td>
<td>MVNP</td>
<td>50</td>
<td>175</td>
<td>9.34</td>
<td>9.34</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run (Continuation 10)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Values</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>570</td>
<td>1st line of data</td>
<td>ING/NL</td>
<td>50 200 .45 .45 0 0 .0</td>
</tr>
<tr>
<td>571</td>
<td>2nd line of data</td>
<td>ADM/NL</td>
<td>300 300 .45 .45 0 0 .0</td>
</tr>
<tr>
<td>572</td>
<td>3rd line of data</td>
<td>NCIC</td>
<td>50 90 .4 9.21 0 0 .0</td>
</tr>
<tr>
<td>573</td>
<td>4th line of data</td>
<td>DB/DLS</td>
<td>90 50 3.49 1.1 0 0 .0</td>
</tr>
<tr>
<td>574</td>
<td>5th line of data</td>
<td>DB/ANT</td>
<td>90 50 .45 .45 0 0 .0</td>
</tr>
<tr>
<td>575</td>
<td>6th line of data</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>576</td>
<td>7th line of data</td>
<td>2AAAA</td>
<td>1</td>
</tr>
<tr>
<td>577</td>
<td>8th line of data</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>578</td>
<td>9th line of data</td>
<td>4AAAA</td>
<td>QSTATE CENTER</td>
</tr>
<tr>
<td>579</td>
<td>10th line of data</td>
<td>6.45</td>
<td>2 TOTAL XSAC &amp; REQ. AT THE AUSTIN SWITCHER WITH F8.5 A13</td>
</tr>
<tr>
<td>580</td>
<td>11th line of data</td>
<td>2AAA</td>
<td>1 GTERMINATE LINE/RESP. TIME: MPROC WITH 13;F5;2;I2</td>
</tr>
<tr>
<td>581</td>
<td>12th line of data</td>
<td>1</td>
<td>1 FOR PLOTTING AND 0 FOR SKIPPING IT</td>
</tr>
</tbody>
</table>
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>Term</th>
<th>AZLI</th>
<th>AZKK</th>
<th>AZKW</th>
<th>AZKZ</th>
<th>AZLD</th>
<th>AZLA</th>
<th>AZLD</th>
<th>AZLC</th>
<th>AZKA</th>
<th>AZ81</th>
<th>AZGL</th>
<th>AZLK</th>
<th>AZLL</th>
<th>AZLF</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAFIN</td>
<td>1.0</td>
<td>.0</td>
<td>.6</td>
<td>1.6</td>
<td>3.8</td>
<td>1.0</td>
<td>1.4</td>
<td>1.2</td>
<td>.6</td>
<td>.7</td>
<td>4.1</td>
<td>39.3</td>
<td>5.3</td>
<td>1.9</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>1.5</td>
<td>2.1</td>
<td>15.1</td>
<td>45.4</td>
<td>7.4</td>
<td>3.2</td>
</tr>
<tr>
<td>TRAFIN</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
</tr>
<tr>
<td>TRFOUT</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
</tr>
<tr>
<td>TRAFIN</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
</tr>
<tr>
<td>TRFOUT</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
</tr>
<tr>
<td>TRAFIN</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
</tr>
<tr>
<td>TRFOUT</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run
(Continuation 1)

<table>
<thead>
<tr>
<th></th>
<th>AZLI</th>
<th>AZLK</th>
<th>AZKV</th>
<th>AZKZ</th>
<th>A110</th>
<th>A111</th>
</tr>
</thead>
<tbody>
<tr>
<td>A111</td>
<td>2.56M</td>
<td>3.61M</td>
<td>1.31M</td>
<td>2.75M</td>
<td>4.13M</td>
<td>6.62M</td>
</tr>
<tr>
<td>A110</td>
<td>7.91M</td>
<td>1.19M</td>
<td>3.76M</td>
<td>2.89M</td>
<td>2.58M</td>
<td>1.87M</td>
</tr>
<tr>
<td>A111</td>
<td>2.29M</td>
<td>1.32M</td>
<td>5.45M</td>
<td>4.50M</td>
<td>2.24M</td>
<td>2.78M</td>
</tr>
<tr>
<td>A110</td>
<td>1.90M</td>
<td>1.70M</td>
<td>1.13M</td>
<td>1.68M</td>
<td>2.00M</td>
<td>1.30M</td>
</tr>
</tbody>
</table>

**TOTAL TRAFFIC ORIGINATED FROM SYS. Termin. (BITES/SEC)**

**TOTAL TRAFFIC DESTINATED TO SYS. Termin. (BITES/SEC)**

**TOTAL SYSTEM TRAFFIC** 112.30
Table 3-3. Printer Output from the Example Run (Continuation 2)

<table>
<thead>
<tr>
<th>POINT TO POINT DISTANCE MATRIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERM</td>
</tr>
<tr>
<td>AZLI</td>
</tr>
<tr>
<td>AZXL</td>
</tr>
<tr>
<td>AZLT</td>
</tr>
<tr>
<td>AZLL</td>
</tr>
<tr>
<td>AZKI</td>
</tr>
<tr>
<td>AZKZ</td>
</tr>
<tr>
<td>AZKL</td>
</tr>
<tr>
<td>AZKL</td>
</tr>
<tr>
<td>AZKZ</td>
</tr>
<tr>
<td>AZKL</td>
</tr>
<tr>
<td>AZKL</td>
</tr>
<tr>
<td>AZKL</td>
</tr>
</tbody>
</table>

CPU UTILIZATION ON PROCESSOR IS .707
Table 3-3. Printer Output from the Example Run  
(Continuation 3)

<table>
<thead>
<tr>
<th>SYSTEM TERMIN.</th>
<th>AGL1</th>
<th>AZXX</th>
<th>AZKL</th>
<th>AGL2</th>
<th>AGLA</th>
<th>AGLM</th>
<th>AGLD</th>
<th>AZLC</th>
<th>AZKA</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. OF LINES AFD.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,2KR</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2,4KR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4,8KR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LINE UTILIZATION</td>
<td>0.984</td>
<td>0.983</td>
<td>0.984</td>
<td>0.903</td>
<td>0.910</td>
<td>0.983</td>
<td>0.985</td>
<td>0.984</td>
<td>0.984</td>
</tr>
<tr>
<td>DISTANCE FROM HSC</td>
<td>309</td>
<td>370</td>
<td>267</td>
<td>313</td>
<td>294</td>
<td>356</td>
<td>356</td>
<td>356</td>
<td>356</td>
</tr>
<tr>
<td>TRAFFIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE TO CPU</td>
<td>7.962</td>
<td>1.411</td>
<td>2.373</td>
<td>1.596</td>
<td>1.064</td>
<td>7.453</td>
<td>1.906</td>
<td>3.707</td>
<td>31.909</td>
</tr>
<tr>
<td>CPU TO LINE</td>
<td>5.562</td>
<td>2.111</td>
<td>2.373</td>
<td>1.596</td>
<td>1.064</td>
<td>7.453</td>
<td>1.906</td>
<td>3.707</td>
<td>31.909</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INST. COSTS</th>
<th>SUPTOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>0</td>
</tr>
<tr>
<td>SER.T.</td>
<td>20</td>
</tr>
<tr>
<td>MODEM</td>
<td>2472</td>
</tr>
<tr>
<td>DROP</td>
<td>0</td>
</tr>
<tr>
<td>ANNUAL RECU. COST</td>
<td>229708</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSTEM TERMIN.</th>
<th>AGL1</th>
<th>AZXX</th>
<th>AZKL</th>
<th>AGL2</th>
<th>AGLA</th>
<th>AGLM</th>
<th>AGLD</th>
<th>AZLC</th>
<th>AZKA</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. OF LINES AFD.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,2KR</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2,4KR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4,8KR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LINE UTILIZATION</td>
<td>0.003</td>
<td>0.014</td>
<td>0.019</td>
<td>0.012</td>
<td>0.005</td>
<td>0.004</td>
<td>0.003</td>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>DISTANCE FROM HSC</td>
<td>373</td>
<td>332</td>
<td>332</td>
<td>332</td>
<td>332</td>
<td>332</td>
<td>332</td>
<td>332</td>
<td>332</td>
</tr>
<tr>
<td>TRAFFIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE TO CPU</td>
<td>7.703</td>
<td>4.294</td>
<td>4.294</td>
<td>5.294</td>
<td>5.294</td>
<td>5.294</td>
<td>5.294</td>
<td>5.294</td>
<td>5.294</td>
</tr>
<tr>
<td>CPU TO LINE</td>
<td>5.150</td>
<td>13.165</td>
<td>4.547</td>
<td>7.888</td>
<td>7.888</td>
<td>7.888</td>
<td>7.888</td>
<td>7.888</td>
<td>7.888</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INST. COSTS</th>
<th>SUPTOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>0</td>
</tr>
<tr>
<td>SER.T.</td>
<td>20</td>
</tr>
<tr>
<td>MODEM</td>
<td>100</td>
</tr>
<tr>
<td>DROP</td>
<td>0</td>
</tr>
<tr>
<td>ANNUAL RECU. COST</td>
<td>15430</td>
</tr>
</tbody>
</table>
### Table 3-3. Printer Output from the Example Run (Continuation 4)

<table>
<thead>
<tr>
<th>SYSTEM TERMIN.</th>
<th>NAAL</th>
<th>NAGP</th>
<th>NAGA</th>
<th>NAFA</th>
<th>NAAM</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>1</td>
</tr>
<tr>
<td>2.4KB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.8KB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DISTANCE FROM SRC 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>.357</td>
<td>.204</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>3.275</td>
<td>3.275</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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MODEM</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DROP</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>ANNUAL INFLAT. COST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>1436A</td>
<td>11160</td>
<td>12780</td>
<td>12096</td>
<td>0</td>
</tr>
<tr>
<td>MODEM</td>
<td>528</td>
<td>528</td>
<td>528</td>
<td>528</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>14424</td>
<td>12288</td>
<td>13408</td>
<td>13124</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL COSTS</td>
<td>259782</td>
<td>259782</td>
<td>259782</td>
<td>259782</td>
<td>0</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>AZLL</td>
<td>AZKS</td>
</tr>
<tr>
<td>NO. OF TERMINALS</td>
<td>19</td>
<td>6</td>
</tr>
<tr>
<td>NO. OF LINES</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1.2 KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.4 KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.8 KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LINE UTILIZATION</td>
<td>.138</td>
<td>.017</td>
</tr>
<tr>
<td>TOTAL MILEAGE</td>
<td>617</td>
<td>385</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>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>0</td>
</tr>
<tr>
<td>SER.T.</td>
<td>260</td>
</tr>
<tr>
<td>MODEM</td>
<td>1390</td>
</tr>
<tr>
<td>DкоРР</td>
<td>0</td>
</tr>
<tr>
<td>ANNUAL RECURR. COST</td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>16422</td>
</tr>
<tr>
<td>SER.T.</td>
<td>6430</td>
</tr>
<tr>
<td>MODEM</td>
<td>6884</td>
</tr>
<tr>
<td>DкоРР</td>
<td>3120</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>31086</td>
</tr>
<tr>
<td>INST. COST</td>
<td>1560</td>
</tr>
<tr>
<td>RECUR. COST</td>
<td>31086</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>AZLI</th>
<th>AZLP</th>
<th>AZLA</th>
<th>AZZE</th>
<th>NACG</th>
<th>AZKW</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZTI</td>
<td>AZKA</td>
<td>NACG</td>
<td>AZXX</td>
<td>AGLD</td>
<td>AGL1</td>
</tr>
<tr>
<td>AZLK</td>
<td>AZLG</td>
<td>AGLR</td>
<td>AZLD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AZKS</td>
<td>AZLJ</td>
<td>ACLF</td>
<td>NAFA</td>
<td>AZLE</td>
<td>AZLI</td>
</tr>
</tbody>
</table>

REGONAL CENTRE: AAAA

SUBNETWORK BEGINS AT
individual line type under consideration. For example, a modem turn-
around time of 50 milli-seconds 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 0.004 and response time 3.279 seconds. It is 399 miles away from AAAA. Based on the tariff applicable for Texas, its 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 milage. 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>ASSUMED NUMBER OF REGIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTER NR AND SPIKE RETURN KEY</td>
</tr>
<tr>
<td>TYPE IN NO. OR NR PER TOWN: DATA BASES AND CITIES WITH FORMAT 915</td>
</tr>
<tr>
<td>THERE ARE 25 SYS. TERMS: DATA BASES 35H CITIES</td>
</tr>
<tr>
<td>TYPE IN DATA BASE LOCATIONS WITH FORMAT 915X066</td>
</tr>
<tr>
<td>4 DATA BASES ARE AT AAAA ID 5555 SIZE</td>
</tr>
<tr>
<td>TYPE IN CITY VR WITH FORMAT 915X066</td>
</tr>
<tr>
<td>4 DATA BASES ARE AT AAAA ID 5555</td>
</tr>
<tr>
<td>TYPE IN NO. OF NAME MAPPING ADDR. AND TRAFFIC</td>
</tr>
<tr>
<td>WITH FORMAT 915X066</td>
</tr>
<tr>
<td>TYPE II NO. OF RATE STRUCTURES UNDER CONSIDERATION WITH FORMAT 913</td>
</tr>
<tr>
<td>TYPE IN RATE APPLICATION TO EACH COMMON</td>
</tr>
<tr>
<td>WHIT EACH SYS. TERM, WITH FORMAT 9112</td>
</tr>
<tr>
<td>READ IN TRAFFIC DENSITY TYPE AND RATE STRUCTURE</td>
</tr>
<tr>
<td>FOR EACH CII WITH FORMAT 9111</td>
</tr>
<tr>
<td>TYPE IN NO. OF LINE TYPES APPLICABLE WITH FORMAT 913</td>
</tr>
<tr>
<td>TYPE IN NAME CAPACITY UTIL. FACTOR AVAIL. FOR</td>
</tr>
<tr>
<td>EACH LINE TYPE WITH FORMAT 915X066</td>
</tr>
<tr>
<td>TYPE IN NO. OF DEVICES AND NAMES FOR EACH LINE TYPE</td>
</tr>
<tr>
<td>WITH FORMAT 913/914/915/916/917/918/919</td>
</tr>
<tr>
<td>TYPE IN INST. AND RECUR. COSTS WHT</td>
</tr>
<tr>
<td>RATE STRUCTURE: LINE TYPE: DEVICE: TRAFFIC DENSITY</td>
</tr>
<tr>
<td>AND DUXLEXING MORE WITH FORMAT 9104/9105/9106/9107/9108/9109/9110</td>
</tr>
<tr>
<td>TYPE IN INST. COSTS OF INES WHT</td>
</tr>
<tr>
<td>RATE: LINE: DENSITY: DU: DUXLEXING MORE</td>
</tr>
<tr>
<td>WITH FORMAT 9102</td>
</tr>
<tr>
<td>TYPE IN INDEX FOR LINEARITY OF LINE RECUR. COST</td>
</tr>
<tr>
<td>FUNCTION WITH 1 LINEAR AND NONLINEAR OTHERWISE</td>
</tr>
<tr>
<td>WITH FORMAT 911 FOR EACH LINE TYPE</td>
</tr>
<tr>
<td>TYPE IN RECUR. COSTS WITH FORMAT 9102 IF LINEAR</td>
</tr>
<tr>
<td>WITH FORMAT 9103/9104/9105/9106/9107/9108/9109/9110/9111/9112</td>
</tr>
<tr>
<td>IF NONLINEAR USE 9102/911</td>
</tr>
<tr>
<td>TYPE IN ACTION IDES FOR EACH REGION</td>
</tr>
<tr>
<td>1ST ELEMENT: 1 = INSERTION TO THIS PRELOADED REGION IS OK</td>
</tr>
<tr>
<td>2ND ELEMENT: 2 = OPTIMIZATION IS NEEDED</td>
</tr>
<tr>
<td>TYPE IN REGION INDEX AND ACTION NUMBER FOR</td>
</tr>
<tr>
<td>WITH FORMAT 9130 AND END IT WITH A 0</td>
</tr>
<tr>
<td>TYPE IN NP: MAX: AC: NAKO: NOKO: MHI:</td>
</tr>
<tr>
<td>TAN: TAO IN FORMAT 914/927/5</td>
</tr>
<tr>
<td>TYPE III: NO. OF MSG TYPES: TRAFFIC STATISTICS</td>
</tr>
<tr>
<td>SUCH AS MSGHAM, WSLIN: WSLIN: HAMLIN: HAMLIN:</td>
</tr>
<tr>
<td>WITH FORMAT 910/911/912/913/914/915/916/917/918/919</td>
</tr>
<tr>
<td>TYPE IN PRELOADED SYSTEM TERMS: AND RSC WITH</td>
</tr>
<tr>
<td>FORMAT 911/912/913/914/915/916/917/918/919</td>
</tr>
<tr>
<td>3233 DISTANCE TERMS ARE OVERSIZED</td>
</tr>
</tbody>
</table>

ASSUME A SYSTEM CENTERED |
ENTER CODE FOR HSCC AND SPIKE RETURN KEY |
INPUT TOTAL NO. OF TRANSACTIONS AND NO. OF ACCESS AT THE SWITCHER |
ENTER WITH 905 AND 13 UNDER XSAC/SC |
READ IN LIMITS ON NO. OF SYS. TERMS, ON A 0 |
RESPONSE TIME READ AND NO. OF PROCESSIONS WITH FORMAT |
13+6+6+12 |
IF PLOTTING IS REQUIRED: TYPE 1 WITH FORMAT 13 |
TRYLINK HAS BEEN ACCESS FOR 10005 TIMES |
UPHETN HAS BEEN ACCESS FOR 25 TIMES |

GIMME THIS 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

5192A*STACOM(1),MAIN/0777

C******************************************************************************C
C
C* STACOM TOPOLOGY PROGRAM *
C
C* JET PROPULSION LABORATORY *
C* 4800 OAK GROVE DRIVE *
C* PASADENA, CALIFORNIA 91103 *
C
C******************************************************************************C
C
THIS PROGRAM IS DESIGNED TO PERFORM FORMATIONS OF REGIONS, SELECTIONS
OF REGIONAL SWITCHING CENTERS, FORMATIONS OF INITIAL REGIONAL NETWORKS,
OPTIMIZATION OF REGIONAL NETWORKS USING THE FRAH-WILLIAMS METHOD IF
REQUESTED, AND FINALLY FORMATION OF AN INTERREGIONAL NETWORK AND ITS
OPTIMIZATION.

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

C THIS TOPOLOGY PROGRAM CONTAINS ONE MAIN PROGRAM AND ELEVEN SUBPROGRAMS.
C THEY ARE AS FOLLOWS:
C MAIN PROGRAM : MAIN (REGION ASSIGNMENTS, SYSTEM TERMINATIONS)
C SUBPROGRAM-1: REGMET (REGIONAL NETWORK OPTIMIZATION AND ITS OPTIMIZATION)
C SUBPROGRAM-2: IRRNP (INTERREGIONAL NETWORK OPTIMIZATION)
C SUBPROGRAM-3: ICOSTJ (COSTING FUNCTION)
C SUBPROGRAM-4: RHOFUN (LINE UTILIZATION FUNCTION)
C SUBPROGRAM-5: LININF (LINE CONFIG, DEFINITION BASED ON TRAFFIC)
C SUBPROGRAM-6: PACK (STORING OR RETYPING DISTANCE DATA)
C SUBPROGRAM-7: DIST (FINDING DISTANCE BETWEEN TWO GIVEN TERMINALS)
C SUBPROGRAM-8: LINK (FINDING COMPRESSED INDEX FOR DIST)
C SUBPROGRAM-9: RECOVR (RECOVERING COMPRESSED DISTANCE DATA)
C SUBPROGRAM-10: PLOTPT (PLOTTING EACH DROP ON A MULTIDROP NETWORK)
C SUBPROGRAM-11: RSMPSE (ESTIMATING RESPONSE TIME)
C
********************************************************************************

PARAMETER NW=4, IWT=100+NLIMIT2, NPC=360, NP=11A
PARAMETER NP=130, NP2=1, NP3=4, NP4=9
PARAMETER NP4=NPC/2-NP+1/4+1
PARAMETER NP7=NPC/NP=10*NPC
COMMON A(IN/SR(NP1)),N5C(NP4),NUMF(NP4),TRAFIT(NP1)

* TRAFIT(NP1)
*/VH/IVERT(NPC),IHRZN(NPC)
*/CONST/NLH2/NLH3/NLH7/MRTY
*/INF/IRATE(NP2,NP2),TRAN(NP2,NP2),FLG(NP2,NP2)
*/HCOST/ANSTC(NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2)
*/ANSTLH(NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2,NP2)
*/LINC/MINC(NP3),LINCAP(NP3),UTIL17(NP3)
*/REF/IRREF(NP3),TRAFA(NP1,NP3,NP3),PDSTNC(NP3),MAPNP(NP1)
*/VFR/VRD(NP2,NP2),I.Never1
*/NAME/NAME(NP1,NP1),NAME(NP3),NAME(NP3),NAME(NP3)
*/SUM/ASUM(NP3)
*/XMT/TIMMT(NP3),WATT(6)
*/MSLA/AMSG(7)
*/RND/NTRMS,NTRMS,MPSC,MPSC,CT
*/ADV/IDG(NP1),KCHG+KAND,OKCHG=FIRST DROP, KADD=JUST FOR LINE
INTEGER DSTNCF
DIMENSION IACT(NP4),INDPVT(NP1)
DIMENSION NUMF(NP1),TRAFITC(2),RSMSE(NP7)
DATA ITRAFC/ITRAFINTROUT/
DIMENSION TRM(WxW), ORM(WxW), NUMR(NP), NUMR(NP+4)
INTEGER SVR
DIMENSION OUTPR(NP)
NMAX=NPO MAXIMUM SIZE FOR OVERFLOW DISTANCE DATA TABLE
C
C SELECT NUMBER OF REGIONS
C
295 WRITE(6,220)
READ(5,735) NR1
WRITE(10,T1011) 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 CREACD(N3)
C
READ IN INSTALLATION AND RECURRING COSTS FOR CHARGABLE ITEMS
C REQUIRED FOR COMMUNICATION LINES
C
CALL CREADD(N4)
C
READ IN INSTALLATION AND RECURRING COSTS FOR LINES
C
CALL CREADF
C
READ IN ACTIONS TO BE PERFORMED ON EACH REGIONAL NETWORK
C
1ST ELEMENT: INSERTIONS TO PRELOADED REGIONS ARE ALLOWED
C
2ND ELEMENT: SUCH AN ACTION IS NOT ALLOWED
C
3RD ELEMENT: NETWORK OPTIMIZATION IS TO BE PERFORMED
C
4TH ELEMENT: OPTIMIZATION IS NOT NEEDED
C
CALL CREADK
C
C READ IN LINE AND LINE PROTOCOL CHARACTERISTICS
C
CALL CREADR
C
CONVERT TRAFFIC FROM CHARACTERS/MIN TO RITS/SEC
C
DO 85 K=1,2
DO 85 I=1,N1
DO 85 L=1,N7
TRAFF(I,K,L)=TRAFF(I,K,L)*P/60.
85 CONTINUE
ISUM=0
DO 25 J=1,NCTY
  ISUM=ISUM+I
  TRAFF1=TRAFF1
CONTINUE
25
DO 701 J=1,NR1
  NUMPR(J) = 0  GNO. OF SYSTEM TERMINATIONS AT EACH REGION
701 CONTINUE
26
READ(5,801) NCODE,NSTATE,NREGO
27
WRITE(6,808) NCODE, NSTATE, NREGO
28
NSTATE=LOCAL(NSTATE)  GREG CARDINAL INDEX
29
GO TO (RO1,A02,240)*NCODE
30
CONTINUE
31
READ(5,808) NCODE,NSTATE,NREGO
32
NUMPR(NREGO) = NUMPR(NREGO) + 1
33
CONTINUE
34
NSTATE=LOCAL(NSTATE)  GREG CARDINAL INDEX
35
GO TO RO5
36
CONTINUE
37
70 TRAFFON(J)=0.
38
70 CONTINUE
39
IOVER1=1  GCOUNT FOR OVERSIZED TRAFFIC DATA
40
C
41
C CALCULATE DISTANCE DATA BETWEEN SYSTEM TERMINATIONS:
42
C
43
DO 50 K=1,NCTY
  IF(J=K) 51,50,50
44
51
CONTINUE
45
ISQ1=ISQ1+(IIVERT(J)-IIVERT(K))*2
46
ISQ1=ISQ1+(1)*II ORZ(J)-II ORZ(K)*1
47
ISQ1=ISQ1/10.
48
NFOH=INT(ISQ1)
49
IffDIFF .GT. 0.) 501=NFOH+1.
50
DO 20 BDIST= SORT(SQ1)
51
KDIST=INT(KDIST)
52
DIFF=KDIST-KDIST
53
IF(DIFF .GT. 0.) KDIST=KDIST+1
54
GOTO 23
55
22
CONTINUE
56
KDIST=0
57
23
CONTINUE
58
JKL=LINK(J,K)
59
IF(KDIST .LE. 510) GOTO 5
60
CALL OVERFL(JKL,KDIST)
61
GOTO 30
62
CONTINUE
63
CALL PACK(JKL,KDIST,1,NETNCF)
64
30
CONTINUE
65
20
CONTINUE
66
IOVER1=IOVER1+1
67
68
WRITE(*,3) IOVER1
69
C
70  C TOTAL INPUT TRAFFIC BY EACH SYS. TERMIN.
TRFALL = 0.0
TALLIT = 0.0
TALLDN = 0.0
DO 41 L = 1, N1
   TRAFIT(L) = 0.0
   TRAFDN(L) = 0.0
   DO 42 J = 1, N7
      TRAFIT(L) = TRAFIT(L) + TRAFDN(L) + TRAFDL(L * 1 + J)
   CONTINUE
   TALLDN = TALLDN + TRAFDN(L)
   TALLIT = TALLIT + TRAFIT(L)
41 CONTINUE
   TRFALL = TALLDN + TALLIT
42 CONTINUE
C PRINT OUT TRAFFIC DATA BETWEEN SYSTEM TERMINATIONS
NTURN = N1/15 + 1
NREM = MOD(N1, 15)
IF (NREM .EQ. 0) NTURN = NTURN - 1
WRITE (IWT, 111)
DO 110 KK = 1, N7
   IF (KK .GT. N1) KK = KK - N1
   WRITE (IWT, 113) (INDXPT(J), J = KK, KK + N7)
110 CONTINUE
WRITE (IWT, 1014) (INDXPT(NJ), TRAFDN(NJ) + MJ = 1, N1)
WRITE (IWT, 741) TALLDN
C PRINT OUT TRAFFIC DESTINED TO EACH SYSTEM TERMINATION
WRITE (IWT, 1014) (INDXPT(NJ), TRAFIT(NJ) + NJ = 1, N1)
WRITE (IWT, 741) TALLIT
WRITE (IWT, 751) TRFALL
C PRINT OUT DISTANCE DATA BETWEEN SYSTEM TERMINATIONS
NTURN = N1/15 + 1
NREM = MOD(N1, 15)
IF (NREM .EQ. 0) NTURN = NTURN - 1
NTURN = 1
IF (NTURN .EQ. 0) NTURN = NTURN - 1
DO 101 KK = 1, N7
228  KK1=(KK-1)*15 + 1
229  KK2=KK*15
230  IF(KK2 .GE. 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 KK1=KK2, KK1, -1
235     OUTPUT('R')=DIST(J+KP)
236     27 CONTINUE
237     WRITE(IWT,112) INDXPT(J),(OUTPUT(K),K=KK2, KK1, -1)
238  99 CONTINUE
239  101 WRITE(6*210)
240  400 CONTINUE
241  READ(5*734) NSCC1
242  WRITE(IWT,1015) NSCC1
243  NSCC1=LOCAL(NSCC1)
244  IF(NSCC1.NE.0) GOTO 4003
245  WRITE(6*4013)
246  GOTO 4005
247  4003 CONTINUE
248  TPR1 = TRAFF
249  WRITE(6*2101)
250  READ(5*2102) XSAC, NREQSW, GNREAD, OF REQUESTS/TRANS AT SWITCH
251  WRITE(6*2103)
252  READ(5*2104) NTMRS, TINV, MPROC
253  WRITE(6*2105)
254  READ(5*2104) MPlot OMPLOP=1 IF PLOT IS NEEDED
255  CALL CWAITE
256  C
257  C SUM UP TOTAL TRAFFIC FOR PRELOADED SYSTEM TERMINATIONS IN REGIONS
258  C WHICH DO NOT ALLOW ANY INSERTIONS OF OTHER SYSTEM TERMINATIONS
259  C
260  TPR2=0
261  DO 77 N=1,N1
262  NK=SVR(N)
263  IF(NK .EQ. 0) GOTO 77 QNAT PRELOADED
264  IF(IACTNL(N1), F0, 0) GOTO 77 QNAT INSERTIONS ARE ALLOWED
265  TPR2=TRAFF(N)+ TRAFF(N)+ TPR?
266  77 CONTINUE
267  DO 76 L=1,NR1
268  IF(IACTNL(L), F0, 0, NR, HIMPRL(L), F0, 0) GOTO 76
269  ANR1=ANR1-1.
270  76 CONTINUE
271  TPR1=TPR1-TPR2
272  IF(NR1 .EQ. 1) GOTO 726 GONE REGION CASE
273  C
274  C DETERMINE LOWER LIMIT FOR AVERAGE REGIONAL TRAFFIC
275  C
276  ZFAT=.1
277  IF(ANR1.EQ.0) GOTO 340
278  TPR=TPR1/ANR1
279  GOTO 350
280  340 CONTINUE
285  TPRI=TPR1
286  CONTINUE
287  TPRl=TPR+(1.-ZETA)
288  DO 909 NREG=1,NRI
289  TRFS=0.
290  AMAX=0.
291  II=0
292  IF(NUMPR(NREG).NE.0) GOTO 5000 QMREG IS A PRELOADED REGION
293  C
294  C ASSIGN SYSTEM TERMINATIONS TO A REGION WITHOUT ANY PRELOADING
295  C
296  NO 400 Nl=1,Nl
297  IF(SVR(Nl) .NE. 0) GOTO 400 QMl IS PRELOADED
298  ADIST=DIST(NSCI,Nl)
299  IF(ADIST .LE. AMAX) GOTO 400
300  AMAX=ADIST QUSH DIST. FROM NSCI
301  II=II UPDATE FARTHEST SYS. TRMN.
302  CONTINUE
303  NS1=II NEW FARTHEST SYSTEM TERMINATION
304  TRFS=TRFS + TRAFN(NS1) + TRAFT(NS1)
305  SVR(NS1)=NREG
306  NUMPR(NREG)=NUMPR(NREG)+1
307  IF(TRFS .GT. TPRl) GOTO 707
308  GOTO 7021
309  5000 CONTINUE
310  IF(IACT(NREG,.1) .EQ. 1) GOTO 909 QINSERTIONS ARE NOT ALLOWED
311  C
312  C SUM UP TRAFFIC IN THIS REGION
313  C
314  NO 702 II=1,Nl
315  IF(SVR(I) .NE. NREG) GOTO 702
316  TRFS=TRFS+TRAF(NI)+TRAFT(I)
317  ADIST=DIST(NSCI,Nl)
318  IF(ADIST .GT. AMAX) II=I
319  CONTINUE
320  IF(TRFS .GT. TPRl) GOTO 707 SUSH TRAFFIC IN THIS REGION
321  NS1=II NEW FARTHEST SYS. TRMN. IN THE REGION
322  IF(NPSC(NREG) .NE. 0) NS1=NSC(NREG)
323  CONTINUE
324  CALL FINDD(NS1,N52)
325  IF(N52 .NE. 0) GOTO 909
326  SVR(NS2)=NREG
327  NUMPR(NREG)=NUMPR(NREG)+1
328  TRFS=TRFS+TRAF(NS1)+TRAFT(NS2)
329  IF(NREG .EQ. NRI) GOTO 7021
330  IF(TRFS .GT. TPRl) GOTO 707
331  GOTO 7021
332  707 CONTINUE
333  TPRI=TPRI-TRFS QUPDATE REMAINING TRAFFIC
334  ANRI=ANRI-1
335  TPRI=TPRI/ANRI QUPDATE AVERAGE TRAFFIC PFF REGION
336  TPRL=TPR+(1.-ZETA) QUPairate LOWER LIMIT
337  CONTINUE
338  GOTO 703
339  726 CONTINUE
340  C
341  C ONE REGION CASE
C DO 727 NN=1,NN1
344 SVR(NN) = 1
345 727 CONTINUE
346 NUMBR(1) = NN1
347 703 CONTINUE
348 C SELECT REGIONAL SWITCHING CENTER
349 350 C PRINT OUT PID AND NAMES FOR SYSTEM TERMINATIONS IN THE REGION j
351 C WRITE(IWT+101A) J,(NUMBR(I),I=1,NN1,NUMBR)
352 C PRINT OUT INDICES OF SYSTEM TERMINATIONS IN THE REGION j
353 C IF(NRSC(J) .NE. 0) GO TO 501
354 N1 = NUMR(K) ASSUMED RSC
355 SUMT = 0.0
356 DO 530 L=1,NN2
357 SUMT = SUMT + (TRAFFN(NN1)L)*DSTT(NN2,J)
358 530 CONTINUE
359 IF(SUMT .LT. WCASE) GO TO 520
360 WCASE = SUMT
361 NRSC(J) = NN1
362 520 CONTINUE
363 501 CONTINUE
364 500 CONTINUE
365 NN4=NRSC(J)
366 WRITE(IWT+1003) INDEXT(NN4),J
367 IF(JACTH(J,.2) .LT. 1) OPTIMIZATION IS REQUIRED
368 CALL RNMET(J,NUMR,NUMR1,GO,NUMR2)
369 500 CONTINUE
370 C GENERATE INTER-REGION ORIGIN-DESTINATION MATRIX
371 C INITIALIZATION
372 C IF(NN1 .LE. 2) GOTO 551
373 DO 902 K1=1,NN7
374 KKK=NBASE(K1)
375 KKK=LOCAL(KKK)
399 IF (KKK, EQ. 0) WRITE (6, 7777) K1
400 NBASF(K1) = NRV(KKK)
401 902 CONTINUE
402 DO 609 K1=1, NR1
403 DO 609 K2=1, NR1
404 TRM(K1, K2) = 0.
405 TRM(K2, K1) = 0.
406 699 CONTINUE
407 DO 900 J=1, NR1
408 NMAP = 0
409 DO 900 K1=1, NR1
410 IF (SRV(K1) .NE. J) GO TO 905
411 400 N2=NBASE(KK) OREGONAL INDEX FOR KK'S DATA PAGE
412 TRM(J, K2) = TRAFN(K1, K2) + TRM(J, NR1) QUITING TRAFFIC
413 TRM(N2+J) = TRAFN(K1, K2) + TRM(N2+J) COMING TRAFFIC
414 915 CONTINUE
415 905 CONTINUE
416 DO 920 J=1, NR1
417 NNP = NRSY(J)
418 DNK(J, J+1) = DIST(NR1, NR1+1)
419 920 CONTINUE
420 900 CONTINUE
421 NTURN=NR1/10+1
422 DO 535 L=1, NTURN
423 LL=L*10 + 1
424 LU=L*10
425 IF (L, GT, NR1) LU=NR1
426 WRITE(1030, NR1+NR1, K, K=LL, LL+1)
427 DO 1022 I=1, NR1
428 WRITE(10, NR1+1) I, TRM(I, J), J=LL, LL+1)
429 1022 CONTINUE
430 535 CONTINUE
431 DO 545 L=1, NTURN
432 LL=L*10 + 1
433 LU=L*10
434 IF (L, GT, NR1) LU=NR1
435 WRITE(1030, NR1+NR1, K, K=LL, LL+1)
436 DO 1024 I=1, NR1
437 WRITE(I, NR1+1) I, TRM(I, J), J=LL, LL+1)
438 1024 CONTINUE
439 545 CONTINUE
440 CALL IRNOPS(NR1, NLIMIT, TRM)
441 74 FORMAT(//40X, ' TOTAL TRAFFIC=+F4.2)
442 75 FORMAT(//35X, ' TOTAL SYSTEM TRAFFIC=+F4.2)
443 220 FORMAT(20, 'ASSUM NUMBER OF REGIONS=')
444 * / ENTER NR AND STRIKE RETURN KEY * 
445 735 FORMAT(I3)
446 88A FORMAT(I1, ' TYPE IN PRELOADED SYSTEM TRAFF, AND SC WITH, '
447 800 FORMAT(11, '.TYPE IN PRELOADED SYSTEM TRAFF, AND SC WITH, '
448 804 FORMAT(I10, ' DISTANCE ITEMS ARE OVERS, PEN')
450 804 FORMAT(10X, 'ENTER CONF FOR NSY AND STRIKE RETURN KEY')
451 3 FORMAT(I1, ' ENTER CONF FOR NSY AND STRIKE RETURN KEY')
452 210 FORMAT(' ASSUME A SYSTEM CENTROID, ')
453 734 FORMAT(A4)
455 4013 FORMAT(' THE GIVEN SYSTEM COMM. CENTROID IS NOT OK, RETYPE IT')
77-53, Vol. IV

456 2101 FORMAT('1X,'10I20) * INPUT TOTAL NO. OF TRANSACTIONS AND NO. OF ACCESS ** 
457       1 * AT THE SWITCHER */1X,'ENTER WITH FA. 85 AND 13 UNDER X$$R/SEC')
458 2102 FORMAT('F6.5','13I3')
459 2103 FORMAT('1X,'READ IN LIMITS ON NO. OF SYS. TERMS. ON A INF**1/ 
460       1 *RESPONSE TIME REGN AND NO. OF PROCESSES WITH FORMAT '** 
461       2 /'F5.2','12I1')
462 2104 FORMAT('F5.2','12I2')
463 2105 FORMAT('1X,'IF PLOTTING IS REQUIRED: TYPE 1 WITH FORMAT 1X')
464 110 FORMAT('(*1X,A6,2X,15F8.1)')
465 111 FORMAT('1X,'TRAFFIC MATRIX (PPS) *')
466 113 FORMAT('1X,'TERM.* ')15X(4X,A4)')
467 109 FORMAT('140X,'POINT TO POINT DISTANCE MATRIX ',
468       * /'1X,'TERM.*15(4X,A4), /')
469 112 FORMAT('2X,'RESPONSE TIME REGN AND NO. OF PREFERENCES WITH FORMAT *')
470 101 FORMAT('1X,'TERM.*15(4X,A4), /')
471 100 FORMAT('1X,'TOTAL TRAFFIC ORIGINATED FROM SYS. TERMINI, ** 
472       10X,'TERMS/SEC)
473 101 FORMAT('1X,'TOTAL TRAFFIC ORIGINATED FROM SYS. TERMINI, ** 
474       10X,'TERMS/SEC)+')
475 104 FORMAT('15X,'TOTAL TRAFFIC DESTINATION TO SYS. TERMINI, ** 
476       * /'15X,'TERMS/SEC)+')
477 105 FORMAT('10X,'NCC= ',2X,A4, /')
478 104 FORMAT('1X,'REGION= 15')
479 777 FORMAT('1X,'TERM.*13,'TH DATA RACE IS NOT RUN AS A SYS. TERM.*')
480 102 FORMAT('1X,'(15X,'TERM.*10(3F10.3), /')
481 102 FORMAT('1X,'INDICES FOR SYS. TERMINI ,15(3F10.3))
482 103 FORMAT('1X,'INITIAL INTERREGION TRAFFIC MATRIX (*)
483       * /'12X,'X12, /'//,21X,10(5X,A1))
484 103 FORMAT('1X,'INTERREGION DISTANCE MATRIX (12,12) //,21X,10(5X,A1))
485       * /'MILF= ,//,21X,10(5X,A1)
486 551 CONTINUE
487 STOP
488 C SUBROUTINE FINDM(N,M)
489 C ***************
490 C
491 C FIND THE NEXT SYSTEM TERMINATION M WHICH IS CLOSEST TO N
492 C WHERE M HAS NOT BEEN ASSIGNED TO ANY REGION YET
493 C
494 C ***************
495 AMIN=20000
496 M=0
497 DO 70A K=1,N1
498 IF(SW(IK).NE.0) GOTO 70A
499 ADIST=DIST(N,K)
500 IF(ADIST .GE. AMIN) GOTO 70B
501 AMIN= ADIST
502 R=K
503 70A CONTINUE
504 RETURN
505 C SUBROUTINE CREADK
506 C ***************
507 C READ IN ACTIONS REGARDING INSPECTIONS OF SYSTEM TERMINATIONS
508 C TO PRELOADED REGIONS AND REGIONAL NETWORK OPTIMIZATIONS
509 C ***************
510 WRITE(6,94)
511 94 FORMAT(' TYPE IN ACTION INDICES FOR EACH REGION *')
512 * /'1ST ELEMENT: 1= INSPECTION TO THIS PRELOADED REGION IE OK *
77-53, Vol. IV

513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  

**FUNCTION OF THIS SUBROUTINE ARE TO**

1. RECEIVE TOTAL NO. OF SYSTEM TERMINATIONS, DATA BASES AND CITIES

2. RECEIVE CITY LOCATIONS (V A H)

3. RECEIVE PID NO., SYS. TERM, NAMES, AND MAPPING AND TRAFFICS

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

C

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

C

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

C

1 FORMAT(6,14): TYPE IN NO. OF SYS. TERMINATIONS, DATA BASES AND CITIES:

* 'WITH FORMAT 315'*

READ(5,10) NINTCITY A NUMBER OF SYSTEM TERMINATIONS

WRITE(6,14) NINTCITY

78 FORMAT(' THERE ARE '15', SYS. TERMINATIONS, '14', DATA BASES, '15',

CITIES', ' TYPE IN DATA BASE LOCATIONS WITH FORMAT 6(1Y,A4)*)

READ(5,15) (NAMEFIT(I),I=1,N7)

15 FORMAT(6,14) NAMEFIT I=1,N7

16 FORMAT(15) DATA BASES ARE AT: 6(2Y,A4)

WRITE(6,16)

161 FORMAT(' TYPE IN CITY V-H WITH FORMAT (13Y,15,2X,15)*)

READ(5,17) (VHRCITY(I),I=1,N7)

17 FORMAT(13Y,15,2X,15)

WRITE(6,17)

76 FORMAT(' TYPE IN PID NO., NAME, MAPPING AND TRAFFIC*/ WITH FORMAT 14,X,14,4F10.2)*)

DO 79 1=1,N1

70 FORMAT((NAMEFIT(I)+J=1,N),TANDFIT(I),MAPFIT(I),(TRAFFIC(I,K,L))

* K=1,2) I=1,N7

74 INDEXFIT(I)=I

79 CONTINUE

40 FORMAT((A4,1X,34G,14,12,14,4F10.2))

10 FORMAT(315)

RETURN

SUBROUTINE CFADRT(N2)
C CREATE A RATE APPLICATION MATRIX IRATEJ(N2,N2)

WRITE(6,83)

83 FORMAT(* TYPE IN NO. OF RATE STRUCTURE UNDER**)

84 FORMAT(* TYPE IN RATE APPLICATION TO EACH COMM. **)

85 FORMAT(10I2)

86 FORMAT(I15)

87 FORMAT(I15)

88 FORMAT(I15)

89 FORMAT(I15)

90 FORMAT(I15)

91 FORMAT(I15)

92 FORMAT(I15)

93 FORMAT(I15)

94 FORMAT(I15)

95 FORMAT(I15)

96 FORMAT(I15)

97 FORMAT(I15)

98 FORMAT(I15)

99 FORMAT(I15)

100 FORMAT(I15)

101 FORMAT(I15)

102 FORMAT(I15)

103 FORMAT(I15)

104 FORMAT(I15)

105 FORMAT(I15)

106 FORMAT(I15)

107 FORMAT(I15)

108 FORMAT(I15)

109 FORMAT(I15)

110 FORMAT(I15)

111 FORMAT(I15)

112 FORMAT(I15)

113 FORMAT(I15)

114 FORMAT(I15)

115 FORMAT(I15)

116 FORMAT(I15)

117 FORMAT(I15)

118 FORMAT(I15)

119 FORMAT(I15)

120 FORMAT(I15)

121 FORMAT(I15)

122 FORMAT(I15)

123 FORMAT(I15)

124 FORMAT(I15)

125 FORMAT(I15)

126 FORMAT(I15)

127 FORMAT(I15)

128 FORMAT(I15)

129 FORMAT(I15)

130 FORMAT(I15)

131 FORMAT(I15)

132 FORMAT(I15)

133 FORMAT(I15)

134 FORMAT(I15)

135 FORMAT(I15)

136 FORMAT(I15)

137 FORMAT(I15)

138 FORMAT(I15)

139 FORMAT(I15)

140 FORMAT(I15)

141 FORMAT(I15)

142 FORMAT(I15)

143 FORMAT(I15)

144 FORMAT(I15)

145 FORMAT(I15)

146 FORMAT(I15)

147 FORMAT(I15)

148 FORMAT(I15)

149 FORMAT(I15)

150 FORMAT(I15)

151 FORMAT(I15)

152 FORMAT(I15)

153 FORMAT(I15)

154 FORMAT(I15)

155 FORMAT(I15)

156 FORMAT(I15)

157 FORMAT(I15)

158 FORMAT(I15)

159 FORMAT(I15)

160 FORMAT(I15)

161 FORMAT(I15)

162 FORMAT(I15)

163 FORMAT(I15)

164 FORMAT(I15)

165 FORMAT(I15)

166 FORMAT(I15)

167 FORMAT(I15)

168 FORMAT(I15)

169 FORMAT(I15)

170 FORMAT(I15)

171 FORMAT(I15)

172 FORMAT(I15)

173 FORMAT(I15)

174 FORMAT(I15)

175 FORMAT(I15)

176 FORMAT(I15)

177 FORMAT(I15)

178 FORMAT(I15)

179 FORMAT(I15)

180 FORMAT(I15)

181 FORMAT(I15)

182 FORMAT(I15)

183 FORMAT(I15)

184 FORMAT(I15)

185 FORMAT(I15)

186 FORMAT(I15)

187 FORMAT(I15)

188 FORMAT(I15)

189 FORMAT(I15)

190 FORMAT(I15)

191 FORMAT(I15)

192 FORMAT(I15)

193 FORMAT(I15)

194 FORMAT(I15)

195 FORMAT(I15)

196 FORMAT(I15)

197 FORMAT(I15)

198 FORMAT(I15)

199 FORMAT(I15)

200 FORMAT(I15)

201 FORMAT(I15)

202 FORMAT(I15)

203 FORMAT(I15)

204 FORMAT(I15)

205 FORMAT(I15)

206 FORMAT(I15)

207 FORMAT(I15)

208 FORMAT(I15)

209 FORMAT(I15)

210 FORMAT(I15)

211 FORMAT(I15)

212 FORMAT(I15)

213 FORMAT(I15)

214 FORMAT(I15)

215 FORMAT(I15)

216 FORMAT(I15)

217 FORMAT(I15)

218 FORMAT(I15)

219 FORMAT(I15)

220 FORMAT(I15)

221 FORMAT(I15)

222 FORMAT(I15)

223 FORMAT(I15)

224 FORMAT(I15)

225 FORMAT(I15)

226 FORMAT(I15)
DO 13 IRATE=1,N2
DO 13 ILINE=1,N3
DO 13 IDNSTY=1,N4
DO 13 IDNS=1,N5
READ(5,100)((AINST(I),IRATE,ILINE,IDNSTY,T,Y,I,J,K,N,N1,J1,J2),J=1,N)
READ (5,101)((RECLN(I),IRATE,ILINE,IDNSTY,T,Y,I,J,K,N,N1,J1,J2),J=1,N)
13 CONTINUE
100 FORMAT (2F9.2/2F9.2)
50 FORMAT (13/10(A6,1X))
RETURN
SUBROUTINE CREATE

C
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 (tü TYPE IN INST. COSTS FOR LINES W/T *
1 /** RATE, LINE, DENSITY, AND MPP/2 MiND_FORMAT **)
2 /** WITH FORMAT 4F9.2**
WRITE(6,90)
90 FORMAT (tü TYPE IN INDEX FOR LINEARITY OF LINE RECUR. COST;
1 /** FUNCTION WITH 1=LINEAR AND NONLINEAR OTHERWISE**;
2 /** WITH FORMAT 11 FOR EACH LINE TYPE**)
WRITE(6,91)
91 FORMAT (tü TYPE IN RECUR. COSTS WITH FORMAT 4F9.2 IF LINEAR **;
1 /** WITH FORMAT 10F9.2/10F9.2** IF NONLINEAR**;
2 /** IF NONLINEAR; USE 10F9.2**)
DO 14 IRATE=1,N2
DO 14 ILINE=1,N3
READ(5,200) INDEX
IFLAG(IRATE,ILINE)=INDEX (LINE COST LINEARITY INDICATOR)
DO 14 IDNSTY=1,N3
READ (5,100) ((ANST(I),IRATE,ILINE,IDNSTY,T,Y,I,J,K,N,N1,J1,J2),J=1,N)
IF (INDEX,NF,1) GO TO 3
C
LINEAR COST FUNCTION
READ (5,100) ((RECLN(I),IRATE,ILINE,IDNSTY,T,Y,I,J,K,N,N1,J1,J2),J=1,N)
GO TO 14
CONTINUE 3
NONLINEAR COST FUNCTION
READ (5,400) ((RECLN(I),IRATE,ILINE,IDNSTY,T,Y,I,J,K,N,N1,J1,J2),J=1,N)
CONTINUE 14
100 FORMAT (L4F9.2)
200 FORMAT (11)
401 FORMAT (10F9.3/10F9.3)
RETURN
FUNCTION LOCAL(NL)
C
C FIND LOCAL INDX FOR SYSTEM TERMINATION WITH YD NL
C
C
LOCAL=0
IF(NL,NE,0) RETURN
DO 400 N=1,N1
IF(INDEX,NL,NE) GO TO 400
CONTINUE 400
**77-53, Vol. IV**

684 RETURN
685 4002 LOCAL=NN
686 RETURN
687 SUBROUTINE OVERFL(J,K)
688 C **************
689 C
690 C STORE OVERFLOW ELEMENT (J,K) AT LOCATION J+K OF TABLE
691 C LA AND PUT A MARK 511 AT LOCATION J AT LOCATION J+K OF TABLE (DISTANCE)
692 C
693 C **************
694 IF(IOVERF(J,K) GE. NMAX) GOTO P000
695 CALL PACK(J,511,DISTANCE)
696 IVRO(IOVERF(J,K))=J
697 IVRO(IOVERF(J,K))=K
698 I0VERF=I0VERF+1
699 RETURN
700 P000 CONTINUE
701 WRITE(*,8001)
702 8001 FORMAT(2X,'THE OVERFLOW TABLE HAS REACHED ITS MAXIMUM SIZE; PLEASE INCREASE ITS SIZE')
703 STOP
704 C SUBROUTINE CREADR
705 C **************
706 C
707 C RECEIVE DATA FOR RESPONSE TIME CALCULATION
708 C
709 C NPLIN(NP9+2)=INPUT MSG LENGTH AS A FUNCTION OF TYPE AND PRIORITY.
710 C NMSOUT(NP9+2)=OUTPUT MSG LENGTH AS A FUNCTION OF TYPE AND PRIORITY.
711 C AMSL(J)= AVERAGE MSG LENGTH FOR
712 C 1=POLLING 2=NAK RESPONSE 3=INPUT MSG WITH PRIORITY 1
713 C 4=INPUT MSG 5=OUTPUT MSG WITH PRIORITY 1
714 C 6=OUTPUT MSGS WITH PRIORITY 2 7=ALL MSGS
715 C TIMXMT(J7)=AVERAGE TRANSMISSION TIME FOR ABOVE ITEMS.
716 C RATPR1(NP9+2)=OUTPUT MSG DISTRIBUTION AND OUTGOING MSG RATIO BY PRIORITY
717 C N+1,1 = PERCENT OF OUTPUT MSG SENT WITH PRIORITY 1 IF ITS TYPE IS N
718 C N+1,2 = PERCENT OF OUTPUT MSG WHOSE DESTINATION IS OUTSIDE OF
719 C N+1,3 = PERCENT OF OUTPUT MSG WHOSE PRIORITY IS A FUNCTION OF TYPE AND PRIORITY
720 C N+1,4 = INPUT TRAFFIC DISTRIBUTION AS A FUNCTION OF TYPE AND PRIORITY
721 C N+1,5 = OUTPUT TRAFFIC DISTRIBUTION AS A FUNCTION OF TYPE AND PRIORITY.
722 C
723 C **************
724 C
725 C DIMENSION MSLIN(NP9+2),RATIO(NP9+2),RATIO(TYPE,NP9+2)
726 C 1: NMLIN(NP9+2)=MSG LENGTH FOR
727 C 2: NPL(NP9+3)=NAK(NP9+3) + PLOH(NP9+3) + NAKOH(NP9+3)
728 C 3: MOH(NP9+3), TAMD(NP9+3), TAPP(NP9+3)
729 C
730 C WRITE(*,771)
731 771 FORMAT(1X,TYPE IN NPL, NAK, NPLAH, NAKOH, MOH,**
732 C * /+ TAMD, TAD IN FORMAT (5I4,2F7.5)**
733 C READ(5,771) (NLPL(I),NAK(I)+PLOH(I)+NAKOH(I))
734 C * MOH(I), TAMD(I), TAPP(I), I=1,N3)
735 C WRITE(1,733) (LINCP(I)+NPL(I)+NAK(I)+PLOH(I)+NAKOH(I))
736 C * MOH(I), TAMD(I), TAPP(I), I=1,N3)
737 C FORMAT(1X,TYPE IN NPL, NAK, NPLAH, NAKOH, MOH,**
738 C * /+ TAMD, TAD IN FORMAT (5I4,2F7.5)**
739 C WRITE(*,772)
740 772 FORMAT(1X,TYPE IN NO. OF MSG TYPES, AND TRAFFIC STATISTICS**

---

A-13
1 /\* SUCH AS MSGNAME, MSGLIN, MSGOUT, RATIO WITH */
2 /\* FORMAT (I4,2(I6,2(2I4,F6.3))) */
3 READ(5,77) NTYP
4 READ(5,174) (MSGNAME(I),MSGLIN(I),MSGOUT(I),RATIO(I,J),I=1,N)
5 
6 179 FORMAT((A6+2(2I4,F6.3)))
7 READ(5,81) CPHAVG
8 
9 IF(RATIO(I,J)/RATIO(I,I,J>=1)) = 1.0) RETURN
10 
11 C CALCULATE AVERAGE MSG LENGTH
12 C
13 DO 61 I=3,N
14
15 ASML(I)=0.
16 
17 61 CONTINUE
18 DO 50 I=1,N
19
20 ASML(I)=2.
21 
22 50 CONTINUE
23 DO 62 I=1,NTYP
24
25 ASML(I)=AMSL(I)/RATIO(I,I,J)
26
27 62 CONTINUE
28 
29 ASML(I)=AMSL(I)/ASML(I)
30 
31 67 CONTINUE
32 IF(ASUM(I) .GT. 0.0) GOTO 6A
33 
34 ASML(I)=AMSL(I)/ASUM(I)
35 
36 68 CONTINUE
37 
38 ASML(I)=AMSL(I)/ASUM(I)
39 
40 IF(I+ASML(I) .GT. 0.5) GOTO 6A
41 ASML(I)=AMSL(I)/ASUM(I)
42 
43 69 CONTINUE
44 
45 WRITE(10,106) ASML(I),I=3,N
46 
47 106 FORMAT(/5X,10I2,A3,*15I1)
48 
49 105 FORMAT(/5X*AVERAGE MSG LENGTH FOR PRIORI
50 
51 104 FORMAT(/5X,AVERAGE MSG LENGTH FOR PRIORI
52 
53 103 FORMAT(/5X,AVERAGE MSG LENGTH FOR PRIORI
54 
55 102 FORMAT(/5X,AVERAGE MSG LENGTH FOR PRIORI
56 
57 101 FORMAT(/5X,AVERAGE MSG LENGTH FOR PRIORI
58 
59 100 FORMAT(/5X,AVERAGE MSG LENGTH FOR PRIORI
60 
61 99 FORMAT(/5X,AVERAGE MSG LENGTH FOR PRIORI
62 
63 98 FORMAT(/5X,AVERAGE MSG LENGTH FOR PRIORI
64 
64 DO 65 K=1,N3
65 
66 AMSL(K)=NPL(K)
67 
68 65 CONTINUE
69 
70 RETURN
71
SUBROUTINE CWAIT

**--------------------------------**

C  PPF=CALCULATE CPU WAIT TIME

C  **--------------------------------**

C  RHOCPU=XSAC+CPUAVG/MPROC

WRITE(100,850) RHOCPU

850 FORMAT(* CPU UTILIZATION FOR PROCESSOR IS F5.3)

IF(RHOCPU .LE. .A) GOTO 851

808 WRITE(6,855)

855 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.*RHN2

+64.*RH02+RHOCPU+248.*RH04)

820 700 CONTINUE

821 WAIT(4)=CPUAVG*(BETA/(MPROC*(1.-RHOCPU)))

822 WAIT(4)=WAIT(4)*NREQSW

823 RETURN

END
77-53, Vol. IV

S192E#STACOM(1), RGNNET/1777
1  SUBROUTINE RGNNET(JREGN, NREGN, NMAP, 160, NUMRR)
2
3  *********************
4  C
5  C DEVELOP A REGIONAL MULTIDROP NETWORK, STARTING WITH A STAR
6  C NETWORK AND THEN OPTIMIZE IT BY ESAIL-WILLIAMS METHOD, GIVEN
7  C THE FOLLOWING ARGUMENTS:
8  C
9  C JREGN= THE INDEX FOR THE REGION UNDER CONSIDERATION
10  C NREGN= THE NUMBER OF SYSTEM TERMINATIONS IN REGION JREGN
11  C NUM= AN ARRAY THAT CONTAINS INDICES FOR ALL SYSTEM
12  C TERMINATIONS IN REGION JREGN
13  C IGO= 1 IF NETWORK OPTIMIZATION IS TO BE PERFORMED
14  C NOTE: NODE AND SYSTEM TERMINATION ARE EXCHANGEABLE
15  C
16  *********************
17  C
18  PARAMETER NP1=130, NP2=1, NP3=4, NP4=3
19  PARAMETER NP7=4
20  PARAMETER NPt=100, MW=4, NPC=60
21  PARAMETER NP6=(NPC+NPC/2-NPC+1)/4+1
22  COMMON/ CONST/N1HP1+NXN4+7, LTCITY
23  * REF/ IREF(NP1), TRFN(NP1:2:NP7), PSTNCF(NP6), MAPAP(NP1)
24  * LINCR/ LTMX(NP3), LTNCAP(NP3), UTIL(NP3)
25  * INF/ IRATF(NP2:NP2), IRAN(NP2:NP2), IFLG(NP2:NP3)
26  * ETR/ SVR(NP1), RSC(MW), NUMPP(MW), TRAF(NP1), TRAFIT(NP1)
27  * NAME/NAMET(NP1:4), LNAME(NP3), NAME(NP4)
28  * SUM/ ASGIN(4), ARUM
29  * /MSPA/AMS/7)
30  /ROUN/ NTERMS,TIME0,MPROCMPL
31  /ADD/ IADD(NP1),KCHO,KAND
32  /REP/ RHO(N1:4),SPTM
33  DIMENSION COSTEW(NP1:4),TAPRAY(NP1:5),ARRAY(NP1:2)
34  DIMENSION TIME(NP1:TFSUM(NP1:1),TIMM(NP1)
35  DIMENSION HINES(NP1:NP3), LDUMY(NP3), NMAP(1)
36  DIMENSION ICST(NP1), LWORK(NP1:NP2+1), ICST(NP1:2), ICTOST(NP1:2)
37  DIMENSION LSUB(NP1), MSUB(NP1), NUMR(NP1)
38  DIMENSION IBLANK(NP1:1), JCHAR(2)
39  DIMENSION NUMRA(1)
40  DIMENSION RHO(NP1)
41  EQUIVALENCE (JCHAR,ICAR)
42  DATA JRLANK/, '/
43  RSPTIM=0.
44  IPOINT=
45  INTEGR TCST1, TCST2, COST, COSTF
46  C
47  C INITIALIZE COST ARRAY, INDEPENDENT OF INF TYPE
48  C
49  DO 399 K1=1,N1
50  DO 399 K2=1,2
51  ICSTLN(K1,K3)=0
52  DO 399 K4=1,N4
53  ICSTHW(K1,K4,K3)=0
54  CONTINUE
55  NN1=RSC(JREGN) GLOBAL INDEX FOR RSC
56  C

A-16
C FInd the local RSC index in the region array

DO 90 INI=1,NOREGN
   IF(INI .EQ. NUMR(INI)) GOTO 199
  90 CONTINUE

C BUILD A STAR NETWORK

CALL STAREW

C PRINT OUT STAR NETWORK

CALL SUMPRT(NOREGN,1)

C DEVELOP A MULTIDROP NETWORK UTILIZING THE

C ESAU-WILLIAMS ALGORITHM

MAXSAV=0
MAXM=0
MAXL=0
MAXK=0
MAXKI=0
MAXLIN=0
MAXNOL=0
LINNEW=0
RSPMAX=0.
RHOMAX=0.
ICHAG=1
ITALLY=0
JTALLY=0
KCHB=1
KADO=1
IKK=0
INTTRY=0
CALL E'S5WIL
WRITE(6,933) ITALLY, JTALLY
933 FORMAT(1X,'TRYLNK HAS BEEN ACCEPTED FOR ',12X,' TIMES')
!
GOTO 979

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

C COST, COSTFW

C NOREGN=NUMBER OF SYSTEM TERMINATIONS IN THE REGION

C

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

C

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

DO 109 K3=1, NOREGN
  109 CONTINUE

K3 = NUMR(K3)
114   IARRAY(K3,1)=IAND(KK)
115   IARRAY(K5,5)=IRSC   @ LOCAL INDEX FOR RSC
116   ARRAY(K3,1)=TRAFIN(KK)
117   ARRAY(K3,2)=TRAFIT(KK)
118   TIMESP(K3)=0
119   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 .EQ. 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 KS=1, NOREGN
129      IF(KS .EQ. IRSC) GOTO 200
130      NM=NM + 1
131      IF(NM .EQ. IRSC) NM=NM + 1
132      IARRAY(IRSC+2)=NM
133   200   CONTINUE
134   C
135   C RELATE ALL OF RSC'S SUCCESSORS
136   C
137   DO 550 NODE=1, NOREGN
138      IF(NODE .EQ. IRSC) GOTO 555
139      TRFIN=ARRAY(NODE+1)+0.5
140      TRFOUT=ARRAY(NODE+2)+0.5
141      NN2=NUMR(NODE)
142      DIST=TRFIN+TRFOUT+LDIMMY(LINOL(NODE)+0)
143      COSTW(NODE+4)=DIST
144   550   CONTINUE
145   C
146   C TAKE A FIRST GUESS FOR LINE CONFIGURATION
147   C
148   COST=0
149   RHO=0.
150   MDROP=IARRAY(NODE+1)+1
151   CALL LINNUM(TRFIN,TRFOUT,LDIMMY,LINOL,N,RHO)
152   781   CONTINUE
153   C
154   C COMPUTE INITIAL RESPONSE TIME
155   C
156   IKONT=0
157   DO 783 I=1, N3
158      IF(LINUMM(I) .NE. 0) IKONT=IKONT+1
159      CONTINUE
160   783   CONTINUE
161   A1TRF=TRFIN
162   OUTTRF=TRFOUT
163      IF(IKONT .EQ. 1 ) AND (LDIMMY(LINOL) .EQ. 1) GOTO 772
164   C
165   C RESPONSE TIME CALCULATION NEEDS MODIFICATION
166   C
167   A1CAP=0.
168   DO 771 NL=1, N3
169      A1CAP=A1CAP+LINCM(NL)+LDIMMY(NL)
77-53, Vol. IV

```fortran
171 L dummy(NL) = 0
172 Continu
173 L dummy(L inol) = 1
174 A intr = a trfin + lin cap(L inol) / acap opfrc + T traffi
175 O uttr = a trf out + lin cap(L inol) / acap
176 Continu
177 Call rspnse(a intr, a uttr, lin ol, maxp, iok)
178 IF(iok .eq. 1) Goto 773
179 IF(lin ol .eq. N3) Goto 774
180 L dummy(L inol) = 0
181 Lin ol = lin ol + 1
182 L dummy(L inol) = 1
183 Goto 775
184 Continu
185 NLL = 0
186 N33 = N3 - 1
187 Do 776 I = 1, N33
188 IF(L dummy(I) .eq. 0) Goto 776
189 NLL = I
190 Goto 780
191 Continu
192 L dummy(I) = 1
193 Goto 775
194 Continu
195 L dummy(N LL) = 0
196 L dummy(N LL + 1) = L dummy(N LL + 1) + 1
197 Continu
198 Call rhofun(trfin, trfout, l dummy, lin ol, pholn, rho)
199 Goto 781
200 Continu
201 T imesp(node) = r spot im
202 K chr = 2
203 Call isimp(ip, rsc, nrod, C, cost)
204 R Hod(nodf) = rho
205 Co stw(node) = C
206 Co stf(node, 2) = lin ol
207 Do 499 NL = 1, N3
208 Nlines(node, NL) = L dummy(NL)
209 Do 499 NM = 1, 2
210 Ics tln(node, NM) = Ics tln(node, NM) + nkln(NL + NM)
211 Do 499 NK = 1, N4
212 Ics thw(node, NK + NM) = Ics thw(node, NK + NM) + tcstw(nodf, NK, NM)
213 Continu
214 Jtraf = trfin + trfout
215 Jtraf = jtraf / ut iliz(l inol)
216 Co stf(node, 3) = jtraf / lin cap(l inol) + 1
217 Goto 555
218 Continu
219 Continu
220 Call assuming traffic at irsc is taken care of automatically
221 Continu
222 Do 498 NL = 1, N3
223 Nlines(node, NL) = 0
224 Continu
225 Co stf(node) = 0
226 Co stf(node, 2) = 0
227 Co stf(node, 3) = 0
```

A-19
COSTFWINODE=4) = 0
RHOF(NODE)=0.

550 CONTINUE
RETURN

SUBROUTINE ISUMUP(L1,L2,LT,IC)

C
**************
C CALCULATE COST BETWEEN NODES L1 AND L2 AND ADD IT TO
C TOTAL COST IC WHERE LT=LINE TYPE
C
C
**************
LL1=NUMR(L1)
LL2=NUMR(L2)
CALL I*CO5TJ(NUMM,LL1,LL2,LNKCLN,LKLN)
KK=3
IF(LL1.NE.0) KK=1
DO 211 LINTYP=1,KK
LTYP=LINTYP
IF(KK.EQ.1) LTYP=LT
DO 221 I1=1,2
IC=IC+LNKCLN(LTYP,I1)
DO 222 I2=1,N4
IC=IC+LNKCLN(LTYP,I2,I1)
222 CONTINUE
211 CONTINUE
211 CONTINUE
RETURN
SUBROUTINE ESSWIL

C
**************
C TRY AGAIN TO OPTIMIZE THE NETWORK
C
C
**************
5000 CONTINUE
K=IARRAY(RSC+1) QFIRST SUBNETWORK UNDER RSC
KNEXT=IARRAY(K+3) QNEXT SUBNETWORK UNDER RSC
IF(KNEXT .NE. 0) GOTO 599 QONLY ONE SUBNETWORK
560 CONTINUE
IKK=0
L=IARRAY(RSC+2) QSUBNET IS TO PF LINKED TO L=SUBNET
570 CONTINUE
IF(L .NE. K) GOTO 575
L=IARRAY(L+3)
IF(LF0+0) GOTO 660
575 CONTINUE
K1=NUMR(K)
DIFF=DIST(N1,K1)
C
C FIT TOTAL NO. OF TERMINALS IF K AND L ARE COMBINED
C
IMTRY=QINDICATION OF ENTRY TO TRYLINK
LINE=COSTF(K+2)
IF(LINCAP(LINF) .GT. 9600) GO TO 585 QNO MULTIDROPPING ON 9600
NODE=IARRAY(K+1)+IARRAY(L+1)+2
IF(NODE .GT. NTERMS) GOTO 585 QTOO MANY TERMINALS
M=L
K=K

A-20
580 CONTINUE
581 M=MMR(M)
582 DTRY=DIST(KI+M)/2.
583 IF(DTRY GT. DREF) GOTO 140
584 CALL TPLNK(K+KI+L/M) Q M IS THE INSERTION NODE.
585 IF(LK EQ. 0) GOTO 585
140 CONTINUE
291 M=NXNOD(L/M) QNEXT NODE UNDER M ON L-SURFET
292 IF(M .NE. 0) GOTO 580 QNO MORE NODES UNDER M ON L-SURFET
293 K=NXNOD(K+KI) QSTART WITH NEXT NODE ON K-SURFET
294 IF(K EQ. 0) GOTO 585
295 K=MMR(K)
296 M=NL
297 GOTO 580
298 GOTO 580
299 585 CONTINUE
300 L=ARRAY(L/M) QNEXT SUCCESSOR
301 IF(L .NE. 0) GOTO 570
302 660 CONTINUE
303 K=ARRAY(K/M) QNEXT SUCCESSOR
304 IF(K .NE. 0) GOTO 560 QNOT AN END YET, REPEAT THE SEARCH
305 C ALL POSSIBLE COMBINATIONS HAVE BEEN TRIED
306 C IF(MAXSAV LE. 0) GOTO 599 QNO NEED TO GO FURTHER
307 C
308 C UPDATING NETWORK BASED ON UP-TO-DATE MAXIMUM COST SAVING
309 C PARAMETERS
310 C JTALLY=JTALLY+1
311 CALL UPNE1W
312 C REFERENCE POINT
313 RSPMAX=0.
314 MAXSAV=0
315 MAXK=0
316 MAXL=0
317 MAXM=0
318 MAXK=0
319 MAXL=0
320 MAXM=0
321 MAXL=0
322 MAXM=0
323 MAXL=0
324 LINNEW=0
325 MAXNOL=0
326 RHOMAX=0
327 GOTO 5000
328 599 CONTINUE
330 C PRINT OUT COSTS FOR THE OPTIMIZED MULTIDROP NETWORK
331 C CALL MUTDRP
332 C PRINT OUT THE OPTIMIZED MULTIDROP NETWORK
333 C CALL NFTPR
334 IF(MPLOT NE. 1) GOTO 50
335 CALL CALPLT
340 50 CONTINUE
341 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 COSTK=COST
C ITALLY=ITALLY+1
C IF(INTRY .EQ. 1) GOTO 719
C TFINARRAY(KL+1)+ARRAY(LL+1)+5
C TRFOUT=ARRAY(KL+2)+ARRAY(LL+2)+5
C C C C C C
C FIND THE LINE WITH THE ENOUGH CAPACITY TO HANDLE
C THE TOTAL TRAFFIC ON THE PROPOSED SUBNETWORK LL
C C CALL LINOFL(TFIN+TRFOUT+LDUMMY+LINNEW+1+AHM)
C IF(LINCAP(LINNEW) .EQ. 9600) GOTO 132
C LINUP=LINNEW-1
C IF(LINUP .EQ. 0) GOTO 712
C DO 711 NL=I+LINUP
C IF(LDUMMY(NL) .EQ. 0) GOTO 711
C GOTO 132
C 711 CONTINUE
C 712 CONTINUE
C NLNEW=LDUMMY(LINNEW)
C IF(NLNEW .LT. I) GOTO 132 MORE THAN I LINE NOT ALLOWFD
C COST=TCOSTF(ML+1)
C LINOLD=TCOSTF(LL+2)
C NLOLD=TCOSTF(LL+3)
C MCOSTL=COST
C 730 CONTINUE
C C TEST RESPONSE TIME, IF NOT SATISFIED, INCREASE LINE CAPACITY
C C CALL RSPTST(KL+LL+LINNEW,INK)
C IF(INK .EQ. 2) GOTO 3001
C 737 CONTINUE
C C IF LINE TYPE IS THE HIGHEST, NO NFND TO GO FURTHER
C C IF(LINNEW .EQ. 1) GOTO 132
C LINNEW=LINNEW+5
C LINNEW=LINNEW+1
C IF(LINCAP(LINNEW) .EQ. 9600) GOTO 132
C LDUMMY(LINNEW)=1
C NLNEW=NLNEW+1
C CALL RHOLUN(TFIN+TRFOUT+LDUMMY+LINNEW+AHM+LINOL+AHO)
C GOTO 3000
C 3001 CONTINUE
C IF(LINNEW.EQ.LNOLD.AND.NLOLD.EQ.1) GOTO 131
C CALL LCOSTK(IRSC+LL+1+MCOSTL) NEW COST FOR SUBNET LL
C 301 CONTINUE
C LINOLD=TCOSTF(KL+2)
C MCOSTK=TCOSTF(KL+1)
C NLOLD=TCOSTF(KL+3)
C IF(LINNEW .EQ. LINOLD .AND. NLOLD .EQ. 1) GOTO 133
C CALL LCOSTK(IRSC+KL+1+MCOSTK) NEW COST FOR SUBNET UNDER KL
C
GO TO 134
CONTINUE
ITEMP=0
KADD=0
KCHG=2
CALL ISUM(IRSCKLNEN+ITEMP)
MCOST=MCOST-ITEMP
CONTINUE
INTRY=1 FLAG THAT INDICATES AN ENTRY TO TRYLINK
JSAV=MCOSTKL+IL+COSTE(KL+1)-(MCOSTK+MCOST)
CONTINUE
COSTKM=0
KADD=0
CALL ISUM(MIL+KIL+LINNEW+COSTKM)
ISAV=JSAV+COSTKM
IF (JSAV +LE. MAXSAV) GO TO 132
RSPEX=RSPEX
MAXSAV=JSAV
MAXK=KL
MAXL=LL
MAXM=ML
MAXI=KIL
MAXLN= LINNEW
MAXNL=HILNEW
Rው-non=RHOD
CONTINUE
RETURN
SUBROUTINE LCOST(INA+N=TCOST)
***************
FIND COST FOR A SUBNETWORK, NA=BEGINNING NODE FOR THE SUBNET
TO BE EVALUATED.
IN=1 WHEN COST FOR CENTRAL LINK NA IS TO BE INCLUDED.
IN=0 WHEN COST FOR CENTRAL LINK NA IS NOT TO BE INCLUDED.

INTEGER TCOST
TCOST=0
KCHG=2
CALL ISUM(INA+LINNEW+TCOST)
C START COMPUTING SUBNET COST
C JSON=ARRAY(INA+2) ① FIRST SUCCESSOR
IF JSON.eq.0 GO TO 400
CONTINUE
JPA=ARRAY(JSON+5)
CALL ISUM(JPA+JSON+LINNEW+TCOST)
JSON=XTNOD(INA+JSON)
IF (JSON +EG. 0) GO TO 400 ② CALL IT AN END
GO TO 300
CONTINUE
IN=1 RETURN
ITEMP=0
KADD=0
CALL ISUM(INA+LINNEW+ITEMP)
TCOST=TCOST+ITEMP
RETURN

FUNCTION NXTNOD(LI,M1)

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

C

FIND THE NEXT NODE IN THE SUBNET LI WHICH M1 BELONGS TO.

C IN THE PROCESS, IF THE NEXT NODE IS LI, 0 IS RETURNED.

C OTHERWISE THE NEXT NODE IS RETURNED.

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

NXTNOD=0

MM=M1

KSON=IARRAY(MM+2)

IF (KSON.EQ.0 .AND. MM.EQ. L1) RETURN 0

IF (KSON.EQ.0) GO TO 1

NXTNOD=KSON

RETURN

CONTINUE

C

LOOK FOR HIS NEXT BROTHER

C

KRRO=IARRAY(MM+3)

IF (KRRO.EQ.0) GO TO 2

GO TO HIS FATHER

C

MM=IARRAY(MM+5)

IF (MM.NE. L1) GO TO 1

RETURN

GOTO THE BEGINNING

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

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

IARRAY(MAX1)=IARRAY(MAX1)+NOK

ARRAY(MAX1)=ARRAY(MAX1)+ARRAY(MAXK)

ARRAY(MAX2)=ARRAY(MAX2)+ARRAY(MAXK)

C

UPDATE THE COSTEW

C

COSTEW(MAX1)=COSTEW(MAX1)+COSTEW(MAXK+1)-MAXSAV

COSTEW(MAX2)=MAXLIN

COSTEW(MAX3)=MAXNOL

COSTEW(MAXK+1)=0

COSTEW(MAXK+2)=0

COSTEW(MAXK+3)=0

MAXK=NUMR(MAXK)

MAXK=NUMR(MAXK)

MAXK=NUMR(MAXK)

COSTEW(MAXL+4)=COSTEW(MAXL+4)+COSTEW(MAXK+1)+DIST(MAXKL+MAXK)
* -DIST(MAXMN,NNI)
  RHOF(MAXL)=RHOMAX
  COSTEW(MAXX+3)=0
  C UPDATE MULTIDROPPED-LINE RESPONSE TIMF
  C
  T1RSP(MAXL)=RSPMAX
  CONTINUE
  KIPA=ARRAY(MAXKI+5)  # REWEIGH KIP'S PRDCESSOR
  MSON=ARRAY(MAXX+2)  # DM'S 1ST SUCCESSOR
  CALL LNKOFF(MAXKI)  # ONE OF KI AS A SUCCESSOR OF KIPA
  IARRAY(MAXX+2)=MAXKI
  IARRAY(MAXKI+5)=MAXM
  IARRAY(MAXKI+3)=MSON
  IF(MSON+NE.0)IARRAY(MSON+4)=MAXKI
  IARRAY(MAXKI+4)=0
  MAXM=MAXKI
  MAXKI=KIPA
  IF(MAXM+NE.MAXKI)GOTO 91
  RETURN
  FUNCTION JCOSTA(N,KREF)
  C
  CJOSTA=0
  DO 777 KI=1,KREF
     JCOSTA=JCOSTA+ICSTLN(KI,N)
  777 CONTINUE
  RETURN
  FUNCTION JCOSTB(N,M,KREF)
  C
  CJOSTB=0
  DO 77A KK=1,KREF
     JCOSTB=JCOSTB+ICSTHN(KK,N,M)
  77A CONTINUE
  RETURN
  SUBROUTINE NETPR
  C
  PRINT OUT CONFIGURATION OF THE MULTIDROP NETWORK
  C
  DO 196 KK=1,N1
     IBLANK(KK)=JBLANK
  196 CONTINUE
  NNP=NUMMAR(IRSC)
  WRITE(INT1,197) NNP
  197 FORMAT(*"REGIONAL CENTER=","A4","/","X","*/ONE NETWORK","/","X",
                 +"BEGIN AT","/","/","/","/"
                 *KP=1
                 KI=1ARAY(IRSC+2)

A-25
C LOOK FOR ITS FIRST SUCCESSOR
574
C
576 CONTINUE
577 ISON=IARRAY(IPOINT+2) % CURRENT NOMINAL INDEX
578 IF(ISON .EQ. 0) GOTO 191 % AND MORE SON
579 KPKP=1 % GA LEVEL DEFFER
580 ISON=NUMR(ISON)
581 WRITE(IWT+192) ( IBLANK(I), I=1,KP ), ISONP
582 FORMAT(1X,P4(A6))
583 IPOINT=ISON
584 GOTO 190
585 CONTINUE
586 C
587 C LOOK FOR NEXT SUCCESSOR WITH THE SAME PREDECESSOR
588 C
589 IRRO=IARRAY(IPOINT+3)
590 IF(IRRO .EQ. 0) GOTO 193
591 IARR=NUMR(IRRO)
592 WRITE(IWT+192) ( IBLANK(I), I=1,KP ), IARR
593 IPOINT=IRRO
594 GOTO 190
595 CONTINUE
596 C
597 C NEXT LEVEL UP
598 C
599 KPKP=1
600 IPOINT=IARRAY(IPOINT+5)
601 IF(KP .EQ. 0) GOTO 194 % AND NPEN TO GO FURTHER
602 GOTO 191
603 CONTINUE
604 RETURN
605 SUBROUTINE CONVRT(ICOST)
606 ***************
607 C
608 C CONVERT A NUMBER INTO ITS FIELD EQUIVALENT
609 C
610 ***************
611 JCHAR(I)=JRLANK
612 JCHAR(2)=JRLANK
613 IF(ICOST .EQ. 0) GOTO 916
614 ENCODE(198,JCHAR) ICOST
615 CONTINUE
616 RETURN
617 SUBROUTINE SUMPRTN(NREF,NN)
618 ***************
619 C
620 C SUM UP COSTS AND PRINTS
621 C
622 ***************
623 TCOST1=0
624 TCOST2=0
625 DO 779 K=1,NREF
626 779

A-26
ITCOST(K+1)=ICSTLN(K+1)
ITCOST(K+2)=ICSTLN(K+2)
DO 7791 K=1,N4
630 ITCOST(K+1)=ITCOST(K+1)+ICSTHW(K,K+1)
631 ITCOST(K+2)=ITCOST(K+2)+ICSTHW(K,K+2)
632 7791 CONTINUE
633 TCOST1=ITCOST1+ITCOST(K+1)
634 TCOST2=ITCOST2+ITCOST(K+2)
635 779 CONTINUE
636 KCOST=TCOST1+TCOST2
637 C
638 C PRINT OUT COST
639 C
640 NTURN=NREF/101
641 IREM=MOD(NREF,10)
642 IF(IREM .EQ. 0) NTURN=NTURN+1
643 LPAGE=1
644 DO 919 KW=1,NTURN
645 KLW=10*(KW-1)+1
646 KWW=10+KW
647 IF(KWW.GT.NREF) KWW=NREF
648 IF(KWW .NE. 1) GOTO 797
649 IF(LPAGE .NE. 1) GOTO 9033
650 WRITE(9031,1)
651 9031 FORMAT('REGISTAL STAR NETWORK AND ITS COSTS='10X)
652 GOTO 9035
653 9033 CONTINUE
654 WRITE(9034,1)
655 9034 FORMAT('REGIONAL STAR NETWORK AND ITS COSTS='10X)
656 9035 CONTINUE
657 WRITE(9036,1)
658 9036 FORMAT('REGIONAL STAR NETWORK AND ITS COSTS='10X)
659 WRITE(9037,1)
660 9037 FORMAT('REGIONAL STAR NETWORK AND ITS COSTS='10X)
661 9038 CONTINUE
662 WRITE(9039,1)
663 9039 FORMAT('REGIONAL STAR NETWORK AND ITS COSTS='10X)
664 WRITE(9040,1)
665 9040 FORMAT('REGIONAL STAR NETWORK AND ITS COSTS='10X)
666 WRITE(9041,1)
667 9041 FORMAT('REGIONAL STAR NETWORK AND ITS COSTS='10X)
668 WRITE(9042,1)
669 9042 FORMAT('REGIONAL STAR NETWORK AND ITS COSTS='10X)
670 GO TO 806
671 879 CONTINUE
672 IF(LPAGE .NE. 1) GOTO 8033
673 WRITE(8034,1) KW
674 8031 FORMAT('FINAL MULTIDROP NETWORK AND ITS COSTS='10X)
675 GOTO 8035
676 8033 CONTINUE
677 WRITE(8034,1) KW
678 8034 FORMAT('FINAL MULTIDROP NETWORK AND ITS COSTS='10X)
679 8035 CONTINUE
680 WRITE(8035,1) (KW,KWL,KWW)
681 803 Final network number 16X,18(1A1X)
682 DO 1803 N=KWL,KWW
683 1803 FORMAT('INKW')

A-27
77-53, Vol. IV

584 MSUB(N)=NUMARR(ID)
585 LSUB(N)=IARRAY(ID)+1
586 1801 CONTINUE
587 WRITE(IWT+1,106) (MSUB(N),N=KW1,KW4)
588 1806 FORMAT(5X*BEGINNING NODE=11X*10(F8.1X,1X))
589 WRITE(IWT+1,107) (LSUB(N),N=KW1,KW4)
590 1807 FORMAT(3X*NO. OF TERMS=12X*10(I4,1X))
591 WRITE(IWT+1,111)
592 811 FORMAT(3X*NO. OF LINES)
593 DO 1008 NJ=1,N3
594 IF(LINEX(NJ,FO,0) GOTO 1008
595 WRITE(IWT+904) LINAME(NJ),(NLINES(K,NJ),K=KW1,KW4)
596 1008 CONTINUE
597 WRITE(IWT+1006) (RHOF(NJ),NJ=KW1,KW4)
598 8036 FORMAT(3X*LINE UTILIZATION=9X*10(F9.3,1X))
599 WRITE(IWT+1008) (DIST(N),N=KW1,KW4)
600 804 FORMAT(3X*TOTAL MILEAGE=12X*10(1B,1X))
601 806 CONTINUE
602 DO 1101 N=KW1,KW4
603 I=1N
604 IF(NN+FO,0) IN=NSUB(N)
605 TRFSUM(N,1)=ARRAY(ID)+1
606 TRFS(N,M,2)=ARRAY(ID)+2
607 TIMOUT(N)=TIMSP(ID)
608 1101 CONTINUE
609 WRITE(IWT+1102) (TRFSUM(N),N=KW1,KW4)
610 1102 FORMAT(3X*TRAFFIC ROUTE=11X*10(F9.3,1X))
611 WRITE(IWT+1103) (TRFSUM(N),N=KW1,KW4)
612 1103 FORMAT(3X*CPU TO LINE TIME=10X*10(F9.3,1X))
613 WRITE(IWT+1104) (TOUTOUT(N),N=KW1,KW4)
614 1104 FORMAT(3X*TOTAL RESPONSE TIME=7X*10(F9.3,1X))
615 WRITE(IWT+907)
616 907 FORMAT(3X*SUBTOTAL=11X*10(F8.1X,1X))
617 COST=JCOST(N,1,REF)
618 IF(KW,NE,1) COST=0
619 CALL CONVRT(COST)
620 WRITE(IWT+908) (JCHAR(L),L=1,2),(ICSTM(NODE,N),NODE=KW1,KW4)
621 908 FORMAT(5X*LINES,8X,6A2,1X*10(1B,1X))
622 DO 1099 K=1,N4
623 COST=JCOST(K,1,REF)
624 IF(KW,NE,1) COST=0
625 CALL CONVRT(COST)
626 WRITE(IWT+909) NAMEHW(K),(JCHAR(L),L=1,2),(ICSTM(NODE,K,1),
627 * NODE=KW1,KW4)
628 909 FORMAT(5X+A6,7X+A6,A2,1X*10(1B,1X))
629 190a CONTINUE
630 WRITE(IWT+910)
631 910 FORMAT(5X*ANNUAL RECURR. COST*)
632 COST=JCOST(2,REF)
633 IF(KW,NE,1) COST=0
634 CALL CONVRT(COST)
635 WRITE(IWT+908) (JCHAR(L),L=1,2),(ICSTM(NODE,K,1),NODE=KW1,KW4)
636 DO 1091 K=1,N4
637 COST=JCOST(K,2,REF)
638 IF(KW,NE,1) COST=0
639 CALL CONVRT(COST)
640 WRITE(IWT+909) NAMEHW(K),(JCHAR(L),L=1,2),(ICSTM(NODE,K,2),
741 * NODE=KWL+KWU*
742 1911 CONTINUE
743 WRITE(IWT,912)
744 912 FORMAT(1X, 'TOTAL COST: ')
745 IF(KW.NE.1) TCOST=1
746 CALL CONVERT(TCOST)
747 WRITE(IWT,913) (JCHAR(L),L=1,2),(ITCOST(K,1),K=KW+KWL)
748 IF(KW+NF.NE.1) TCOST=2
749 CALL CONVERT(TCOST)
750 WRITE(IWT,914) (JCHAR(L),L=1,2),(ITCOST(K,2),K=KWL+KWU)
751 913 FORMAT(4X,'INST. COST: ',4X,'A6=A2+1X*1019+1X')
752 914 FORMAT(4X,'RECUR. COST: ',3X,'A6=A2+1X*1019+1X')
753 LPAGE=LPAGE+1
754 LPAGE=MOD(LPAGE,2)
755 919 CONTINUE
756 WRITE(IWT,695) K45T
757 695 FORMAT(1X, TOTAL COST= ' I8)
758 RETURN
759 SUBROUTINE MUTDRP
760 C ********************************************
761 C PRINT OUT FINAL MULTIDROP NETWORK WITH ITS COSTS
762 C ********************************************
763 DO 590 NL=1,N3
764 LDUMMY(NL)=0
765 590 CONTINUE
766 IRO=IAARRAY(IRSC+2) 'FIRST SUCCESSOR
767 K1=1
768 699 CONTINUE
769 IF(IRO.EQ.0) GOTO 698
770 NK=NMB(IRO)
771 NK1=NI1
772 NSIR(K1)=IRO
773 LINE=COSTW(IRO+2)
774 LDUMMY(LINE)=COSTFW(IRO+3)
775 JS=IAARRAY(IRO+2)
776 IF(JSN.EQ.0) GOTO 694
777 DO 592 NK=1,N4
778 ICSWN(K1+NM)=0
779 DO 592 NK=1,N4
780 ICSTWN(K1+NK+NM)=0
781 592 CONTINUE
782 DO 596 NL=1,N3
783 NLINES(K1+NL)=LDUMMY(NL)
784 596 CONTINUE
785 KCHG=2
786 312 CONTINUE
787 CALL COSTJ(LDUMMY,NK1+NK2+LNKCHW1+NCWL)
788 DO 595 NL=1,N3
789 595 CONTINUE
790 DO 595 NM=1,N4
791 ICSTLN(K1+NM)=ICSTLN(K1+NM)+LNKCLN(LNL+NM)
792 DO 595 NK=1,N4
793 ICSTWN(K1+NK+NM)=ICSTWN(K1+NK+NM)+LNK.CHW(LNL+NM)
794 IF(JSN.EQ.0) GOTO 311
795 595 CONTINUE
796 IF(JSN.EQ.0) GOTO 311
797 NK2=NMB(JSN) 'GLOBAL INDEX FOR NEXT NODE
NK1=ARRAY(JSON,5) GPREDECFSOR
NK1=NUMARR(NKI) @GLOBAL INDEX FOR PRDECFSOR
JSON=JSOND(IARO,JSON)
GOTO 312
CONTINUE

CONTINUE
LIUMMY(LINE)=0
GOTO 591
CONTINUE

C USE PREVIOUS DATA
DO 597 NL=1,N3
NLINES(K1+NL)=NLINES(IARO,NL)
CONTINUE
DO 598 NM=1,2
ICSTLN(K1,NM)=ICSTLN(IARO,NM)
DO 598 NK=1,N4
ICSTHW(K1,NK+NM)=ICSTHW(IARO,NK,NM)
CONTINUE
LUMMY(LINE)=0
CONTINUE

IF (IARO+4) GOTO 599
CONTINUE

CONTINUE
IF (JSON+2) = FIRST SUCCESSOR
CONTINUE

IF (JSON+2) = FIRST SUCCESSOR
IF (JSON+2) = FIRST SUCCESSOR
GOTO 191
CONTINUE

CONTINUE
GOTO 190
CONTINUE

CONTINUE
IARO=ARRAY(IPOINT,3)
855  IP=IARRAY(IPOINT+5) GOTO ITS PREDECESSOR
856  CALL TRSFPM(2)
857  IF(KP .EQ. 0) GOTO 193
858  IPOINT=KP
859  CALL TRSFPM(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 TRSFPM(3)
870  RETURN
871  SUBROUTINE TRSFPM(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(IPOINT) GGLOBAL INDEX
880  IDD=MAPADR(LKI) GMAPADR INDEX FOR LKI
881  IF(IDD .EQ. IP) RETURN
882  IP=IDD
883  666 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  IFRON=IARRAY(MP+4) THE SUCCESSOR REFOR MP
893  IRAK=IARRAY(MP+3) THE SUCCESSOR AFTER MP
894  IF(IFRON .NE. 0) GOTO 92
895  MP=IARRAY(MP+5)
896  IARRAY(MP+2)= IRAK 1ST SUCCESSOR UNDER NEW MP
897  GOTO 99
898  92 CONTINUE
899  IARRAY(IFRON+3)=IRAK
900  CONTINUE
901  IF(IRAK .EQ. 0) RETURN
902  IARRAY(IRAK+4)=IFRON
903  RETURN
904  SUBROUTINE RSPTST(KKK,LLL,LINMAX,ITOK)
905  ***************
906  C
907  C TEST RESPONSE TIME, SATISFIED WHEN ITOK=1
908  C
909  ***************
910  MDROP=IARRAY(LLL+1)+IARRAY(KKK+1)+2
911  TRFIN=ARRAY(LLL+1)+ARRAY(KKK+1)
912  TRFOUT=ARRAY(LLL+2)+ARRAY(KKK+2)
913  CALL RSPNSE(TRFIN,TRFOUT,LINMAX,MDROP,ITOK)
914  RETURN
915  END

QPR TRACOM IRNOP/0777
**STACOM/11, IRNOP/0777**

1. SUBROUTINE IRNOP(NR,LIMIT,TRM)

   1.1 ***************

2. C SUPROGRAM FOR THE INTER-REGION NETWORK OPTIMIZATION

3. C LIMIT=MINIMAL NUMBER OF PATHS NEEDED PER REGIONAL

4. C SWITCHING CENTER

5. C

6. C ***************

7. C PARAMETER NP1=130, NP2=1, NP3=4, NP4=3

8. C PARAMETER NPC=360, NP6=(NPC+NP/C-1)/4+1

9. C PARAMETER MP7=4, INT=100, MW=4

10. C COMMON/CONST/N1,N2,N3,N4,MW,MP1

11. C COMMON/LINCHAR/LINMIX(NP3),LIMIT(NP3)* UTILIZ(NP3)

12. C * /BCOS*/AINSTC(NP2,NP3,NP4,NP5,NP6,NP7),RFRCRN(NP2,NP3,NP4,NP5,NP6,NP7)

13. C * /NAME/LINXPT(NP1),NAMEF(NP1),LINAME(NP3),NAMECW(NP3)

14. C * /FAC/SYRA(NP1),NPSC(NP1),NP6R(NP1),TRFAN(NP1),TRAFIT(NP1)

15. C * /RFFF/IFEF(NP1),TRFN(NP1),TRNS(NP1),MAPADP(NP1)

16. C DIMENSION NETSUM(NP3,2),ORINET(MW,MW,NP3)

17. C DIMENSION NLINK(NP3),LNKCHV(NP3,NP4,2),LNCLIN(NP3,2)

18. C INTEGER SUMCST

19. C INTEGER ORINT

20. C DIMENSION TRM(MW,MW),TR(MW,MW)

21. C INTEGER ORICST,ORICS1,ORIC52

22. C INTEGER DTVTRJ(MW),DTVTRJ(MW)

23. C DIMENSION TRM(MW,MW),TR(MW,MW)

24. C NETCF(MW,NP3),LINAD(NP3),LINAD(NP3)

25. C LINEI(NP3),LINEA(NP3),LINEC(NP3)

26. C DIMENSION RHEF2(MW,MW)

27. C DIMENSION TRM(MW,MW)

28. C EQUIVALENCE (LINE1,LINCF,LINRD)

29. C

30. C RESET UTILIZATION FACTOR TO .5

31. C

32. C

33. C

34. C DO 70 NN1=1,N3

35. C DO 70 CONTINUE

36. C UTILIZ(NN1)=.5

37. C DO 101 I=1,NR1

38. C

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

40. C THE NUMBER OF CHARGEABLE ITEMS

41. C

42. C ORICST=0

43. C ORICS1=0

44. C ORIC52=0

45. C NR1=NR1-1

46. C DO 203 NN1=1,N3

47. C DO 203 NN2=1,N2

48. C NETSUM(NN1,NN2)=0 GCOST SUM

49. C CONTINUE

50. C

51. C MODIFY Duplexing MODE FROM HALF TO FULL DUPLF

52. C

53. C DO 667 K1=1,N3

54. C IDUPLX(K1)=2

55. C CONTINUE

56. C DO 101 I=1,NR1

---

A-32
77-S3, Vol. IV

57 NLINK(I)=NR1  QNR1 LINKS AT THE BEGINNING
58 II=I+1
59 DO 102 J=I1+NR
60 II=NRSC(I)
61 JUNRSC(J)
62 ATRMAX=AMAX1(TR(I,J), TR(J,I))  QASSUMING FULL DUPLEX
63 CALL LINNUM (ATRMAX,0.*LINE0+LINUP+0.*RHO)
64 RHOF2(I,J)=RHO
65 RHOF2(J,I)=RHO
66 CALL ITCOSTJ(LINEQ*I,J,J, LNKCHW*LN CLN)
67 DO 104 NN=1+NN
68 ORINET(I,J,NN)= LINEQ(1NN)
69 ORINET(J,I,NN)= LINEQ(1NN)
70 DO 105 NN=1+NN  G LINE COST
71 NETSUM(NN+NM)= NETSUM(NN+NM)+ LNKCHW(NN,NN)
72 DO 106 NN=1+NN  G HARDWARE COSTS
73 NETSUM(NN+MN)= NETSUM(NN+NM)+ LNKCHW(NN+NN)
74 106 CONTINUE
75 105 CONTINUE
76 104 CONTINUE
77 103 CONTINUE
78 102 CONTINUE
79 DO 107 K1=1+NR
80 DO 107 NN=1+NN
81 ORINET(K1,K1,NN)=0
82 107 CONTINUE
83 CALL OUTFRT(I)
84 ITALL=0
85 999 CONTINUE
86 MAXSAV=0
87 DO 777 I=1+NR
88 IF (NLINK(I) .LE. LIMIT) GO TO 777
89 I=I+1
90 DO 788 J=I1+NR
91 IF (NLINK(J) .LE. LIMIT) GO TO 788
92 IN=TEST(ORINET+I,J)
93 IF (IN .EQ. 0) GO TO 788  G NO LINK TO BE DELETE
94 C
95 C DETERMINE WHETHER THERE IS A LINK CONNECTED BY AT MOST ONE INDIRECT
96 C ROUTE BETWEEN ANY TWO REGIONS IN THE NETWORK WHEN THE DIRECT LINK
97 C BETWEEN I AND J IS ELIMINATED. THE INDIRECT LINK ONLY GOES THROUGH
98 C ONE INTERMEDIATE RSC.
99 C
100 DO 139 L=1+NR
101 LI=L+1
102 DO 138 ML=1+NR
103 ITTEST(I,J,L,M)
104 IF (1Y.EQ. 1) GO TO 110  GNEXT STEP NOT TO BE TESTED
105 ITTEST(ORINET+L,M)
106 IF (IN .EQ. 1) GO TO 138
107 R10 CONTINUE
108 DO 137 NL=1+NR
109 IF (L .EQ. 0) GO TO 137
110 ITTEST(I,J,L,N)
111 IF (1Y.EQ. 1) GO TO 137
112 ITTEST(ORINET+L,N)
113 IF (IN .EQ. 0) GO TO 137
114    IY=IY-TEST(I,J,N,M)
115    IF(IY .LE. 1) GO TO 137
116    IY=IY-TEST(ORINFT,M,N)
117    IF (IN .EQ. 1) GO TO 138
118  137 CONTINUE
119    GO TO 788
120  138 CONTINUE
121  139 CONTINUE
122    SIDE=0
123    SOUTJ=0
124    CALL TRDIV(IFLOP)
125    IF (IFLOP .EQ. 1) GO TO 201
126    CALL MINAD(IJAD,MINCT)
127    GO TO 202
128  201 CONTINUE
129    CALL NETWKC(MINCT)
130  202 CONTINUE
131    ISAV=ORICST-MINCT
132    IF (MAXSAV .GE. ISAV) GO TO 78A
133    MAXSAV=ISAV
134    IFLOP=IFLOP .OR. GLOBAL INDICATOR
135    IF (IFLOP .EQ. 1) GO TO 204
136    IMAX=IJAD
137    JMAX=J
138    DO 666 NN=1,N3
139       ILINAD(NN)=LINADT(NN) CHANGE OF LINE RFG.
140       JLINAD(NN)=LINADJ(NN)
141  666 CONTINUE
142  204 CONTINUE
143    DO 331 K1=1,NR
144    DO 331 K2=1,NR
145    TRRM(K1,K2)=TRR(K1+K2)
146  331 CONTINUE
147  208 CONTINUE
148  788 CONTINUE
149  777 CONTINUE
150    IF (MAXSAV .LE. 0) GO TO 9999
151    CALL NETUP(IFLIP,JMAX,JMAX,JMAX)
152    ITALL=ITALL+1
153    GO TO 9999
154  9999 CONTINUE
155    109 FORMAT(1x, 'THIS NETWORK HAS BEEN UPDATED FOR *H* TIMES**//')
156    WRITE(6,109) ITALL
157    DO 81 I=1,N3
158    DO 81 J=1,2
159    NETSUM(I,J)=0
160  81 CONTINUE
161    DO 91 I=1,NR1
162    K=I+1
163    DO 92 J=K,NR
164    DO 93 K1=1,N3
165    LINEO(K1)=ORINET(I,J,K1)
166  93 CONTINUE
167    II=NRSC(I)
168    JJ=NRSC(J)
169    CALL ICOSTJ(LINE9,II,JJ,LNKCHW,LNKCLN)
170    DO 94 KK=1,N3

A-34
NETSUM(KK+1) = NETSUM(KK+1) + LNKCLN(KK+1)
NETSUM(KK+2) = NETSUM(KK+2) + LNKCLN(KK+2)

DO 95 KL=1, N4
NETSUM(KK+1) = NETSUM(KK+1) + LNKCHW(KK+1)
NETSUM(KK+2) = NETSUM(KK+2) + LNKCHW(KK+2)

95 CONTINUE
94 CONTINUE
92 CONTINUE
91 CONTINUE
ORICS1=0
ORICS2=0
CALL OUTPR T2
RETURN

C SUBROUTINE OUTPR T(N)
C
C PRINT OUT INTERREGIONAL NETWORK CONFIGURATIONS AND ITS COSTS
C
C
DO 110 I=1, N4
ORICS1=ORICS1+NETSUM(I+1)
ORICS2=ORICS2+NETSUM(I+2)
110 CONTINUE
NTURN=NR/N4+1
DO 201 L=1,NTURN
LL=(NTURN-1)*10 + 1
LU=LL+10
IP(LU,LL)=NR
IF(LL .GE. 200) GOTO 2100
WRITE(WT,2002) (J,J=LL,LU)
2002 FORMAT(I1,1X,INITIAL INTERREGIONAL NETWORK CONFIGURATION)
203 GOTO 2101
2100 CONTINUE
2101 WRITE(WT,2010) (J,J=LL,LU)
2010 FORMAT(F11,10X,FINAL OPTIMAL INTERREGIONAL NETWORK CONFIGURATION)
207 * //20X,10(5X,I3,2X)
203 GOTO 2101
2101 CONTINUE
DO 209 I=1, NR
WRITE(WT,2004) I
2004 FORMAT(1X,REGION//,I4)
DO 212 M=1, N4
WRITE(WT,2008) LINAME(M), (ORINFT(I:M), K=LL,LU)
2008 FORMAT(1X,REGION//,15X,POINT NAME(M), (ORINFT(I:M), K=LL,LU)
212 CONTINUE
DO 220 K=1, N3
WRITE(WT,2006) (K,0.0,1X,INST. COST•7X•RECV. COST•9X•SHUT-TOTAL•;
220 CONTINUE
WRITE(WT,2007) (K,LINAME(K), (NETSUM(K), I=1, N3)
2007 FORMAT(/15,3X,A6,5X,16,1X,I6,10Y,1X)
220 ORICS1=ORICS1
227 ORICS1=ORICS1+ORICS2

A-35
WRITE(IWT,2009) ORICS1,ORICS2,OP1CST
200  FORMAT(//9X,TOTAL,4X,17,10X,17,10X,1A)
230  RETURN
231  FUNCTION TEST(I,J,K,L)
232  *****************************************
233  C
234  C TEST EQUIVALENCE BETWEEN SUBSET(I,J) AND SUBSET(K,L)
235  C
236  C *****************************************
237  TEST=N
238  IF (I.EQ. K .AND. J.EQ. L) TEST=T
239  IF (I.EQ. L .AND. J.EQ. K) TEST=T
240  RETURN
241  FUNCTION NETEST(NET,I,J)
242  *****************************************
243  C
244  C TEST DIRECT LINE CONNECTIVITY BETWEEN T AND J.
245  C
246  C *****************************************
247  DIMENSION NET(MW,MW,N3)
248  DO 103 I1=1,N3
249  IF (NET(I,J,I1) .GT. 0) GO TO 10A
250  103 CONTINUE
251  NET=N  NO CONNECTION
252  RETURN
255  10A NET=T
256  RETURN  YES; THERE IS A CONNECTION
257  SUBROUTINE TRFDIV(iflop)
258  *****************************************
259  C
260  C DIVERT TRAFFIC BETWEEN I AND J THROUGH OTHER REGIONS.
261  C
262  C *****************************************
263  C IT RETURNS WITH IFLOP=1 WHEN SUCCESSFUL; OTHERWISE IFLOP=0.
264  C IT ALSO CREATES TEMPORARY MATRICES TRA AND NETCF.
265  C
266  SNDIVT=0, QTOTAL TRAFFIC DIVERTED (I TO J)
267  SNDIVT=0, QTOTAL TRAFFIC DIVERTED (J TO I)
268  DO 205 K=1,NR
269  DIVTR(I,K)=0. Q TRAFFIC DIVERTED THRU REGION K (I TO J)
270  DIVTR(J,K)=0. Q TRAFFIC DIVERTED THRU REGION K (J TO I)
271  205 CONTINUE
272  DO 220 I1=1,NR
273  IF (I1.EQ.I .OR. I1.EQ.J) GO TO 220
274  IC1=NETEST(ORINFET,I1)
275  IC2=NETEST(ORINFET,J1)
276  IF (IC1.EQ.0 .OR. IC2.EQ.0) GOTO 220
277  C
278  C DIVERT I TO J TRAFFIC THRU II
279  C
280  DIVTR(I)=0.
281  DIVTR(J)=0.
282  CALL LINTRF(I,J,A)
283  DELTR= A-TR(I,J)
284  IF (DELTR.IE.0.0) GO TO 160
DIVTR=DELT
CALL LINTR(I,J+A)
DFLTR=A-TR(I,J)
IF (DFLTR.LE.0.0) GO TO 169
DIVTR(I)=AMIN(DFLTR,DIVTR)
IF ((DIVTR(I)+SDIVTI).GT.TR(I,J)) GO TO 140
SDIVTI=SDIVTI+DIVTR(I)
GO TO 160

C
140 DIVTR(I)=TR(I,J)-SDIVTI
SDIVTI=SDIVTI
C
DIVERT J TO I TRAFFIC THRU II
C
160 CONTINUE
CALL LINTR(J+II+A)
DFLTR=A-TR(J+II)
IF (DFLTR.LE.0.0) GO TO 220

C
DIVTR=DELT
CALL LINTR(I,I+B)
DFLTR=A-TR(I,I)
IF (DFLTR.LE.0.0) GO TO 220
DIVTR(I)=AMIN(DFLTR,DIVTR)
IF ((DIVTR(I)+SDIVTI).GT.TR(I,I)) GO TO 140
SDIVTI=SDIVTI+DIVTR(I)
GO TO 220

C
180 DIVTR(I)=TR(J+I)-SDIVTI
SDIVTI=TR(J+I)
C
200 CONTINUE
IF ((SDIVTI.EQ.TR(J+I)).AND.(SDIVTI.EQ.TR(J+I))) GO TO 340

C
220 CONTINUE
IFLOP=0
GO TO 360

C
340 IFLOP=1
CONTINUE
C
C CREATE A NEW TRAFFIC MATRIX WHICH ELIMINATES THE TRAFFIC BETWEEN
C NODES I AND J AND A TEMPORARY NETWORK NETCNF FOR THE PURPOSE
C OF COST EVALUATION
C
DO 191 K1=1+NR
DO 191 K2=1+NR
TR(K1,K2)=TR(K1,K2)
DO 191 K3=1+NR
NETCNF(K1,K2,K3)=ORINT(K1,K2,K3)
191 CONTINUE
DO 190 K1=1+NR
NETCNF(I,J+K1)=0
NETCNF(J+K1)=0
NETCNF(I,K1)=0
190 CONTINUE
TH(R(I,J))=0
TH(R(J,I))=0
DO 380 IK=1, NR
344 IF (IR.EQ. IK .OR. J.EQ. IK) GO TO 380
345 TTR(I+IK)= TTR(I+IK)+ DIVTR(IK)
346 TTR(J+IK)= TTR(J+IK)+ DIVTR(IK)
347 TTR(I+IK)= TTR(I+IK)+ DIVTR(JK)
348 ATRMAX= MAX (TTR(I+IK)+TTR(J+IK))
349 BTRMAX= MAX (TTR(J+IK)+TTR(I+IK))
350 IDR1= NRSC(I)
351 IDR2= NRSC(IK)
352 CALL LINNUN (ATRMAX, 0, LINFO, LINUP, 0, RHO)
353 RHOF2(I+IK)= RHO
354 RHOF2(IK+J)= RHO
355 IDR1= NRSC(J)
356 CALL LINNUN (BTRMAX, 0, LINFO, LINUP, 0, RHO)
357 RHOF2(I+JK)= RHO
358 RHOF2(IK+J)= RHO
359 DO 430 NR=1, NR3
360 NETCNF(I, INN)= LINFO(NNN)
361 NETCNF(IK, INN)= LINFO(NNN)
362 NETCNF(I+IK, INN)= LINFO(NNN)
363 NETCNF(JK+IK, INN)= LINFO(NNN)
364 430 CONTINUE
365 380 CONTINUE
366 RETURN
367 C SURORUTINE LINTRF(I, J, A)
368 C
369 C CONVERT LINES INTO TRAFFIC CAPACITIES BETWEEN NODES I AND J.
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
C CAPACITY INCREASE IS REQUIRED WHEN TFLOP=0. AND THE CAPACITY AT
C MAXIMUM COST SAVINGS.
C
C ****************************
C
C DIMENSION LINDI(NP3),LINDJ(NP3)
C MINCST=0
C RTRF=TR(I,J)-SPLIT REMAINING TRAFFIC FROM I TO J
C RTRF=TR(J,I)-SPLIT REMAINING TRAFFIC FROM J TO I
C DO 500 II=1,NR
C IF(IT=EO.I.OR.II=FO.J) GO TO 500
C IF(IT=II) .GO TO 501
C RETURN
C SIRROUTINE MINAD(IAD,MINCST)
C
C **********************************************************
C
C C DETERMINE DELTA COST FOR INCREASED CAPACITY IN ALTERNATE ROUTES
C
C C LINK (I+II)
C
C AII=TRR(I+II)+RTAF
C AJI=TRR(I+II)+RTAF
C AM=AMAX1(AII,AJI)
C IDR1=NRSIC(I)
C IDR2=NRSIC(II)
C CALL LINNUM(AW+0.,LINEOA+LINUS+0.+RHO)
C DO 151 NR=1,NR
C LINDI(IN)=LINEOA(NI)+NETCNF(I+II,NN)
C NETCNF(I+II,NN)=LINEOA(NN)
C NETCNF(I+I+II,NN)=LINEOA(NN)
C 151 CONTINUE
C
C C LINK(I+J)
C
C RII=TRR(I+J)+RTAF
C BJ=TRR(J+I)+RTAF
C RMI=AMAX1(RII,BJ)
C IDR=NRSIC(J)
C CALL LINNUM(AW+0.,LINEOB+LINUS+0.+RHO)
C DO 111 NN=1,N3
C LINDJ(NN)=LINEOB(NN)-NETCNF(J+II,NN)
C NETCNF(J+II,NN)=LINEOB(NN)
C NETCNF(J+I+II,NN)=LINEOB(NN)
C 111 CONTINUE
C
C CALL NETWKC(SUMCST)
C C IF (SUMCST,GT,MINCST) GO TO 120
C DO 207 NN=1,N3
C LINDJ(NN)=LINDJ(NN)
C 207 CONTINUE
C 11AD =1
MINCST = SUMCST

C

C RESET TO INITIAL NETWORK CONFIGURATION FOR NEXT TRY

C

DO 250 NN= 1,N3

NETCNF(I*J,NN) = NETCNF(I*J,NN) + LIMD(J,NN)

NETCNF(J*I,NN) = NETCNF(J*I,NN) + LIMD(J,NN)

NETCNF(J,J,NN) = NETCNF(J,J,NN) + LIMD(J,NN)

250 CONTINUE

C

500 CONTINUE

TRR(I,IIAD) = TRR(I,IIAD) + RTPJ

TRR(I,IIAD) = TRR(I,IIAD) + RTPJ

TRR(I,IIAD) = TRR(I,IIAD) + RTPJ

RETURN

C

NUMOUT(I,IFLP,IIAD, I,J)

C

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

C

IF (IFLP, 0, 1) GO TO 700

C

700 CONTINUE

C

DO 900 NN= 1,N3

ORINET(I*J,NN) = ORINET(I*J,NN) + LIMD(J,NN)

ORINET(J*I,NN) = ORINET(J*I,NN) + LIMD(J,NN)

ORINET(J,J,NN) = ORINET(J,J,NN) + LIMD(J,NN)

900 CONTINUE

701 CONTINUE

C

C RESET TRAFFIC MATRIX TR(NR*NR)

C

DO 910 IR= 1,NR

DO 910 IK= 1,NR

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

910 CONTINUE

900 CONTINUE

C

C UPDATE TOTAL COST FOR OVERALL NETWORK

C

910 CONTINUE

C

C UPDATE NLINK MATRIX

C

NLINK(I) = NLINK(I) + 1

NLINK(J) = NLINK(J) + 1

RETURN

C

END
SUBROUTINE ICASTJILThlF17U

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

C CALCULATE INSTALLATION ANNUAL RECURRING COSTS NEEDED FOR
C COMMUNICATION LINK BETWEEN NODES I AND J. LNKCHW= OTHERS
C LNKCLN= LINES; I AND J ARE GLOBAL INDICE FOR SYSTEM TERMINATIONS
C UNDER CONSIDERATION LINE= LINE CONFIGURATION BETWEEN T AND J
C
C

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

PARAMETER NP1=130*NP2=1*NP3=4*NP4=3*NP=360
PARAMETER NP7=4

DIMENSION LINK(INP3), LNKCHW(NP2*NP3*NP4, NP2*NP3*NP4)
COMMON/LINCHW/LINMX(INP3), INCAP(INP3), ITILZ(INP3)

/COST/N1=N2*N4=N7=NNCY
/COST/AINSTC(NP2*NP3*NP4*NP2*NP3*NP4*NP4), RFCPC(NP2*NP3*NP4, NP2*NP3)
*AMSTLN(NP2*NP3*NP4), RECRLN(NP2*NP3*NP4), IMPLX(INP3)
*RF(INF/INFJ(NP2*NP3), IRAND(INP3), IFLAG(NP2*NP3))
*/ADV/ IADV(INP1)*KCHG*KAND*GTERMINALS WITH SAME V-H
*/RF/RF(NP), TRAFD(NP2*NP7), DSTMCE(INP6), MAPADDR(INP1)

C

C INITIALIZATION

II=MAPADR(I)
JJ=MAPADR(J)
IADV=IADV(IN)
DO 100 NL=1+NL
DO 100 NM=1+NM

100 CONTINUE

KRATE= IRAND(I+1) @ RATE STRUCTURE TYPE FOR NODE I
KRATF= IRAND(J+1) @ RATE STRUCTURE TYPE FOR NODE J
KDENS= IRAND(I+2) @ TRAFFIC DENSITY TYPE FOR NODE I
KDENS= IRAND(J+2) @ TRAFFIC DENSITY TYPE FOR NODE J

K=KDENS+1 @ ACTUAL DENSITY (NP=NP+1, 1SH-L AND 2SL-L)

K=KDENS+1 @ ODENO, 1SH-L AND 1SL-L

DST = DIST(I,J) @ DISTANCE BETWEEN NODES I AND J

INSTCOST

ITIP=1 @ PRIME COST FOR H/W UNIT
IF(DST-Le.0.5) ITIP=0 @ SYSCOPK COST FOR ADDITIONAL UNIT

KP = IRATF(KRATE, KRATF) @ ACTUAL RATE STRUCTURE TO BE USED

DO 43 IL=1+IL

INDP= DUPXL(INL) @ DUPLEXING MODF 1SH AND 2EF

43 MODV = LIGEI(INL) @ NUMBER OF LINES REQUIRED

MODV=MODO+IADV*KAND

C

C CALCULATE COSTS FOR NON-LINE TYPE CHARGES

IF (MODV.EQ.0) GO TO 1 @ NO LINES ARE REQUIRED

DO 50 IL=1+IL, 4 @ HIGH DENSITY RATE

C

C INSTALLATION COSTS FOR NON-LINE TYPE CHARGES

L=LNKCHW (IL, IV+1) = AINSTC(KP, IL, IV, KP, INDP, ITIP)*KCHG*MODV

A-41
PROGRAM NAME: ANNUAL RECURRING COSTS FOR NON-LINE TYPE CHARGES

C 1  + ANSTH(KR,IL,IV,KK)1DPX+2)1NDV
C 2 ANNUAL RECURRING COSTS FOR NON-LINE TYPE CHARGES
C
C LINKCH (IL,IV)= (REPC(KR,IL,IV,KK,IMPX,TIP)*KCMH*HNDV
C 1 + REPC(KR,IL,IV,KK,IMPX,TIP)*HNDV1)*12.
C 2 CONTINUE
C
C CALCULATE LINK COSTS
C
C LIN = IFLAG(KR,IL) 0 LINEAR IF 1 AND NONLINEAR OTHERWISE
C
C ANNUAL LINE INSTALLATION COST
C
C AN=1,
C
C LINKLN(IL,1) = ANSTLN(KR,IL,KK,1DPX+21*AN*NDV
C IF (LIN.NE.1) GO TO 41
C
C LINEAR LINE RECURRING COST FUNCTION
C
C BN=EDIT/RECRNLN(KR,IL,KK,1DPX+11)
C LINKLN(IL,2) = RECRNLN(KR,IL,KK,1DPX+21*AN*NDV+12,
C GO TO 32
C 41 CONTINUE
C
C NONLINEAR LINE RECURRING FUNCTION
C
C DO 10 NON=18
C NON2=2*NON
C 10 CONTINUE
C
C COST=RECRNLN(KR,IL,KK,1DPX+NON2)
C DT=EDIT(KR,IL,KK,1DPX+NON1)
C IF (EDIT.GT.0) GO TO 51
C LINKLN(IL,2) = COST*EDIT*NDV+12+LINKLN(IL,2)
C 51 CONTINUE
C
C DST=EDIT-NT
C 10 CONTINUE
C 32 CONTINUE
C 47 CONTINUE
C 98 KCMH=1
C 99 KAP=1
C 100 RETURN
C 101 END
**Routine RHOFUN**

**Purpose:**
- Calculate Line Utilization
- Routine to obtain the highest line type and line configuration for a line utilizing a switcher.

**Parameters:**
- T1 = Line to Switcher Traffic
- T2 = Switcher to Line Traffic
- LNLMT = Highest Line Type
- LINEQU = Line Configuration
- N1 = EC
- N2, N3 = EC
- N4 = NC

**Variables:**
- CAP = Capability of Switcher
- LINEO = Line Configuration
- LHOLIN = Line Holiness
- XSA = AMSL
- XSA1 = ASUM
- XSA2 = ASUM
- XSA3 = AMSL
- XSA4 = AMSL
- XSA5 = AMSL
- XSA6 = AMSL
- XSA7 = AMSL
- XSA8 = AMSL

**Variables Initialization:**
- COMMON/LINCHR/ LNLMT(NP) = LINCAP(NP) * UTILIZ(NP)
- /SUM/ ASUM(4) = ASUM
- /SUM/ ASUM(5) = ASUM
- /SUM/ ASUM(6) = ASUM
- /SUM/ ASUM(7) = ASUM

**Routine Execution:**
- Calculate CAP using LINEOU(N) and LINCAP(N)
- Continue if AMSL is greater than 0
- Calculate XSA using AMSL

**End Routine:**
- RETURN
SUBROUTINE LINNUM(T1,T2,LINEQU,LNLMT,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/LINCHRT/ LINMIX(NPS),LINCAP(NPS),UTILIZ(NPS)
/CONST/ N1,N2,N3,N4,N7,MCITY
/MSLA/ AMGL(7)
INTEGER TRAF
DIMENSION LINEQU(1),RHLIN(3)
TRAF=T1+T2
DO 1 I=1,N3
LINEQU(I)=0
1 CONTINUE
LNLMT=0
CALL REFER
LNLMT=LNLMT+1
CALL REFER

C SET UP INITIAL LINE CONFIGURATION
C
IF(JFLAG .EQ. 1) GOTO 10
3 CONTINUE
LINEQU(LNLMT)=LINEQU(LNLMT)+1
LCAP=LINCAP(LNLMT)+UTILIZ(LNLMT)
IF(TRAF.LT.LCAP) GOTO 7
TRAF=TRAF-LCAP
CALL REFER
GOTO 3
10 CONTINUE
LINEQU(LNLMT)=TRAF/(LINCAP(LNLMT)+UTILIZ(LNLMT))
7 CONTINUE
70 CONTINUE
CALL RHOFUN(T1,T2,LINEQU,LNLMT,RHLIN,RHO)
IF(RHO .LT. UTILIZ(LNLMT)) GOTO 150
IF(LNLMT NE.N3) GOTO 72
IF(JFLAG .NE.1) GOTO 73
LINEQU(N3)=LINEQU(N3)+1
GOTO 70
73 CONTINUE
DO 2 N=1,N3
IF(LINEQU(N) .NE. 0) GOTO 20
2 CONTINUE
20 CONTINUE
NL=N
IF(NL.EQ.N3) GOTO 74
LINEQU(NL)=0
22 CONTINUE
NL=NL+1
55 IF(LINMIX(NL)=0) GOTO 22
LINEQU(NL)=LINEQU(NL)+1

77-53, Vol. IV

A-44
57 74 GOTO 70
58 74 CONTINUE
59 LINEGU(1)=1
60 GOTO 70
61 72 CONTINUE
62 LINEGU(LNLMT(1))=0
63 LNLMT=LNLMT+1
64 IF(LINMIX(LNLMT),FG.0) GOTO 79
65 LINEGU(LNLMT(1))=1
66 GOTO 70
67 150 CONTINUE
68 LNLMT=LNLMT
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 STACOM.PACK/0777
51928*STACOM(1).PACK/0777

1     COMPILER (FLD=ABS)
2     SUBROUTINE PACK(I,K,L,IA)
3
4     ****************************************
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     ****************************************
10     DIMENSION IA(1)
11     IQ=(I-1)/4 THE WORD LOCATION
12     IR=I-IA+4 THE QUARTER CONCERNED
13     IQ=IA+1
14     IS=(IR-1)*4
15     IF(L.EQ.1) GOTO 10
16     C
17     C RETRIEVE IT(9 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 IT(9 BITS) BEGINNING AT IS-TH BIT OF THE IQ-TH WORD
23     C
24     10 CONTINUE
25     FLD(15..9,IA(IQ))=K
26     RETURN
27     END

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

PARAMETER NP1=130; NPC=360
PARAMETER NP6=(NPC*NPC/2-1)/4+1
PARAMETER NP7=4
COMMON /REF/REF(NPC),TRAF(NP1,2,NP7),
* DSNCE(NP6),MAPADR(NP1)

INTEGER DSNCE
DST=0.
IF(I.EQ.J) RETURN
II=MAPADR(I) GACTUAL CITY INDEX
JJ=MAPADR(J)
IF(I.EQ.JJ) RETURN
II=LINK(IJ)
CALL PACK(IJL,IDIST,2,DSTNCF)
DIST=IDIST
IF(DIST GT 511) RETURN
DIST=COVR(IJL) RETURN

END
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 ********************************************************
  C
  PARAMETER NP1=130+NPC=360
  PARAMETER NP6=(NPC+NPC/2-NPC+1)/4+1
  PARAMETER NP7=4*NPC=4
  COMMON /CONST/ N1*N2*N3*N4*N5*N7*NCY
  1 /REF/REF(NPC)*TRAFF(NP1+2,NP7),REF(NPC)*MAPADR(NP1)
  INTENGER DSCORE
  LINK=0
  IF(J.GT.NCITY*OR.K.GT.NCITY*OR.K.FLJ) RETURN
  JJ=J
  KK=K
  IF(J.LT.K) GOTO 1
  JJ=K
  KK=J
  1 CONTINUE
  LINK=(J-J-1)*NCY + KK = IREF(JJ)
  RETURN
END
FUNCTION RECOVR(I)

C RETRIEVE OVERFLOW DISTANCE DATA FROM IVRD

C

PARAMETER NPC=360, NPO=10*NPC
COMMON /OVR/ IVRD(NP+2)*IOVER
DO 10 I=1,IOVER
10 IF(I.EQ. IVRD(N)) GOTO 20
CONTINUE
WRITE(6,99) I
99 FORMAT('NO OVRFLOW DATA HAS BEEN FOUND FOR LOCAL INDEX', I2)
STOP
20 CONTINUE
RECOVR=IVRD(N)
RETURN
END
SUBROUTINE PLOTPT(L1,L3)

***********

C SUBROUTINE FOR MOVING CANON PEN WITH OR WITHOUT PEN DOWN
6 C L3=1 FOR MOVING WITH PEN DOWN
7 C =2 FOR MOVING WITH PEN UP
8 C =3 FOR CLOSING THE PLOTTING
C
9 C
C
10 PARAMETER NPC=360
11 COMMON/VH/VERT(NPC),HORZ(NPC)
12 DIMENSION BUF(1000)
13 DATA IP/0/ FOR PLOTS CALL
14 DATA X/1.2566/
15 IF(IP .NE. 0) GOTO 50
16 CALL PLOTS
17 50 CONTINUE
18 IF(L3 .EQ. 3) GOTO 100 PLOTTING IS TO BE CLOSED
19 AV=VERT(L1)
20 AH=HORZ(L1)
21 BV=AV*COS(X)+AH*SIN(X)
22 BH=AV*SIN(X)-AH*COS(X)
23 IF(BH .LT. 0.0000001) GOTO 80
24 BV=(BV-5500)/301.
25 IF(L3 .EQ. 2) GOTO 80
26 CALL SYMBOL(BH,BV,0.0254,0.0254,2) OPEN IS DOWN
27 80 CONTINUE
28 CALL PLOT(RH,RV) OPEN IS UP
29 IP=1 PLOTS CALL IS NOT NEEDED ANY MORE
30 RETURN
31 100 CONTINUE
32 CALL PLOT(10,0.0,999)
33 RETURN
34 END

GPRT L_RSPNSE/0777
SUBROUTINE RSPNSE(T1,T2,LI1NTYP,N,ICK)

***************
C CALCULATE MEAN RESPONSE TIME FOR THE PROPOSED MULTIDROP LINE

C WAIT(6)=TMIZED DELAYS DUE TO
C 1=WAIT FOR POLLING \ 2=WAIT FOR I/O \ 3=INPUT XMT TIME
C 4=CPU TURNAROUND \ 5=OUTPUT QUEUE \ 6=OUTPUT XMT TIME

PARAMETER NPS=4
COMMON/RSP/ RHOLIN(4),RSPTIM
COMMON/XMT/ TIMXMT(7,NPS), WATT(6)
COMMON/INTERM,TIMXMT(7,NPS)
COMMON/CONT/N1,N2,N3,N4,N7,N7

DIMENSION LDUMMY(NPS)

DO 10 N=1,N3
LDUMMY(N)=0

10 CONTINUE
LDUMMY(LI1NTYP)=1
IK=0

CALL RHOFUN(T1,T2,LDUMMY,LI1NTYP,RHOLIN,PHN)

RHOLIN(4)=1.-RHO

IF(RHOLIN(4) .LE. 0.) RETURN

WAIT(1)=(TIMXMT(1,LI1NTYP)*TIMXMT(2,LI1NTYP))*(M-1)/2

WAIT(2)=((1.-RHOLIN(4))*TIMXMT(7,LI1NTYP))/((1.-RHOLIN(1))-RHOLIN(2))*RHOLIN(4))

WAIT(3)=TIMXMT(3,LI1NTYP)

WAIT(4)=((1.-RHOLIN(4))*TIMXMT(7,LI1NTYP))/((1.-RHOLIN(1))-RHOLIN(2))*RHOLIN(4))

WAIT(5)=TIMXMT(5,LI1NTYP)

RSPTIM=0.

DO 15 J=1,6
RSPTIM=WAIT(J)+RSPTIM

15 CONTINUE

IF(RSPTIM .GT. TIMREQ) RETURN

IK=1
RSPTIM=0.
RETURN
END
### APPENDIX B

#### GLOSSARY

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</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>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<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>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UNIVAC</td>
<td>UNIVersal Automatic Computer, a computer trade name by Sperry Rand Corporation</td>
</tr>
<tr>
<td>Vertical Horizontal (V-H) Coordinates</td>
<td>A pair of numbers which are designated by AT&amp;T for cities and used for the purpose of calculating distance between any two cities</td>
</tr>
</tbody>
</table>