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


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

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

### REFERENCES .......................................................... 4-1

### APPENDIXES

A STACOM PROGRAM LISTING .......................................... A-1
B GLOSSARY .............................................................. B-1

### 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
# Tables

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

INTRODUCTION

1.1 PURPOSE AND SCOPE

The STACOM (STATE Criminal Justice COMMunication) network topology program is a software tool which has been developed and utilized during the STACOM project. This Software Use'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$,G9PLTF will direct the punch card images to a CalComp plotter designated with G9PLTF.

1.2.4.2 Starting a Run.

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

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

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

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

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

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

1.2.5 Aborting and Recovering a Run

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

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

2.1.6 The STACOM Program and its Purposes

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

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

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

2.1.7 Functions Performed by the STACOM Program

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

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

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

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

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

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

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

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

2.2 MAIN FEATURES

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

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

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

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

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

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

2.2.1 Structure

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

---

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

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

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

$$
\begin{array}{c}
IA \\
f_1 & f_2 & f_3 & f_4 & f_5
\end{array}
$$

Let $c(f_i)$ = content of $i$-th field in a storage cell $IA$, where $IA$ 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 $IA$, $IB$, $IC$, $IP$, and $IP$ which are internal cardinal numbers for system terminations $A$, $B$, $C$, $D$, and $E$.

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

Figure 2-5. A Tree with $A$ as its Root
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 IB. Its third field has a pointer pointing to ID, and its fourth field a pointer pointing to IB.

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

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

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

(1) MAIN Routine

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

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

(2) RGNNET Routine

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

(3) IRNOP Routine

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

(4) LINNUM Routine

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

(5) RHOFUN Routine

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

(6) ICOSTJ Routine

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

(7) DIST Routine

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

(8) PACK Routine

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

(9) RECOVR Routine

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

(10) LINK Routine

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

(11) PLOTPT Routine

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

2.2.2 Response Time Algorithm -- RSPNSE Routine

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

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

2.2.3 Flexibility

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

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

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

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

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

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

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

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

2.2.4 Programming Language
The STACOM program is implemented with the FORTRAN V language of UNIVAC systems, compiled with the EXEC-8 FORTRAN processor, and mapped by its MAP processor.
Detailed features of FORTRAN V programming language are described in Reference 2.

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

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

As an example, the following is a complete list of EXEC-8 control statements which need to be prepared or typed in after the run card for properly executing the STACOM program.
The @SYM,P command directs the resulting plot card images to a 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.
2.3.1.17 **Constraints on Multidropped Lines and Average System Response Time.** The user can impose a constraint on the number of terminals allowed on a multidrop line either by limiting the number of terminals on a multidrop line, or by setting up a maximum average response time limit to the multidrop line or both.

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

2.4 **PROCESSING LOGIC**

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

2.4.1 **Traffic Calculation**

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

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

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

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

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>V</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin</td>
<td>9005 3996</td>
</tr>
<tr>
<td>Dallas</td>
<td>8436 4034</td>
</tr>
<tr>
<td>Difference</td>
<td>569 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
   \[ L1 = \left\lfloor \frac{(L-1)}{4} \right\rfloor + 1 \]
   \[ S1 = (L-1) \mod 4 + 1 \]
where \( \lfloor x \rfloor \) = the integer part of \( x \) and

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

(5) Store \( D_1 \) in segment \( S_1 \) of entry \( L_1 \) of table 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 \( L \), \( L_1 \) and \( S_1 \) as described above.

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

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

### 2.4.3 Formation of Regions

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

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

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

with

\[
TPR_1 = \sum_{1 \leq i \leq N_1} (\text{TRAFIT}(i) + \text{TRAFDN}(i))
\]

where

- \( I \) = the set of system terminations in preloaded regions which do not allow other system terminations to be inserted to them
- \( ANR_1 = NR_1 - \) [number of preloaded regions which do not allow any insertions]
Figure 2-8. Flow Chart for Formations of Regions
When the number of regions is 1, all of the system terminations are assigned to the region and no other region formation process is performed. Otherwise, the program starts assigning system terminations to regions (in a cardinal order) which allow their entries.

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

2.4.3.1 Assigning System Terminations to a Region without Preloading. When a region NREG is not preloaded with any system termination, processing continues with the finding of the farthest unassigned system termination (NS1) from the system centroid (N^SGC1). 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:

\[
TPR1 = TPR1 - TPFS
\]
\[
TPR = TPR1/(ANR1 - 1.)
\]
\[
TPRL = 0.9 * TPR
\]

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

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

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

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

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

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

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

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

2.4.4 Selection of Regional Switchers

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

The functional flow chart of RSC selection is depicted in Figure 2-10. Processing begins with assigning 1012 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:

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

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

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

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

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=1}^{NMBR} (TRAFFIC(i) \times TRAFFIC(i))^2 \times \text{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 NMBR times, the resulting NRSC is the index for the selected RSC and 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 RHO, at a value less or equal to a preselected number, say, 0.7.

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

The derivation of line configuration LDUMMY by subroutine LINNUM and the associated cost, COST, deserves more explanation. The program assumes that the duplexing mode for all line types under consideration to be half-duplexed. Therefore, subroutine LINNUM will sum up TRFIN and TRFOUT and find an applicable line with the least capacity which assures less than 0.7 of utilization. When the highest capacity line cannot handle the traffic, the routine will try to add one additional line with least capacity until the constraint of 0.7 utilization factor is satisfied. With line configuration LDUMMY obtained, calculation of cost,
START

K = 1

TRFIN = TRAFIT(K)
TRFOUT = TRAFDN(K)
DSTN = DISTK, NN1)

1. DETERMINE LINE
   CONFIGURATION
   LDUMMY
2. CALCULATE LINE
   UTILIZATION RHO
   (CALL LNUM)

INCREMENT
K BY 1

CALL (COSTJ)

CALCULATE COST,
COST, FOR LDUMMY

CALCULATE
RESPONSE TIME
RSPTIM

STORE ALL RELEVANT DATA FOR
LATER PRINTOUT

N

IS
K = NOREGN?

Y

TERMINATE

Figure 2-11. Flow Chart for Regional Star
Network Formation
COST, for the direct link between system termination K and the RSC is performed by ICOSTJ. The routine ICOSTJ calculates all of the related installation and annual recurring costs for lines and other chargeable items. All of these itemized costs are then summarized as COST. Cost calculations are performed on the basis of the rate structures applicable to system terminations at both ends.

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

2.4.6 Optimization of Regional Networks

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

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

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

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

The following section describes the logic implemented in this program.

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

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

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

\( L_i = j \) for which the integration of \( K \) and \( L \) sub-networks provides \( S_K \) 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 \( K_i \), starts evaluating possible cost saving by eliminating central link \( K \) and
Figure 2-14. Flow Chart for Subroutine ESSWIL
reconnecting the rest of sub-network K to sub-network L through system terminations M of L and KI of K as shown in Figure 2-15.

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

For each given set of K and L, the program tests whether the sum, NT, of numbers of system terminations for both sub-networks exceeds the value of NTERMS which constrains the number of system terminations on a multidrop line. If this is true, it skips the process of calling 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.

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

Figure 2-15. Relationship among K, L, KI, and M Parameters
The program first calculates the distance DREF between system termination K and the region switching center for each K, and then the DTRY which is half of the distance between system terminations KI and M for each combination.

If DTRY is greater than DREF, 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 LINNUFM 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.

![Flow Chart for Subroutine RSPNSE](image)

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

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

2.4.7 **Formation of an Interregional Network**

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

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

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

2.4.8 **Optimization of an Interregional Network**

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

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

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

This cycle of searching for the best cost saving and updating the network repeats continuously until cost savings can no longer be realized.
Figure 2-18. Flow Chart for Intraregional Line Selection
2.4.8.1 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 2-19. Basic Topology of Line Elimination
Figure 2-20. Flow Chart for Interregion Network Optimization
If all 1-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 preloading 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 database; 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 drdp. Partial sums for the line are also printed. Finally, total installation and annual recurring costs for each chargeable item and for the overall star network are printed.
2.5.1.8 Final Optimized Network and its Costs. After performing optimizations on the star network, the program prints out descriptions for each multidrop line in the final optimized network. The following list shows the data items which may be printed.

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

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

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

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

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

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

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

2.5.2 CalComp Plot

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

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

2.6.1 Hardware

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

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

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

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

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

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

3.1 INTRODUCTION

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

3.2 ENVIRONMENT

3.2.1 Hardware

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

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

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

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

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

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

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

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

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

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

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

3.2.3 Functional Limitations

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

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

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

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

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

3.3 RUN DESCRIPTION

3.3.1 Initialization and Setup

When the STACOM program is executed from an 80-character/line demand terminal, an alternate file, 100, to be used as a printer output file, must be defined. Otherwise, all printout data will be directed to the terminal which will produce interleaving output. The statement #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>(3I1X,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,I5,2X,I5)</td>
</tr>
<tr>
<td>5</td>
<td>ID, name, city index, additional no. of terminals and traffic to/from each data base for each system termination.</td>
<td>INDXPT(N1), NAMEST(H,N1), IADD(N1)</td>
<td>a. N1 if N7\leq2</td>
<td>(A4,1X,3A6,A4, 12,14,4F10.2/ 6F10.2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAPADR(N1), TRAFD(N1,2,N7)</td>
<td>b. 2N1 if N7\geq2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>No. of rate structures</td>
<td>N2</td>
<td>1</td>
<td>(I3)</td>
</tr>
<tr>
<td>7</td>
<td>Rate application matrix</td>
<td>IRATEJ(N2,N2)</td>
<td>N2</td>
<td>(10I2)</td>
</tr>
<tr>
<td>8</td>
<td>Traffic density index and applicable rate structure for each city</td>
<td>IRAND(NCITY,2)</td>
<td>[NCITY/40]</td>
<td>(80I1)</td>
</tr>
<tr>
<td>9</td>
<td>No. of applicable line types</td>
<td>N3</td>
<td>1</td>
<td>(I3)</td>
</tr>
</tbody>
</table>
### Table 3-1. Formats for Input Data (Continuation 1)

<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) a. (4F9.2/I1/10F8.3) b. 2xN2(2+N3x3x2) b. (4F9.2/I1/10F8.3/3x2) if non linear 10F8.3</td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>Action indices for regions</td>
<td>NRFG,NCODE for IACTN(NR1,2)</td>
<td>Variable</td>
<td>(2I2)</td>
</tr>
</tbody>
</table>
Table 3-1. Formats for Input Data  
(Continuation 2)

<table>
<thead>
<tr>
<th>Item No.</th>
<th>Item Description</th>
<th>Names of Internal Variables/Arrays</th>
<th>Number of Cards Needed</th>
<th>Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.</td>
<td>No. of polling characters no. of NAK characters, no. of polling overhead characters, no of NAK overhead characters, message overhead characters, Modem turnaround time, and other delay for each line type</td>
<td>NPL(N3), NAK(N3), NPLOH(N3), NAKOH(N3), MOH(N3), TAMOH(N3), TAPD(N3)</td>
<td>N3</td>
<td>(SI4,2F7.5)</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), RATIO1(NTYP,2), RATIO(NTYP,2)</td>
<td>NTYP</td>
<td>(A6,2(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>(I1,1X,A4,A5)</td>
</tr>
<tr>
<td>21.</td>
<td>System Center 1</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>MPLOT</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 $^6$E$^6$ after interrupting the line communication by pressing the BREAK key will terminate a program execution at any time. On the other hand, the EXEC-$^3$ 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.
77-53, Vol. IV

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 JJI, JG3YL, 51928, 20, 90/1000
@ASG, UP 100
@SYM, P PUNCH$, G9PLTF
@LOT 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 JJI, identifies its account number as JG3YL, assigns project ID as 51928, requests a maximum of 20 minutes of SUP-time and finally asks for a limit of 90 printer pages and 1,000 punch cards. The limits on SUP-time, number of printer pages and number of punch cards deserve some attention when making a run. If there is an underestimate in any of these three limits, the run may abort due to insufficient resource assignment.

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

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

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

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

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

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

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

3.4.2 Input

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>State</th>
<th>City</th>
<th>Code</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ANDERSON</td>
<td>P A L E S T I N E</td>
<td>8550</td>
<td>3750</td>
</tr>
<tr>
<td>2</td>
<td>ANDREWS</td>
<td>FALLS</td>
<td>8857</td>
<td>4495</td>
</tr>
<tr>
<td>3</td>
<td>ANGELINA</td>
<td>BEEVILLE</td>
<td>8575</td>
<td>3671</td>
</tr>
<tr>
<td>4</td>
<td>ARENAS</td>
<td>BELTON</td>
<td>8405</td>
<td>3694</td>
</tr>
<tr>
<td>5</td>
<td>ATASCOSA</td>
<td>BELFORE</td>
<td>8343</td>
<td>3436</td>
</tr>
<tr>
<td>6</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>7</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>8</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>9</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>10</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>11</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>12</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>13</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>14</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>15</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>16</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>17</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>18</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>19</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>20</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>21</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>22</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>23</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>24</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>25</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>26</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>27</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>28</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>29</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>30</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>31</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>32</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>33</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>34</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>35</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>36</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>37</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>38</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>39</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>40</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>41</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>42</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>43</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>44</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>45</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>46</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>47</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>48</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>49</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>50</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>51</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>52</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>53</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>54</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>55</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>56</td>
<td>BELL</td>
<td>BELL</td>
<td>8218</td>
<td>3665</td>
</tr>
<tr>
<td>57</td>
<td>COLUMBUS</td>
<td>COLORADO</td>
<td>9032</td>
<td>3749</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>GOWANCHE</td>
<td>COMANCHE</td>
<td>8775</td>
<td>4275</td>
</tr>
<tr>
<td>60</td>
<td>GAINESVILLE</td>
<td>COOKE</td>
<td>8269</td>
<td>4162</td>
</tr>
<tr>
<td>61</td>
<td>COPPERAS COVE</td>
<td>CORYELL</td>
<td>8844</td>
<td>4099</td>
</tr>
<tr>
<td>62</td>
<td>GATESVILLE</td>
<td>CRANE</td>
<td>8771</td>
<td>4080</td>
</tr>
<tr>
<td>63</td>
<td>CRANE</td>
<td>CRANE</td>
<td>9073</td>
<td>4096</td>
</tr>
<tr>
<td>64</td>
<td>OZONA</td>
<td>CROCKETT</td>
<td>9144</td>
<td>4692</td>
</tr>
<tr>
<td>65</td>
<td>DALHART</td>
<td>DALLAS</td>
<td>8129</td>
<td>4269</td>
</tr>
<tr>
<td>66</td>
<td>ADDISON</td>
<td>DALLAS</td>
<td>8494</td>
<td>4082</td>
</tr>
<tr>
<td>67</td>
<td>CEDAR HILL</td>
<td>DALLAS</td>
<td>8495</td>
<td>4087</td>
</tr>
<tr>
<td>68</td>
<td>DALLAS</td>
<td>DALLAS</td>
<td>8436</td>
<td>4034</td>
</tr>
<tr>
<td>69</td>
<td>DESOTO</td>
<td>DALLAS</td>
<td>8478</td>
<td>4030</td>
</tr>
<tr>
<td>70</td>
<td>DUNCANVILLE</td>
<td>DALLAS</td>
<td>8469</td>
<td>4044</td>
</tr>
<tr>
<td>71</td>
<td>FARMERS BRANCH</td>
<td>DALLAS</td>
<td>8414</td>
<td>4062</td>
</tr>
<tr>
<td>72</td>
<td>GARLAND</td>
<td>DALLAS</td>
<td>8400</td>
<td>4018</td>
</tr>
<tr>
<td>73</td>
<td>GRAND PRAIRIE</td>
<td>DALLAS</td>
<td>8458</td>
<td>4066</td>
</tr>
<tr>
<td>74</td>
<td>HIGHLAND PARK</td>
<td>DALLAS</td>
<td>8436</td>
<td>4034</td>
</tr>
<tr>
<td>75</td>
<td>IRVING</td>
<td>DALLAS</td>
<td>8440</td>
<td>4064</td>
</tr>
<tr>
<td>76</td>
<td>LANCASTER</td>
<td>DALLAS</td>
<td>8470</td>
<td>4013</td>
</tr>
<tr>
<td>77</td>
<td>MESQUITE</td>
<td>DALLAS</td>
<td>8426</td>
<td>4000</td>
</tr>
<tr>
<td>78</td>
<td>RICHARDSON</td>
<td>DALLAS</td>
<td>8369</td>
<td>4035</td>
</tr>
<tr>
<td>79</td>
<td>SEAGOVILLE</td>
<td>DALLAS</td>
<td>8447</td>
<td>3986</td>
</tr>
<tr>
<td>80</td>
<td>UNIVERSITY PARK</td>
<td>DALLAS</td>
<td>8436</td>
<td>4034</td>
</tr>
<tr>
<td>81</td>
<td>LAVEZA</td>
<td>DAWSON</td>
<td>8779</td>
<td>4094</td>
</tr>
<tr>
<td>82</td>
<td>HEREFORD</td>
<td>DEAF SMITH</td>
<td>8378</td>
<td>4143</td>
</tr>
<tr>
<td>83</td>
<td>DENTON</td>
<td>DENTON</td>
<td>8372</td>
<td>4127</td>
</tr>
<tr>
<td>84</td>
<td>LEXINGTON</td>
<td>DENTON</td>
<td>8398</td>
<td>4099</td>
</tr>
<tr>
<td>85</td>
<td>CUFRO</td>
<td>DEWITT</td>
<td>9209</td>
<td>3923</td>
</tr>
<tr>
<td>86</td>
<td>SPIRIT</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>3988</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>4093</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>5659</td>
</tr>
<tr>
<td>93</td>
<td>STEPHENVILLE</td>
<td>ERATH</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>3996</td>
</tr>
<tr>
<td>96</td>
<td>ROY</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>4092</td>
</tr>
<tr>
<td>98</td>
<td>FAIRFIELD</td>
<td>FREESTONE</td>
<td>8602</td>
<td>3830</td>
</tr>
<tr>
<td>99</td>
<td>PEARL</td>
<td>FRIO</td>
<td>9374</td>
<td>4120</td>
</tr>
<tr>
<td>100</td>
<td>RICHMOND</td>
<td>Ft BEND</td>
<td>9009</td>
<td>3598</td>
</tr>
<tr>
<td>101</td>
<td>ROSENBerg</td>
<td>Ft BEND</td>
<td>9009</td>
<td>3598</td>
</tr>
<tr>
<td>102</td>
<td>SEMINOLE</td>
<td>GAINES</td>
<td>8922</td>
<td>4940</td>
</tr>
<tr>
<td>103</td>
<td>FRIENDSWOOD</td>
<td>GALVESTON</td>
<td>8960</td>
<td>3430</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>8999</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>3463</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>9313</td>
<td>3687</td>
</tr>
<tr>
<td>112</td>
<td>GONZALEZ</td>
<td>GONZALEZ</td>
<td>9137</td>
<td>3840</td>
</tr>
<tr>
<td>113</td>
<td>PANAMA</td>
<td>GRAY</td>
<td>8148</td>
<td>4952</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>114</td>
<td>DENISON</td>
<td>GRAYSON</td>
<td>825</td>
<td>4066</td>
</tr>
<tr>
<td>115</td>
<td>SHERMAN</td>
<td>GRAYSON</td>
<td>8253</td>
<td>4072</td>
</tr>
<tr>
<td>116</td>
<td>GLANEMONT</td>
<td>GREGG</td>
<td>8394</td>
<td>3649</td>
</tr>
<tr>
<td>117</td>
<td>KILGORE</td>
<td>GREGG</td>
<td>8379</td>
<td>3674</td>
</tr>
<tr>
<td>118</td>
<td>LONGVIEW</td>
<td>GREGG</td>
<td>8348</td>
<td>3660</td>
</tr>
<tr>
<td>119</td>
<td>NAVASOTA</td>
<td>GRIMES</td>
<td>8665</td>
<td>3715</td>
</tr>
<tr>
<td>120</td>
<td>SEGUIN</td>
<td>GUADALUPE</td>
<td>9161</td>
<td>3981</td>
</tr>
<tr>
<td>121</td>
<td>PLAINVIEW</td>
<td>HAILE</td>
<td>8465</td>
<td>3991</td>
</tr>
<tr>
<td>122</td>
<td>HAMILTON</td>
<td>HAMILTON</td>
<td>8744</td>
<td>4177</td>
</tr>
<tr>
<td>123</td>
<td>SPARKMAN</td>
<td>HAMISFORD</td>
<td>8026</td>
<td>5037</td>
</tr>
<tr>
<td>124</td>
<td>QUANAH</td>
<td>HINDEN</td>
<td>8324</td>
<td>4654</td>
</tr>
<tr>
<td>125</td>
<td>KOUNTZE</td>
<td>HARRIN</td>
<td>8735</td>
<td>3405</td>
</tr>
<tr>
<td>126</td>
<td>SILSBEE</td>
<td>HARRIS</td>
<td>8770</td>
<td>3380</td>
</tr>
<tr>
<td>127</td>
<td>BAYTOWN</td>
<td>HARRIS</td>
<td>8916</td>
<td>3466</td>
</tr>
<tr>
<td>128</td>
<td>BELLAIRE</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>129</td>
<td>DEEP PARK</td>
<td>HARRIS</td>
<td>8929</td>
<td>3491</td>
</tr>
<tr>
<td>130</td>
<td>GALENA PARK</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>131</td>
<td>HOUSTON</td>
<td>HARRIS</td>
<td>8881</td>
<td>3540</td>
</tr>
<tr>
<td>132</td>
<td>HUMBLE</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>133</td>
<td>JARINTO CITY</td>
<td>HARRIS</td>
<td>8925</td>
<td>3581</td>
</tr>
<tr>
<td>134</td>
<td>JERSEY VILLAGE</td>
<td>HARRIS</td>
<td>8965</td>
<td>3624</td>
</tr>
<tr>
<td>135</td>
<td>KATY</td>
<td>HARRIS</td>
<td>8829</td>
<td>3470</td>
</tr>
<tr>
<td>136</td>
<td>LA PORTE</td>
<td>HARRIS</td>
<td>8938</td>
<td>3436</td>
</tr>
<tr>
<td>137</td>
<td>PASADENA</td>
<td>HARRIS</td>
<td>8945</td>
<td>3560</td>
</tr>
<tr>
<td>138</td>
<td>SEABROOK</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>139</td>
<td>SOUTH HOUSTON</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>140</td>
<td>SOUTHSIDE PLACE</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>141</td>
<td>SPRING VALLEY</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>142</td>
<td>THOMAS</td>
<td>HARRIS</td>
<td>8889</td>
<td>3509</td>
</tr>
<tr>
<td>143</td>
<td>VILLAGE</td>
<td>HARRIS</td>
<td>8934</td>
<td>3536</td>
</tr>
<tr>
<td>144</td>
<td>WEBSTER</td>
<td>HARRIS</td>
<td>8967</td>
<td>3664</td>
</tr>
<tr>
<td>145</td>
<td>WEST UNIV PL</td>
<td>HARRIS</td>
<td>8938</td>
<td>3536</td>
</tr>
<tr>
<td>146</td>
<td>MARSHALL</td>
<td>HARRISON</td>
<td>8311</td>
<td>3562</td>
</tr>
<tr>
<td>147</td>
<td>HASSELL</td>
<td>HASKELL</td>
<td>8554</td>
<td>4567</td>
</tr>
<tr>
<td>148</td>
<td>SAN MARCOS</td>
<td>HAYS</td>
<td>9006</td>
<td>4001</td>
</tr>
<tr>
<td>149</td>
<td>CANADIAN</td>
<td>HEMPHILL</td>
<td>8036</td>
<td>4882</td>
</tr>
<tr>
<td>150</td>
<td>ATHENS</td>
<td>HENDERSON</td>
<td>8444</td>
<td>3826</td>
</tr>
<tr>
<td>151</td>
<td>DONNA</td>
<td>HIDALGO</td>
<td>8849</td>
<td>3724</td>
</tr>
<tr>
<td>152</td>
<td>EDINBURG</td>
<td>HIDALGO</td>
<td>8682</td>
<td>3754</td>
</tr>
<tr>
<td>153</td>
<td>Hidalgo</td>
<td>HIDALGO</td>
<td>8946</td>
<td>3764</td>
</tr>
<tr>
<td>154</td>
<td>MCALLEN</td>
<td>HIDALGO</td>
<td>8945</td>
<td>3704</td>
</tr>
<tr>
<td>155</td>
<td>MERCEDES</td>
<td>HIDALGO</td>
<td>8941</td>
<td>3781</td>
</tr>
<tr>
<td>156</td>
<td>MISSION</td>
<td>HIDALGO</td>
<td>8944</td>
<td>3758</td>
</tr>
<tr>
<td>157</td>
<td>PHARR</td>
<td>HIDALGO</td>
<td>8947</td>
<td>3716</td>
</tr>
<tr>
<td>158</td>
<td>WESLACO</td>
<td>HIDALGO</td>
<td>8947</td>
<td>3716</td>
</tr>
<tr>
<td>159</td>
<td>HILLSBORO</td>
<td>HILL</td>
<td>8612</td>
<td>4026</td>
</tr>
<tr>
<td>160</td>
<td>LEVELLAND</td>
<td>HOCKLEY</td>
<td>8629</td>
<td>5873</td>
</tr>
<tr>
<td>161</td>
<td>SULPHUR SPRINGS</td>
<td>HOPKINS</td>
<td>8241</td>
<td>3561</td>
</tr>
<tr>
<td>162</td>
<td>BIG SPRING</td>
<td>HOWARD</td>
<td>8847</td>
<td>4000</td>
</tr>
<tr>
<td>163</td>
<td>COMMERCE</td>
<td>HUNT</td>
<td>8240</td>
<td>3921</td>
</tr>
<tr>
<td>164</td>
<td>GLENVILLE</td>
<td>HUNT</td>
<td>8317</td>
<td>3949</td>
</tr>
<tr>
<td>165</td>
<td>BOISDORF</td>
<td>HUTCHINSON</td>
<td>8146</td>
<td>5033</td>
</tr>
<tr>
<td>166</td>
<td>JACKS</td>
<td>JACKSON</td>
<td>9186</td>
<td>3698</td>
</tr>
<tr>
<td>167</td>
<td>EDNA</td>
<td>JEFFERSON</td>
<td>8777</td>
<td>3444</td>
</tr>
<tr>
<td>168</td>
<td>BEAUMONT</td>
<td>JEFFERSON</td>
<td>879</td>
<td>3316</td>
</tr>
<tr>
<td>169</td>
<td>NENNERLAND</td>
<td>JEFFERSON</td>
<td>8406</td>
<td>3298</td>
</tr>
<tr>
<td>170</td>
<td>PORT ARTHUR</td>
<td>JEFFERSON</td>
<td>8406</td>
<td>3298</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run  
(Continuation 3)

<table>
<thead>
<tr>
<th></th>
<th>ALICE</th>
<th>JIM WELLS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BURLES0N</td>
<td>JOHNSON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>CLFRURNE</td>
<td>JOHNSON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>173</td>
<td>ANSON</td>
<td>JONES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>174</td>
<td>STAMFORD</td>
<td>JONES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>175</td>
<td>KARNES CITY</td>
<td>KARNES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>KAUFMAN</td>
<td>KAUFMAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>177</td>
<td>TERRELL</td>
<td>KAUFMAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>178</td>
<td>BOFRINE</td>
<td>KENDALL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>179</td>
<td>JAYTON</td>
<td>KENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>KERRVILLE</td>
<td>KERR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>181</td>
<td>JUNCTION</td>
<td>KIMBLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>KINGSVILLE</td>
<td>KLEBERG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>183</td>
<td>BENJAMIN</td>
<td>KNOX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>184</td>
<td>PARTS</td>
<td>LAMAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>185</td>
<td>LITTLEFIELD</td>
<td>LAMB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>186</td>
<td>OLTON</td>
<td>LAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>187</td>
<td>LAMpasas</td>
<td>LAMpasas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>188</td>
<td>HALLETTsville</td>
<td>LAVACA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>189</td>
<td>YoKum</td>
<td>LAVACA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>CEnTERVILLE</td>
<td>LEOEn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>CLEVELAND</td>
<td>LIBERTY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>LIBERTY</td>
<td>LIBERTY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>MEYLA</td>
<td>LIMESTONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>194</td>
<td>GEORGE WEST</td>
<td>LIVE OAK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>195</td>
<td>Librock</td>
<td>LUBBOck</td>
<td></td>
<td></td>
</tr>
<tr>
<td>196</td>
<td>SLATON</td>
<td>LUBBOck</td>
<td></td>
<td></td>
</tr>
<tr>
<td>197</td>
<td>TAHOKA</td>
<td>LYNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>198</td>
<td>MADISONVILLE</td>
<td>MADISON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>199</td>
<td>JEFFERSON</td>
<td>MARION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>BAY CITY</td>
<td>MARINA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>EAALE PASS</td>
<td>MAVERICK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>202</td>
<td>BRANDY</td>
<td>MCCULLOCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>BELLMEAD</td>
<td>MCLennAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>204</td>
<td>BEVERLY HILLS</td>
<td>MCLennAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>KNOX</td>
<td>MCLennAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>WOODWAY</td>
<td>MCLennAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>207</td>
<td>HONDO</td>
<td>MEDINA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>208</td>
<td>MENARD</td>
<td>MENARD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>209</td>
<td>MINDLAND</td>
<td>MIDLAND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>CAMPRON</td>
<td>MILAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>ROCKDALE</td>
<td>MILAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>COLORADO CITY</td>
<td>MITCHELL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>213</td>
<td>BOWIE</td>
<td>MONTAGUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>214</td>
<td>MONTAGUE</td>
<td>MONTAGUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>215</td>
<td>CONROE</td>
<td>MONTGOMRY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>216</td>
<td>OUNAS</td>
<td>MOORE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>217</td>
<td>DANGERFIELD</td>
<td>MORRIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>218</td>
<td>NACOGDOCHES</td>
<td>NACOGDOCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>219</td>
<td>CORSICANA</td>
<td>NAVARRO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>220</td>
<td>SWETTWARE</td>
<td>NOLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>221</td>
<td>CORPUS CHRISTI</td>
<td>NEUcES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>222</td>
<td>KOWSfOWN</td>
<td>NEUcES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>223</td>
<td>PEPPRYTON</td>
<td>OCHILTyREF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>VEGA</td>
<td>OLDHAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>225</td>
<td>MINEWELL</td>
<td>PARKER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>226</td>
<td>ORANGE</td>
<td>PARKER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>227</td>
<td>MINERAl WELfES</td>
<td>PARKER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run (Continuation 4)

<table>
<thead>
<tr>
<th>Location</th>
<th>County</th>
<th>Code</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carthage</td>
<td>Panola</td>
<td>8345</td>
<td>3564</td>
</tr>
<tr>
<td>Weatherford</td>
<td>Parker</td>
<td>8598</td>
<td>4206</td>
</tr>
<tr>
<td>Farwell</td>
<td>Parker</td>
<td>8503</td>
<td>5221</td>
</tr>
<tr>
<td>Giddings</td>
<td>Pecos</td>
<td>8432</td>
<td>5185</td>
</tr>
<tr>
<td>Fort Stockton</td>
<td>Potter</td>
<td>8266</td>
<td>5076</td>
</tr>
<tr>
<td>Canyon</td>
<td>Randall</td>
<td>8317</td>
<td>5075</td>
</tr>
<tr>
<td>Clarksville</td>
<td>Red River</td>
<td>8147</td>
<td>3809</td>
</tr>
<tr>
<td>Pecos</td>
<td>Reeves</td>
<td>9136</td>
<td>5101</td>
</tr>
<tr>
<td>Big Lake</td>
<td>Regan</td>
<td>9062</td>
<td>4793</td>
</tr>
<tr>
<td>Hearne</td>
<td>Robertson</td>
<td>8802</td>
<td>3846</td>
</tr>
<tr>
<td>Rockwall</td>
<td>Rockwall</td>
<td>8384</td>
<td>5099</td>
</tr>
<tr>
<td>Ballinger</td>
<td>Runnels</td>
<td>8855</td>
<td>4408</td>
</tr>
<tr>
<td>Henderson</td>
<td>Rusk</td>
<td>8420</td>
<td>3640</td>
</tr>
<tr>
<td>Aransas Pass</td>
<td>San Patco</td>
<td>9437</td>
<td>3700</td>
</tr>
<tr>
<td>Gregory</td>
<td>San Patco</td>
<td>9455</td>
<td>3731</td>
</tr>
<tr>
<td>Ingleside</td>
<td>San Patco</td>
<td>9447</td>
<td>3711</td>
</tr>
<tr>
<td>Portland</td>
<td>San Patco</td>
<td>9455</td>
<td>3731</td>
</tr>
<tr>
<td>Stony</td>
<td>San Patco</td>
<td>9436</td>
<td>3777</td>
</tr>
<tr>
<td>Eldorado</td>
<td>Schleicher</td>
<td>9076</td>
<td>4547</td>
</tr>
<tr>
<td>Snyder</td>
<td>Scurry</td>
<td>8718</td>
<td>4737</td>
</tr>
<tr>
<td>Stratford</td>
<td>Sherman</td>
<td>8049</td>
<td>4194</td>
</tr>
<tr>
<td>Tyler</td>
<td>Smith</td>
<td>8417</td>
<td>3764</td>
</tr>
<tr>
<td>Breckenridge</td>
<td>Stephens</td>
<td>8582</td>
<td>4304</td>
</tr>
<tr>
<td>Sterling City</td>
<td>Stephenville</td>
<td>8080</td>
<td>4646</td>
</tr>
<tr>
<td>Aspermont</td>
<td>St. New</td>
<td>8589</td>
<td>4650</td>
</tr>
<tr>
<td>Tulia</td>
<td>Swisher</td>
<td>8937</td>
<td>5016</td>
</tr>
<tr>
<td>Arlington</td>
<td>Tarrant</td>
<td>8472</td>
<td>4085</td>
</tr>
<tr>
<td>Bedford</td>
<td>Tarrant</td>
<td>8447</td>
<td>4092</td>
</tr>
<tr>
<td>Cleburne</td>
<td>Tarrant</td>
<td>8447</td>
<td>4117</td>
</tr>
<tr>
<td>Crowley</td>
<td>Tarrant</td>
<td>8518</td>
<td>4118</td>
</tr>
<tr>
<td>Eulens</td>
<td>Tarrant</td>
<td>8447</td>
<td>4092</td>
</tr>
<tr>
<td>Forrest Hill</td>
<td>Tarrant</td>
<td>8479</td>
<td>4122</td>
</tr>
<tr>
<td>Fort Worth</td>
<td>Tarrant</td>
<td>8479</td>
<td>4122</td>
</tr>
<tr>
<td>Grapevine</td>
<td>Tarrant</td>
<td>8425</td>
<td>4009</td>
</tr>
<tr>
<td>Halot City</td>
<td>Tarrant</td>
<td>8479</td>
<td>4122</td>
</tr>
<tr>
<td>Hurst</td>
<td>Tarrant</td>
<td>8447</td>
<td>4117</td>
</tr>
<tr>
<td>Lake Worth</td>
<td>Tarrant</td>
<td>8471</td>
<td>4158</td>
</tr>
<tr>
<td>No Richland Hls</td>
<td>Tarrant</td>
<td>8447</td>
<td>4117</td>
</tr>
<tr>
<td>Richland Hills</td>
<td>Tarrant</td>
<td>8479</td>
<td>4122</td>
</tr>
<tr>
<td>Southlake</td>
<td>Tarrant</td>
<td>8425</td>
<td>4094</td>
</tr>
<tr>
<td>White Settlement</td>
<td>Tarrant</td>
<td>8485</td>
<td>4153</td>
</tr>
<tr>
<td>Ayleen</td>
<td>Taylor</td>
<td>8668</td>
<td>4513</td>
</tr>
<tr>
<td>Sanderson</td>
<td>Terrell</td>
<td>9333</td>
<td>4816</td>
</tr>
<tr>
<td>Brownfield</td>
<td>Terry</td>
<td>8705</td>
<td>5007</td>
</tr>
<tr>
<td>Mt Pleasant</td>
<td>Titus</td>
<td>8234</td>
<td>3755</td>
</tr>
<tr>
<td>San Angelo</td>
<td>Tom Green</td>
<td>8944</td>
<td>4583</td>
</tr>
<tr>
<td>Austin</td>
<td>Travis</td>
<td>9005</td>
<td>3906</td>
</tr>
<tr>
<td>Gilmer</td>
<td>Upshur</td>
<td>8317</td>
<td>3716</td>
</tr>
<tr>
<td>Rankin</td>
<td>Upton</td>
<td>9089</td>
<td>4811</td>
</tr>
<tr>
<td>Uvalde</td>
<td>Uvalde</td>
<td>9357</td>
<td>4279</td>
</tr>
<tr>
<td>Del Rio</td>
<td>Val Verde</td>
<td>9399</td>
<td>4490</td>
</tr>
<tr>
<td>Canton</td>
<td>Van Zandt</td>
<td>8414</td>
<td>3858</td>
</tr>
<tr>
<td>Victoria</td>
<td>Victoria</td>
<td>9245</td>
<td>3748</td>
</tr>
<tr>
<td>Huntsville</td>
<td>Walker</td>
<td>875A</td>
<td>3652</td>
</tr>
<tr>
<td>Hempstead</td>
<td>Waller</td>
<td>8923</td>
<td>3691</td>
</tr>
<tr>
<td>Monahans</td>
<td>Ward</td>
<td>9066</td>
<td>5065</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>285</td>
<td>BRENHAM</td>
<td>WASHINGTON</td>
<td>8942</td>
</tr>
<tr>
<td>286</td>
<td>LAREDO</td>
<td>WASHINGTN</td>
<td>8972</td>
</tr>
<tr>
<td>287</td>
<td>PIERCE</td>
<td>WHARTON</td>
<td>9115</td>
</tr>
<tr>
<td>288</td>
<td>WHARTON</td>
<td>WHARTON</td>
<td>9078</td>
</tr>
<tr>
<td>289</td>
<td>SHARROCK</td>
<td>WHEELER</td>
<td>8170</td>
</tr>
<tr>
<td>290</td>
<td>BURKBURNETT</td>
<td>WICHITA</td>
<td>8204</td>
</tr>
<tr>
<td>291</td>
<td>WICHITA FALLS</td>
<td>WICHITA</td>
<td>8326</td>
</tr>
<tr>
<td>292</td>
<td>VERNON</td>
<td>WILBARGER</td>
<td>8326</td>
</tr>
<tr>
<td>293</td>
<td>RAYMONDVILLE</td>
<td>WILLACY</td>
<td>9768</td>
</tr>
<tr>
<td>294</td>
<td>FLORESVILLE</td>
<td>WILSON</td>
<td>9261</td>
</tr>
<tr>
<td>295</td>
<td>KERMIT</td>
<td>WINKLER</td>
<td>9024</td>
</tr>
<tr>
<td>296</td>
<td>DECATURE</td>
<td>WISE</td>
<td>8509</td>
</tr>
<tr>
<td>297</td>
<td>WINNSBORO</td>
<td>WOOD</td>
<td>8295</td>
</tr>
<tr>
<td>298</td>
<td>DENVER CITY</td>
<td>YOAKUM</td>
<td>8781</td>
</tr>
<tr>
<td>299</td>
<td>GRAHAM</td>
<td>YOUNG</td>
<td>8492</td>
</tr>
<tr>
<td>300</td>
<td>OLENE</td>
<td>YOUNG</td>
<td>8450</td>
</tr>
<tr>
<td>301</td>
<td>CRYSTAL CITY</td>
<td>ZAVALA</td>
<td>9466</td>
</tr>
<tr>
<td>302</td>
<td>ARANSAS PASS</td>
<td>ARANSAS</td>
<td>9437</td>
</tr>
<tr>
<td>303</td>
<td>PLEASANTON</td>
<td>ATASCOSA</td>
<td>9320</td>
</tr>
<tr>
<td>304</td>
<td>BANCERA</td>
<td>BANDERA</td>
<td>9255</td>
</tr>
<tr>
<td>305</td>
<td>BASTROP</td>
<td>BASTROP</td>
<td>9007</td>
</tr>
<tr>
<td>306</td>
<td>MARBLE FALLS</td>
<td>BURNET</td>
<td>8960</td>
</tr>
<tr>
<td>307</td>
<td>LOCKHART</td>
<td>CALDWILL</td>
<td>9077</td>
</tr>
<tr>
<td>308</td>
<td>LULING</td>
<td>CALDWILL</td>
<td>9117</td>
</tr>
<tr>
<td>309</td>
<td>BAIRD</td>
<td>CALLAHAN</td>
<td>868A</td>
</tr>
<tr>
<td>310</td>
<td>PITTSBURG</td>
<td>CAMP</td>
<td>8264</td>
</tr>
<tr>
<td>311</td>
<td>PAHANDLE</td>
<td>CARSON</td>
<td>8210</td>
</tr>
<tr>
<td>312</td>
<td>ATLANTA</td>
<td>CASS</td>
<td>8182</td>
</tr>
<tr>
<td>313</td>
<td>RUSK</td>
<td>CHEROKEE</td>
<td>8515</td>
</tr>
<tr>
<td>314</td>
<td>HENRIETTA</td>
<td>CLAY</td>
<td>8326</td>
</tr>
<tr>
<td>315</td>
<td>RUSK</td>
<td>CROSBY</td>
<td>8545</td>
</tr>
<tr>
<td>316</td>
<td>CAPRALLOW</td>
<td>DALLAS</td>
<td>8410</td>
</tr>
<tr>
<td>317</td>
<td>COOPER</td>
<td>DELTA</td>
<td>8241</td>
</tr>
<tr>
<td>318</td>
<td>CARRIZO SPRS.</td>
<td>DIMMIT</td>
<td>9520</td>
</tr>
<tr>
<td>319</td>
<td>CITO</td>
<td>EASTLAND</td>
<td>8662</td>
</tr>
<tr>
<td>320</td>
<td>LA GRANGE</td>
<td>FAYETTE</td>
<td>9016</td>
</tr>
<tr>
<td>321</td>
<td>MT VERNON</td>
<td>FRANKLIN</td>
<td>8246</td>
</tr>
<tr>
<td>322</td>
<td>ANNERSON</td>
<td>GRIMES</td>
<td>8846</td>
</tr>
<tr>
<td>323</td>
<td>MEMPHIS</td>
<td>HALL</td>
<td>8297</td>
</tr>
<tr>
<td>324</td>
<td>ALAMO</td>
<td>Hidalgo</td>
<td>9894</td>
</tr>
<tr>
<td>325</td>
<td>GRANBURY</td>
<td>HOOD</td>
<td>8572</td>
</tr>
<tr>
<td>326</td>
<td>CROCKETT</td>
<td>HOUSTON</td>
<td>8634</td>
</tr>
<tr>
<td>327</td>
<td>STINNETT</td>
<td>HUTCHINSON</td>
<td>8117</td>
</tr>
<tr>
<td>328</td>
<td>JASPER</td>
<td>JASPER</td>
<td>8643</td>
</tr>
<tr>
<td>329</td>
<td>GROVES</td>
<td>JEFFERSON</td>
<td>8759</td>
</tr>
<tr>
<td>330</td>
<td>PORT NECHES</td>
<td>JEFFERSON</td>
<td>8799</td>
</tr>
<tr>
<td>331</td>
<td>COTULLA</td>
<td>LASALLE</td>
<td>9476</td>
</tr>
<tr>
<td>332</td>
<td>GINDINGS</td>
<td>LEE</td>
<td>8468</td>
</tr>
<tr>
<td>333</td>
<td>GROSBECK</td>
<td>LEMES</td>
<td>8671</td>
</tr>
<tr>
<td>334</td>
<td>LIND</td>
<td>LIND</td>
<td>8979</td>
</tr>
<tr>
<td>335</td>
<td>NEWTON</td>
<td>NEWTON</td>
<td>8460</td>
</tr>
<tr>
<td>336</td>
<td>BRIDGE CITY</td>
<td>ORANG</td>
<td>8774</td>
</tr>
<tr>
<td>337</td>
<td>VINO</td>
<td>ORANG</td>
<td>8761</td>
</tr>
<tr>
<td>338</td>
<td>PALO PINTO</td>
<td>PLO PINTO</td>
<td>8541</td>
</tr>
<tr>
<td>339</td>
<td>LIVINGSTON</td>
<td>POLK</td>
<td>8716</td>
</tr>
<tr>
<td>340</td>
<td>REFUGIO</td>
<td>RFFUGIO</td>
<td>9345</td>
</tr>
<tr>
<td>341</td>
<td>FRANKLIN</td>
<td>ROBERTSON</td>
<td>8766</td>
</tr>
</tbody>
</table>
### Table 3-2. Input Data for the Example Run (Continuation 6)

<table>
<thead>
<tr>
<th>Location</th>
<th>Value</th>
<th>Location</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEMPHILL</td>
<td>8511</td>
<td>SARINE</td>
<td>3413</td>
</tr>
<tr>
<td>SAN AUGUSTINE</td>
<td>8491</td>
<td>SN AUGUST</td>
<td>3471</td>
</tr>
<tr>
<td>COLSPRING</td>
<td>8754</td>
<td>SN JACINT</td>
<td>3567</td>
</tr>
<tr>
<td>SAN SABA</td>
<td>8886</td>
<td>SAN SABA</td>
<td>4242</td>
</tr>
<tr>
<td>MATHIS</td>
<td>9448</td>
<td>SAN PTEO</td>
<td>3048</td>
</tr>
<tr>
<td>CENTER</td>
<td>9443</td>
<td>SHELBY</td>
<td>3505</td>
</tr>
<tr>
<td>RIO GRANDE CITY</td>
<td>9861</td>
<td>STARR</td>
<td>3847</td>
</tr>
<tr>
<td>BENRROOK</td>
<td>8499</td>
<td>TARRANT</td>
<td>4140</td>
</tr>
<tr>
<td>EVERMAN</td>
<td>8505</td>
<td>4110</td>
<td></td>
</tr>
<tr>
<td>RIVERAOKS</td>
<td>8479</td>
<td>4122</td>
<td></td>
</tr>
<tr>
<td>GROVETON</td>
<td>8661</td>
<td>Trinity</td>
<td>3605</td>
</tr>
<tr>
<td>WOODVILLE</td>
<td>9664</td>
<td>Tyler</td>
<td>3458</td>
</tr>
<tr>
<td>EL CAMPO</td>
<td>9115</td>
<td>Wharton</td>
<td>3649</td>
</tr>
<tr>
<td>WHEELER</td>
<td>8126</td>
<td>Wheeler</td>
<td>4820</td>
</tr>
<tr>
<td>IOWA PARK</td>
<td>8527</td>
<td>Wichita</td>
<td>4445</td>
</tr>
<tr>
<td>GEORGETOWN</td>
<td>8927</td>
<td>Williamso</td>
<td>4014</td>
</tr>
<tr>
<td>TAYLOR</td>
<td>8922</td>
<td>3962</td>
<td></td>
</tr>
<tr>
<td>QUITMAN</td>
<td>8340</td>
<td>Wood</td>
<td>3806</td>
</tr>
<tr>
<td>PLAINES</td>
<td>8735</td>
<td>Yoakum</td>
<td>5105</td>
</tr>
<tr>
<td>ZAPATA</td>
<td>9746</td>
<td>4009</td>
<td></td>
</tr>
</tbody>
</table>

1. AZLI MULESHOE PD: 7.35, 10.23, 4.38, 5.12
2. AZKK MORTON SO: 47, 6.28, 5.60, 11.91
3. AZKW SPUR PD: 94, 0.00, 0.00, 0.00
4. AZK7 FLOYDADA SO: 106, 2.86, 8.16, 8.61
5. AZLQ POST SO: 118, 25.27, 55.61, 0.00
6. AZLA PLAINVIEW PD: 118, 7.67, 14.67, 0.00
7. AZLR PLAINVIEW SO: 118, 0.00, 0.00, 0.00
8. AZLG LEVELAND PD: 157, 10.45, 27.40, 0.00
9. AZLC LITTLEFIELD PD: 183, 8.81, 21.04, 0.00
10. AZK6 LITTLEFIELD SO: 183, 4.57, 11.05, 0.00
11. AZT7 OLTON PD: 184, 5.36, 16.01, 0.00
12. AZTI OLTON SO: 0.00, 0.00, 0.00, 0.00
13. AZGL LUROCK DPS: 193, 31.12, 94.15, 0.00
14. AZLK LUROCK PD: 193, 242.28, 340.68, 0.00
15. AZLL LUROCK SO: 193, 39.61, 59.13, 0.00
16. AZLR SLATON PD: 194, 10.01, 23.71, 0.00
17. AZLJ TAHOKA PD: 195, 5.72, 17.21, 0.00
18. AZK9 TAHOKA SO: 195, 4.03, 9.94, 0.00
19. AZLF BROWNFIELD PD: 269, 17.09, 40.09, 0.00
20. AZLF BROWNFIELD SO: 269, 4.03, 8.36, 0.00

3-19
Table 3-2. Input Data for the Example Run (Continuation 7)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>399</td>
<td>AZLN</td>
<td>DENVER</td>
<td>CITY</td>
<td>PD</td>
<td>.00</td>
<td>.00</td>
<td>295</td>
<td>7.39</td>
<td>20.87</td>
<td>.00</td>
<td>.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>NAFE</td>
<td>MULESHOE</td>
<td>S:O.</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>8</td>
<td>1.66</td>
<td>5.26</td>
<td>.00</td>
<td>.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>401</td>
<td>NABG</td>
<td>CROSBTON</td>
<td>S:O.</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>312</td>
<td>1.66</td>
<td>5.26</td>
<td>.00</td>
<td>.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>402</td>
<td>NACG</td>
<td>LEVELLAND</td>
<td>S:O.</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>157</td>
<td>2.68</td>
<td>8.48</td>
<td>.00</td>
<td>.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>403</td>
<td>NAFA</td>
<td>PLAINES</td>
<td>S:O.</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>357</td>
<td>1.53</td>
<td>4.46</td>
<td>.00</td>
<td>.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>404</td>
<td>AAA</td>
<td>JUSTIN</td>
<td>SWITCHER</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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>
</tr>
</thead>
<tbody>
<tr>
<td>513</td>
<td>100.</td>
<td>100.</td>
<td></td>
</tr>
<tr>
<td>514</td>
<td>100.</td>
<td>100.</td>
<td></td>
</tr>
<tr>
<td>515</td>
<td>135.</td>
<td>135.</td>
<td></td>
</tr>
<tr>
<td>516</td>
<td>135.</td>
<td>135.</td>
<td></td>
</tr>
<tr>
<td>517</td>
<td>100.</td>
<td>100.</td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>100.</td>
<td>100.</td>
<td></td>
</tr>
<tr>
<td>519</td>
<td>135.</td>
<td>135.</td>
<td></td>
</tr>
<tr>
<td>520</td>
<td>135.</td>
<td>135.</td>
<td></td>
</tr>
<tr>
<td>521</td>
<td>100.</td>
<td>100.</td>
<td></td>
</tr>
<tr>
<td>522</td>
<td>100.</td>
<td>100.</td>
<td></td>
</tr>
<tr>
<td>523</td>
<td>135.</td>
<td>135.</td>
<td></td>
</tr>
<tr>
<td>524</td>
<td>135.</td>
<td>135.</td>
<td></td>
</tr>
<tr>
<td>525</td>
<td>0.</td>
<td>0.</td>
<td></td>
</tr>
<tr>
<td>526</td>
<td>0.</td>
<td>0.</td>
<td></td>
</tr>
<tr>
<td>527</td>
<td>10.</td>
<td>10.</td>
<td></td>
</tr>
<tr>
<td>528</td>
<td>10.</td>
<td>10.</td>
<td></td>
</tr>
<tr>
<td>529</td>
<td>0.</td>
<td>0.</td>
<td></td>
</tr>
<tr>
<td>530</td>
<td>0.</td>
<td>0.</td>
<td></td>
</tr>
<tr>
<td>531</td>
<td>10.</td>
<td>10.</td>
<td></td>
</tr>
<tr>
<td>532</td>
<td>10.</td>
<td>10.</td>
<td></td>
</tr>
<tr>
<td>533</td>
<td>0.</td>
<td>0.</td>
<td></td>
</tr>
<tr>
<td>534</td>
<td>0.</td>
<td>0.</td>
<td></td>
</tr>
<tr>
<td>535</td>
<td>10.</td>
<td>10.</td>
<td></td>
</tr>
<tr>
<td>536</td>
<td>10.</td>
<td>10.</td>
<td></td>
</tr>
<tr>
<td>537</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>538</td>
<td>900.</td>
<td>0</td>
<td>900.</td>
</tr>
<tr>
<td>539</td>
<td>1.</td>
<td>3.</td>
<td>1.</td>
</tr>
<tr>
<td>540</td>
<td>900.</td>
<td>0</td>
<td>900.</td>
</tr>
<tr>
<td>541</td>
<td>1.</td>
<td>3.</td>
<td>1.</td>
</tr>
<tr>
<td>542</td>
<td>900.</td>
<td>0</td>
<td>900.</td>
</tr>
<tr>
<td>543</td>
<td>1.</td>
<td>6</td>
<td>1.</td>
</tr>
<tr>
<td>544</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>545</td>
<td>900.</td>
<td>0</td>
<td>900.</td>
</tr>
<tr>
<td>546</td>
<td>1.</td>
<td>3.</td>
<td>1.</td>
</tr>
<tr>
<td>547</td>
<td>900.</td>
<td>0</td>
<td>900.</td>
</tr>
<tr>
<td>548</td>
<td>1.</td>
<td>3.</td>
<td>1.</td>
</tr>
<tr>
<td>549</td>
<td>900.</td>
<td>0</td>
<td>900.</td>
</tr>
<tr>
<td>550</td>
<td>1.</td>
<td>6</td>
<td>1.</td>
</tr>
<tr>
<td>551</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>552</td>
<td>900.</td>
<td>0</td>
<td>900.</td>
</tr>
<tr>
<td>553</td>
<td>1.</td>
<td>3.</td>
<td>1.</td>
</tr>
<tr>
<td>554</td>
<td>900.</td>
<td>0</td>
<td>900.</td>
</tr>
<tr>
<td>555</td>
<td>1.</td>
<td>3.</td>
<td>1.</td>
</tr>
<tr>
<td>556</td>
<td>900.</td>
<td>0</td>
<td>900.</td>
</tr>
<tr>
<td>557</td>
<td>1.</td>
<td>6</td>
<td>1.</td>
</tr>
<tr>
<td>558</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>559</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>560</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>561</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>562</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>563</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>564</td>
<td>35</td>
<td>300</td>
<td>4.26</td>
</tr>
<tr>
<td>565</td>
<td>60</td>
<td>8615.25</td>
<td>14.47</td>
</tr>
<tr>
<td>566</td>
<td>500</td>
<td>500</td>
<td>3.62</td>
</tr>
<tr>
<td>567</td>
<td>300</td>
<td>300</td>
<td>13</td>
</tr>
<tr>
<td>568</td>
<td>426</td>
<td>459</td>
<td>5.78</td>
</tr>
<tr>
<td>569</td>
<td>50</td>
<td>175</td>
<td>9.34</td>
</tr>
</tbody>
</table>

77-53, Vol. IV
**Table 3-2. Input Data for the Example Run (Continuation 10)**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</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>
<th>Value 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>570</td>
<td>ING/NL</td>
<td>50 200 .45 .45 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>571</td>
<td>ADM/NL</td>
<td>300 300 .45 .0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>572</td>
<td>NCIC</td>
<td>50 90 .0 9.21 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>573</td>
<td>DB/DLS</td>
<td>90 50 3.49 1.1 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>574</td>
<td>DB/ANT</td>
<td>90 50 .45 .45 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>575</td>
<td>11</td>
<td>110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>576</td>
<td>2</td>
<td>AAAA</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>577</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>578</td>
<td>4</td>
<td>AAAA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>579</td>
<td>5</td>
<td>QSTATE CENTER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>580</td>
<td>6</td>
<td>6.63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>581</td>
<td>7</td>
<td>7.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- CPU: .787
- CTP: .091
- SUPS: 17.904
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>AAZK</th>
<th>AZKH</th>
<th>AAZK</th>
<th>AZLO</th>
<th>AZLA</th>
<th>AZLD</th>
<th>AZLC</th>
<th>AAZK</th>
<th>AZFL</th>
<th>AZLF</th>
<th>AZLL</th>
<th>AAZL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAFIN</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>3.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.2</td>
<td>1.6</td>
<td>2.7</td>
<td>4.1</td>
<td>5.3</td>
<td>7.0</td>
</tr>
<tr>
<td>TRAFOUT</td>
<td>2.2</td>
<td>1.1</td>
<td>2.4</td>
<td>1.1</td>
<td>7.6</td>
<td>1.0</td>
<td>1.0</td>
<td>2.1</td>
<td>2.1</td>
<td>15.4</td>
<td>35.4</td>
<td>16.9</td>
<td>3.0</td>
</tr>
<tr>
<td>TRAFIN</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>TRAFOUT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>TRAFIN</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>TRAFOUT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>TRAFIN</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>TRAFOUT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run (Continuation 1)

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>APLA</td>
<td>3.051</td>
<td>APLH</td>
<td>1.823</td>
<td>APLM</td>
<td>1.093</td>
<td>AZLC</td>
<td>1.145</td>
</tr>
<tr>
<td>AZLI</td>
<td>1.715</td>
<td>APLH</td>
<td>0.189</td>
<td>APLM</td>
<td>0.906</td>
<td>AZLC</td>
<td>0.281</td>
</tr>
<tr>
<td>AZLI</td>
<td>0.339</td>
<td>AZL6</td>
<td>0.347</td>
<td>AZLF</td>
<td>2.370</td>
<td>AZLF</td>
<td>0.357</td>
</tr>
<tr>
<td>NAAF</td>
<td>0.221</td>
<td>NAMG</td>
<td>0.220</td>
<td>NAMG</td>
<td>0.308</td>
<td>NAMG</td>
<td>0.800</td>
</tr>
</tbody>
</table>

TOTAL TRAFFIC ORIGINATED FROM SYS. TERMS (MTR/SEC) = 41.00

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>APLA</td>
<td>2.564</td>
<td>APLK</td>
<td>1.411</td>
<td>APLM</td>
<td>2.373</td>
<td>APLF</td>
<td>1.544</td>
</tr>
<tr>
<td>AZLA</td>
<td>7.415</td>
<td>AZLU</td>
<td>1.056</td>
<td>AZLF</td>
<td>3.767</td>
<td>AZLI</td>
<td>2.054</td>
</tr>
<tr>
<td>AZLI</td>
<td>2.155</td>
<td>AZLU</td>
<td>1.140</td>
<td>AZLF</td>
<td>45.420</td>
<td>AZLI</td>
<td>7.480</td>
</tr>
<tr>
<td>AZLI</td>
<td>2.245</td>
<td>AZK6</td>
<td>1.225</td>
<td>AZLF</td>
<td>5.450</td>
<td>AZLF</td>
<td>1.114</td>
</tr>
<tr>
<td>NAAF</td>
<td>0.401</td>
<td>NAMG</td>
<td>0.791</td>
<td>NAMG</td>
<td>1.131</td>
<td>NAMG</td>
<td>0.648</td>
</tr>
</tbody>
</table>

TOTAL TRAFFIC = 119.30

TOTAL SYSTEM TRAFFIC = 119.30
Table 3-3. Printer Output from the Example Run
(Continuation 2)

<table>
<thead>
<tr>
<th>TERM.</th>
<th>AZLH</th>
<th>AGLL</th>
<th>AGLK</th>
<th>AGLM</th>
<th>AZTI</th>
<th>AGLA</th>
<th>AGLC</th>
<th>AGLD</th>
<th>AGLH</th>
<th>AGLI</th>
<th>AGLJ</th>
<th>AGLK</th>
<th>AGLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZLH</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>AZLA</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>AZLJ</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>AZLL</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 2)

1. POINT TO POINT DISTANCE MATRIX

<table>
<thead>
<tr>
<th>TERM.</th>
<th>AZLH</th>
<th>AGLL</th>
<th>AGLK</th>
<th>AGLM</th>
<th>AZTI</th>
<th>AGLA</th>
<th>AGLC</th>
<th>AGLD</th>
<th>AGLH</th>
<th>AGLI</th>
<th>AGLJ</th>
<th>AGLK</th>
<th>AGLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZLH</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>AZLA</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>AZLJ</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>AZLL</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>

1. NCSS = AAAA

2. CPU UTILIZATION PER PROCESSOR IS .700

3. REGS: 1 * SYS. TERM:

- AZLH MULESHOE PD
- AZLA POST SD
- AZLJ LITTLEFIELD PD
- AZLE LUBBOCK PD
- AZLS TAHOMA SD
- NAAS MULESHOE 9.0
- AAAA AUSTRALIAN SWITCHE

4. INDICES FOR SYC. TERM:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

5. RSC = AAAA FOR REGION 1
<table>
<thead>
<tr>
<th>SYSTEM TERMIN.</th>
<th>AZL1</th>
<th>AZLH</th>
<th>AZLZ</th>
<th>AZLC</th>
<th>AZLA</th>
<th>AZLM</th>
<th>AZLD</th>
<th>AZLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. OF LINES AFD.</td>
<td>1.2KB</td>
<td>2.4KB</td>
<td>4.8KB</td>
<td>1.2KB</td>
<td>2.4KB</td>
<td>4.8KB</td>
<td>1.2KB</td>
<td>2.4KB</td>
</tr>
<tr>
<td>LINE UTILIZATION</td>
<td>0.003</td>
<td>0.016</td>
<td>0.024</td>
<td>0.012</td>
<td>0.005</td>
<td>0.003</td>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>DISTANCE FROM NSC TRAFFIC</td>
<td>0.715</td>
<td>0.715</td>
<td>0.715</td>
<td>0.715</td>
<td>0.715</td>
<td>0.715</td>
<td>0.715</td>
<td>0.715</td>
</tr>
<tr>
<td>LINE TO CPU</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
</tr>
<tr>
<td>CPU TO LINE</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
</tr>
<tr>
<td>LINE RESPONSE TIME</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
<td>5.277</td>
</tr>
</tbody>
</table>

**REGIONAL STN NETWORK AND ITS COSTS 1**

Table 3-3. Printer Output from the Example Run (Continuation 3)
Table 3-3. Printer Output from the Example Run
(Continuation 4)

<table>
<thead>
<tr>
<th>DROP</th>
<th>TOTAL COST</th>
<th>INST. COST</th>
<th>RECUR. COST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>240</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>TOTAL INST. Gd.$</td>
<td>120</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>RECUR. Gd.$</td>
<td>14350</td>
<td>1514</td>
<td>3518</td>
</tr>
<tr>
<td>TOTAL INST. COST</td>
<td>14350</td>
<td>1514</td>
<td>3518</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSTEM TERMINI</th>
<th>NAES</th>
<th>NAPG</th>
<th>NACA</th>
<th>NAFQ</th>
<th>NAAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE UTILIZATION</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td></td>
</tr>
<tr>
<td>DISTANCE FROM RSC TRAFFIC LINE TO CPU</td>
<td>399</td>
<td>314</td>
<td>361</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CPU TO LINE</td>
<td>221</td>
<td>221</td>
<td>221</td>
<td>221</td>
<td>221</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>MODEN</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DROP</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>ANNUAL DECUM. COST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>14364</td>
<td>11160</td>
<td>12790</td>
<td>12790</td>
<td>12896</td>
</tr>
<tr>
<td>MODEN</td>
<td>528</td>
<td>528</td>
<td>528</td>
<td>528</td>
<td>528</td>
</tr>
<tr>
<td>DROP</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>TOTAL INST. COST</td>
<td>14350</td>
<td>1514</td>
<td>3518</td>
<td>3518</td>
<td>1250</td>
</tr>
<tr>
<td>RECUR. COST</td>
<td>14350</td>
<td>1514</td>
<td>3518</td>
<td>3518</td>
<td>1250</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>2549702</td>
<td></td>
<td></td>
<td></td>
<td></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>1.2KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.8KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LINE UTILIZATION</td>
<td>.138</td>
<td>.017</td>
</tr>
<tr>
<td>TOTAL MILEAGE</td>
<td>617</td>
<td>308</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>13.617</td>
</tr>
<tr>
<td>LINE RESPONSE TIME</td>
<td>4.038</td>
<td>3.416</td>
</tr>
<tr>
<td>INST. COSTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>260</td>
<td>190</td>
</tr>
<tr>
<td>MODEM</td>
<td>1300</td>
<td>950</td>
</tr>
<tr>
<td>DROP</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ANNUAL RECURR. COST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>16882</td>
<td>11941</td>
</tr>
<tr>
<td>SER.T.</td>
<td>4640</td>
<td>3820</td>
</tr>
<tr>
<td>MODEM</td>
<td>6184</td>
<td>5016</td>
</tr>
<tr>
<td>DROP</td>
<td>3120</td>
<td>2286</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>31086</td>
<td>22127</td>
</tr>
<tr>
<td>INST. COST</td>
<td>1560</td>
<td>1140</td>
</tr>
<tr>
<td>RECUR. COST</td>
<td>31086</td>
<td>22127</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>32646</td>
<td></td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run
(Continuation 6)

<table>
<thead>
<tr>
<th>Regional Center</th>
<th>Subnetwork Begins at</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZL</td>
<td>AZL</td>
</tr>
<tr>
<td>AZLA</td>
<td>AZKZ</td>
</tr>
<tr>
<td>AZTI</td>
<td>AZKA</td>
</tr>
<tr>
<td>AZLG</td>
<td>AZKG</td>
</tr>
<tr>
<td>AZLD</td>
<td>AZKL</td>
</tr>
<tr>
<td>AZLS</td>
<td>AZLJ</td>
</tr>
<tr>
<td>AZLF</td>
<td>AZLI</td>
</tr>
<tr>
<td>AZLE</td>
<td>AZLH</td>
</tr>
</tbody>
</table>

Note: The diagram may not be fully legible due to the image quality.
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 .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 mileage. Summarized costs are also provided.

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

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

ASSUMED NUMBER OF REGIONS
ENTER NR AND STRIKE RETURN KEY
TYPE IN NO. OF SYS. TERMS; DATA HASSES AND CITIES WITH FORMAT 45
THERE ARE 2 SYS. TERMS; 4 DATA HASSES 35H CITIES
TYPE IN DATA BASE LOCATIONS WITH FORMAT 61X(AN)
4 DATA BASES ARE AT MMMM NNNN HHHH
TYPE IN CITY Y=H WITH FORMAT 1(3X15.2Y15)
4 DATA BASES ARE AT PIP NO.: NAME MAPPING ADD. AND TRAFFIC
WITH FORMAT 14.1X46,14.6F9.2
TYPE IN NO. OF RATE STRUCTURES UNDER
CONSIDERATION WITH FORMAT 13
TYPE IN RATE APPLICATION TO EACH COMMON.
WHICH EACH SYS. TERM, WITH FORMAT 1012
READ IN TRAFFIC DENSITY TYPE AND RATE STRUCTURE
FOR EACH CITY WITH FORMAT 4011
TYPE IN NO. OF LINF TYPES APPLICABLE WITH FORMAT 13
TYPE IN NAME+ CAPACITY; UTIL FACTOR AvAIL. FOR
EACH LINE TYPE WITH FORMAT A6,1X14.1X15.2X12.2X11
TYPE IN NO. OF DEVICES AND NAMES FOR EACH LINF TYPE
WITH FORMAT 13/101(AD+1Y)
TYPE IN INST. AND RECUR. COSTS WHT
RATE STRUCTURE: LINF TYPE; DEVICE TRAFFIC DENSITY
AND DUEPLEXING RATE WITH FORMAT 2F9.2/2F9.2
TYPE IN INST. COSTS WITH INST. WHT
RATE LINF. DENSITY + DUEPLEXING RATE
WITH FORMAT 4F9.2
TYPE IN INDEX FOR LEEVITY OF LINF RECUR. COST
FUNCTION WITH LINEAR AND NONLINEAR OTHERWISE
WITH FORMAT 11 FOR EACH LINE TYPE
TYPE IN RECUR. COSTS WITH FORMAT 4F9.2 IF LINEAR
WITH FORMAT 10F9.3/10F9.3 IF NONLINEAR
IF NONLINEAR USE 1NF8.2
TYPE IN ACTION INDICES FOR EACH REGION
1ST ELEMENT: 1 = INSERTION TO THIS PRELOADED REGION IS OK
2ND ELEMENT: 1 = OPTIMIZATION IS NEEDED
TYPE IN REGION INDEX AND ACTION NUMBER NEED
WITH FORMAT 2I AND END IT WITH A .D
TYPE IN NPL, MAX, MIN, LROI, LROI, MROI, MROI
TANITA TAO IN FORMAT (514.2F7.5)
TYPE IN NO. OF MSG TYPE+, AND TRAFFIC STATISTICS
SUCH AS MSGAM, MSQAM, MSQAM, RATIO WITH
FORMAT 14/1AD+7(219.2F8.3)
TYPE IN PRELOADED SYSTEM TERMINALS; AND RFC WITH
FORMAT 11X14.4F9.2
32/3 DISTANCE TYPES ARE OVERSIZED
ASSUME A SYSTEM CENTERED
ENTER CODE FOR MSGC AND STRIKE RETURN KEY
INPUT TOTAL NO. OF TRANSACTIONS AND NO. OF ACCESS AT THE SWITCHER
ENTER WITH 4,5 AND 13 UNDER X5C/FRC
READ IN LIMITS ON NO. OR SYS. TERMS, ON A .R.
RESPONSE TIME READ AND NO. OF PROCESSORS WITH FORMAT
13+FD.2+12
IF PLOTTING IS REQUIRED, TYPE 1 WITH FORMAT 13
TRY LINK HAS BEEN ACCESSED FOR 10052 TIMES
ONE TIME HAS BEEN ACCESSED FOR 22 TIMES

3-33
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.

![CalComp Plot from the Example Run](image)
REFERENCES


APPENDIX A

STACOM PROGRAM LISTING

5192<STACOM(1).MAIN/D777

C*************************************************************************************C
C*C STACOM TOPOLOGY PROGRAM
C*C JET PROPULSION LABORATORY
C*C 400 OAK GROVE DRIVE
C*C PASADENA, CALIFORNIA 91103
C*************************************************************************************C
C THIS PROGRAM IS DESIGNED TO PERFORM FORMATIONS OF REGIONS, SELECTIONS
C OF REGIONAL SWITCHING CENTERS, FORMATIONS OF INITIAL REGIONAL NETWORKS,
C OPTIMIZATION OF REGIONAL NETWORKS USING THE PEAN-WILLIAMS METHOD IF
C REQUESTED, AND FINALLY FORMATION OF AN INTERREGION NETWORK AND ITS
C OPTIMIZATION
C************************************************************************************

*************************************************************************************
C THIS PROGRAM CONTAINS ONE MAIN PROGRAM AND ELEVEN SUBPROGRAMS.
C THEY ARE AS FOLLOWS:
C MAIN PROGRAM: MAIN (REGION ASSIGNMENTS OF SYSTEM TERMINATIONS)
C SUBPROGRAM-1: RGNMET (REGIONAL NETWORK FORMATION AND ITS OPTIMIZATION)
C SUBPROGRAM-2: IRNOP (INTER-REGION NETWORK OPTIMIZATION)
C SUBPROGRAM-3: ICOSTJ (COSTING FUNCTION)
C SUBPROGRAM-4: RHOFUN (LINE UTILIZATION FUNCTION)
C SUBPROGRAM-5: LINLX (LINE CONFL. 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: PLOP (PLOTTING EACH DROP ON A MULTIDROP NETWORK)
C SUBPROGRAM-11: RSPNSE (ESTIMATING RESPONSE TIME)

*************************************************************************************
PARAMETER MW=4, IWT=100, NLIMIT =2, NPC=360, IDDP=1A
PARAMETER NP1=130, NP2=1, NP3=4, NP4=9
PARAMETER NP5=NP6+NP7=NP8/2-NPC+1/4+1
PARAMETER NP9=NP7/2=10-NPC
COMMON AFIN/SVR(NP1), NRSC(MW), NUMPR(MW), TRAFDN(NP1),
   TRAFIT(NP1)
   /VH/ IVER(NP1), ITHORN(NP1)
   /CONST/ NL1*N2*NP1*NP2*NP3*NP4*NP5*NP6*NP7*NP8
   /INF/ IRATE(NP9), TRAND(NP1)*IFLAG(NP2)*NP3
   /NCOST/ ARMSTC(NP2)*NP3*NP4, FRTCC(NP2)*NP5, NP6, NP7, NP8
   /NKST/ ANSTL(NP2)*NP5, NP3, NP4, RECLN(NP2)*NP5, NP3, NP4, NP6, NP7, NP8
   /LINCN/ LINCNN(NP3), LINCAP(NP3), UTIL17(NP3)
   /REF/ IFREF(NP1), TRAFC(NP1), PSNCF(NP6), MAPADP(NP1)
   /OVFR/ TVRO(NP2)*NP3+1, IVER1
   /NAME/ NAMEST(NP1)+1, LNAME(NP3), NLNAME(NP3), NAMFH(NP4)
   /SUM/ ASMU(NP1), SUM
   /XMT/ TIMXMT(NP3), WATT(NP1)
   /WA/ AMSL(1)
   /RND/ NTERMNIT=TIME0, MPROC, MPR, GP
   /ADV/ IADC(NP1)*KCHG*KADD, OKCHG=FIRST DROP, KADD=JUST FOR LINE
   INTEGER DSFNC
DIMENSION IACT(NP3), INDVPT(NP1)
DIMENSION NUMPR(NP1), TRAFC(NP1), HRAF(NP7)
57  DATA ITAFC/TRAFCINTRAFCOUT/  
58  DIMENSION TRM(NMAX,NPO), ORM(NMAX,NPO), NUMR(NP1), NUMR(NP1+4)  
59  INTEGER SRV  
60  DIMENSION OUTPR(NP1)  
61  NMAX=NPO  MAXIMUM SIZE FOR OVERFLOW DISTANCE DATA TABLE  
62  CPUAVG=0.  
63  
64  C SELECT NUMBER OF REGIONS  
65  C  
66      226 WRITE(6,226)  
67      READ(5,735) NR1  
68      WRITE(10,1011) NR1  
69      ANR1=NP1  
70  C  
71  C READ IN TRAFFIC DENSITY INDEX AND RATE STRUCTURE FOR EACH SYSTEM  
72  C TERMINATION IN THE SYSTEM  
73  C  
74      CALL CREADA(N1)  
75  C  
76      C READ IN RATE APPLICATION MATRIX  
77  C  
78      CALL CREADO(N2)  
79  C  
80      C READ IN NAMES; CAPACITIES, UTILIZATION FACTORS, AND AVAILABILITIES  
81  C FOR LINES APPLICABLE IN THE SYSTEM  
82  C  
83      CALL CREADD(N3)  
84  C  
85      C READ IN INSTALLATION AND RECURRING COSTS FOR CHARGEABLE ITEMS  
86  C REQUIRED FOR COMMUNICATION LINES  
87  C  
88      CALL CREADD(N4)  
89  C  
90      C READ IN INSTALLATION AND RECURRING COSTS FOR LINES  
91  C  
92      CALL CREADF  
93  C  
94      C READ IN ACTIONS TO BE PERFORMED ON EACH REGIONAL NETWORK  
95  C 1ST ELEMENT: 1=INSERTIONS TO PRELOADED REGIONS ARE ALLOWED  
96  C 0= SUCH AN ACTION IS NOT ALLOWED  
97  C 2ND ELEMENT: 1=NETWORK OPTIMIZATION IS TO BE PERFORMED  
98  C 0=NO OPTIMIZATION IS NEEDED  
99  C  
100      CALL CREADDK  
101  C  
102      C READ IN LINE AND LINE PROTOCOL CHARACTERISTICS  
103  C  
104      CALL CREADR  
105  C  
106      C CONVERT TRAFFIC FROM CHARACTERS/SEC TO RITS/SFC  
107  C  
108      DO 85 K=1,2  
109      DO 85 I=1,N1  
110      DO 85 L=1,N7  
111      TRAFD(I,K,L)=TRAFD(I,K,L)*R+60.  
112      85 CONTINUE  
113      ISUM=0

A-2
DO 25 J=1,NCITY
   TSUM=TSUM+1
    TRAFF(J)=TSUM
25 CONTINUE

DO 70 I=1,NR1
   NUMPR(I) = 0
    NUMPR(I) = NQNO. OF SYSTEM TERMINATIONS AT EACH REGION
70 CONTINUE

WRITE(6,200)

READ(5,800) NCODE,NSTATE,NREGO
WRITE(6,800) NCODE, NSTATE, NREGO

NSTATE=LOCAL(NSTATE) FIND CARDINAL INDEX
GO TO (RO1+RO2+240)+NCODE

CONTINUE

NPR1=NSTATE+1
NPR1=NPR1+NREGO = NUMPR(NREGO) + 1
CONTINUE

NRCN(NREGO) = NSTATE
GO TO R0S
CONTINUE

CONTINUE

70 CONTINUE

40 CONTINUE

MIRIT=NBR13)
3F05 RER0I
3rA00) NCODE ,NSTATE,NREGO}
WRITE(11,WTRJ)
NCOW
NSTATE
RER^3
NSTATF
LOCATI+JSTATE}
0FIN(+5
GD TO ?O1rAO?NC0DE
802 CONTINUE
801 CONTINUE
5VR I NSTATE }	 NREt{}
728 Ni1MPRiNREGQ)	 a Nl1MpR1lIRgGO} 	 +	 1
ROi CO^JTIN1fE
130 CONTINUE
129 NRSC(NREGO) = NSTATE
128 GO TO R0S
127 CONTINUE
126 GO TO 710 L=1+N1
125 TRAFON(I)=0.
124 70 CONTINUE
123 IOVERI=1 G COUNTER FOR OVERSIZED TRAFFIC DATA
122 C CALCULATE DISTANCE DATA BETWEEN SYSTEM TERMINATIONS:
121 C
120 DO 30 K=1,NCITY
119 DO 30 J=1,NCITY
118 IF(J,K) 51,30,30
117 51 CONTINUE
116 50 CONTINUE
115 49 CONTINUE
114 48 CONTINUE
113 47 CONTINUE
112 46 CONTINUE
111 45 CONTINUE
110 44 CONTINUE
109 43 CONTINUE
108 42 CONTINUE
107 41 CONTINUE
106 40 CONTINUE
105 39 CONTINUE
104 38 CONTINUE
103 37 CONTINUE
102 36 CONTINUE
101 35 CONTINUE
100 34 CONTINUE
99 33 CONTINUE
98 32 CONTINUE
97 31 CONTINUE
96 30 CONTINUE
95 29 CONTINUE
94 28 CONTINUE
93 27 CONTINUE
92 26 CONTINUE
91 25 CONTINUE
90 24 CONTINUE
89 23 CONTINUE
88 22 CONTINUE
87 21 CONTINUE
86 20 CONTINUE
85 19 CONTINUE
84 18 CONTINUE
83 17 CONTINUE
82 16 CONTINUE
81 15 CONTINUE
80 14 CONTINUE
79 13 CONTINUE
78 12 CONTINUE
77 11 CONTINUE
76 10 CONTINUE
75 9 CONTINUE
74 8 CONTINUE
73 7 CONTINUE
72 6 CONTINUE
71 5 CONTINUE
70 4 CONTINUE
69 3 CONTINUE
68 2 CONTINUE
67 1 CONTINUE
66 0 CONTINUE
65 -1 CONTINUE
64 -2 CONTINUE
63 -3 CONTINUE
62 -4 CONTINUE
61 -5 CONTINUE
60 -6 CONTINUE
59 -7 CONTINUE
58 -8 CONTINUE
57 -9 CONTINUE
56 -10 CONTINUE
55 -11 CONTINUE
54 -12 CONTINUE
53 -13 CONTINUE
52 -14 CONTINUE
51 -15 CONTINUE
50 -16 CONTINUE
49 -17 CONTINUE
48 -18 CONTINUE
47 -19 CONTINUE
46 -20 CONTINUE
45 -21 CONTINUE
44 -22 CONTINUE
43 -23 CONTINUE
42 -24 CONTINUE
41 -25 CONTINUE
40 -26 CONTINUE
39 -27 CONTINUE
38 -28 CONTINUE
37 -29 CONTINUE
36 -30 CONTINUE
35 -31 CONTINUE
34 -32 CONTINUE
33 -33 CONTINUE
32 -34 CONTINUE
31 -35 CONTINUE
30 -36 CONTINUE
29 -37 CONTINUE
28 -38 CONTINUE
27 -39 CONTINUE
26 -40 CONTINUE
25 -41 CONTINUE
24 -42 CONTINUE
23 -43 CONTINUE
22 -44 CONTINUE
21 -45 CONTINUE
20 -46 CONTINUE
19 -47 CONTINUE
18 -48 CONTINUE
17 -49 CONTINUE
16 -50 CONTINUE
15 -51 CONTINUE
14 -52 CONTINUE
13 -53 CONTINUE
12 -54 CONTINUE
11 -55 CONTINUE
10 -56 CONTINUE
9 -57 CONTINUE
8 -58 CONTINUE
7 -59 CONTINUE
6 -60 CONTINUE
5 -61 CONTINUE
4 -62 CONTINUE
3 -63 CONTINUE
2 -64 CONTINUE
1 -65 CONTINUE
0 -66 CONTINUE
-1 -67 CONTINUE
-2 -68 CONTINUE
-3 -69 CONTINUE
-4 -70 CONTINUE
-5 -71 CONTINUE
-6 -72 CONTINUE
-7 -73 CONTINUE
-8 -74 CONTINUE
-9 -75 CONTINUE
-10 -76 CONTINUE
-11 -77 CONTINUE
-12 -78 CONTINUE
-13 -79 CONTINUE
-14 -80 CONTINUE
-15 -81 CONTINUE
-16 -82 CONTINUE
-17 -83 CONTINUE
-18 -84 CONTINUE
-19 -85 CONTINUE
-20 -86 CONTINUE
-21 -87 CONTINUE
-22 -88 CONTINUE
-23 -89 CONTINUE
-24 -90 CONTINUE
-25 -91 CONTINUE
-26 -92 CONTINUE
-27 -93 CONTINUE
-28 -94 CONTINUE
-29 -95 CONTINUE
-30 -96 CONTINUE
-31 -97 CONTINUE
-32 -98 CONTINUE
-33 -99 CONTINUE
-34 -100 CONTINUE
-35 -101 CONTINUE
-36 -102 CONTINUE
-37 -103 CONTINUE
-38 -104 CONTINUE
-39 -105 CONTINUE
-40 -106 CONTINUE
-41 -107 CONTINUE
-42 -108 CONTINUE
-43 -109 CONTINUE
-44 -110 CONTINUE
-45 -111 CONTINUE
-46 -112 CONTINUE
-47 -113 CONTINUE
-48 -114 CONTINUE
-49 -115 CONTINUE
-50 -116 CONTINUE
-51 -117 CONTINUE
-52 -118 CONTINUE
-53 -119 CONTINUE
-54 -120 CONTINUE
-55 -121 CONTINUE
-56 -122 CONTINUE
-57 -123 CONTINUE
-58 -124 CONTINUE
-59 -125 CONTINUE
-60 -126 CONTINUE
-61 -127 CONTINUE
-62 -128 CONTINUE
-63 -129 CONTINUE
-64 -130 CONTINUE
-65 -131 CONTINUE
-66 -132 CONTINUE
-67 -133 CONTINUE
-68 -134 CONTINUE
-69 -135 CONTINUE
-70 -136 CONTINUE
-71 -137 CONTINUE
-72 -138 CONTINUE
-73 -139 CONTINUE
-74 -140 CONTINUE
-75 -141 CONTINUE
-76 -142 CONTINUE
-77 -143 CONTINUE
-78 -144 CONTINUE
-79 -145 CONTINUE
-80 -146 CONTINUE
-81 -147 CONTINUE
-82 -148 CONTINUE
-83 -149 CONTINUE
-84 -150 CONTINUE
-85 -151 CONTINUE
-86 -152 CONTINUE
-87 -153 CONTINUE
-88 -154 CONTINUE
-89 -155 CONTINUE
-90 -156 CONTINUE
-91 -157 CONTINUE
-92 -158 CONTINUE
-93 -159 CONTINUE
-94 -160 CONTINUE
-95 -161 CONTINUE
-96 -162 CONTINUE
-97 -163 CONTINUE
-98 -164 CONTINUE
-99 -165 CONTINUE
-100 -166 CONTINUE
-101 -167 CONTINUE
-102 -168 CONTINUE
-103 -169 CONTINUE
-104 -170 CONTINUE

C TOTAL INPUT TRAFFIC BY EACH SYS. TERMIN.
C TRFALL=0.0
TALLIT=0.
TALLDN=0.
DO 41 L=1+N1
   TRAFIL(L)= 0.0
   TRAFDN(L)= 0.0
   DO 42 J=1+N7
      TRAFIT(L) = TRAFIT(L) + TRAFD(J+L) + TRAFD(J+L+1)
   CONTINUE
42 TALLDN=TALLDN+TRAFDN(L)
   TALLIT=TALLIT+TRAFIT(L)
   CONTINUE
C TRFALL=TALLDN+TALLIT
C PRINT OUT TRAFFIC DATA BETWEEN SYSTEM TERMINATIONS
C NTURN=N1/15 + 1
NRFM=MOD(N1,15)
IF(NRFM .EQ. 0) NTURN=TURN-1
WRITE(IWT+11)
DO 110 K=1+N7 $FOR TEST ONLY
   KK=KK+1+1
110 CONTINUE
DO 28 KR=KK1+KK2
   WRITE(IWT+101) (INDEX(J), J=KK1KK2)
   WRITE(IWT+74) TALLDN
C PRINT OUT TRAFFIC ORIGINATED FROM EACH SYSTEM TERMINATION
28 WRITE(IWT+1013)
   WRITE(IWT+1001) (INDEX(J), J=KK1KK2)
   WRITE(IWT+74) TALLDN
C PRINT OUT TRAFFIC DESTINATED TO EACH SYSTEM TERMINATION
210 WRITE(IWT+1014)
   WRITE(IWT+1001) (INDEX(J), J=KK1KK2)
   WRITE(IWT+74) TALLIT
C PRINT OUT DISTANCE DATA BETWEEN SYSTEM TERMINATIONS
220 WRITE(IWT+75) TRFALL
C NTURN=N1/15+1
NRFM=MOD(N1,15)
IF(NRFM .EQ. 0) NTURN=TURN-1
NTURN=1 $FOR SHORT OUTPUT
DO 101 KK=1, NTURN
77-53, Vol. IV

228     KK1=(KK-1)*15 + 1
229     KK2=KK*15
230     IF(KK2 > N1) KK2=N1
231     WRITE(IWT+100) (INDXPT(J),J=KK2,1K1+1)
232     DO 99 J=1, KK2
233     IF(J.GE.KK1) KK1=J+1
234     DO 27 KK=KK2, KK1,-1
235     OUTPUT('R')=DIST(J,KP)
236     CONTINUE
237     WRITE(IWT+112) INDXPT(J), (OUTPUT(K),K=KK2, KK1,-1)
238     CONTINUE
239     WRITE(6+210)
240     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     CONTINUE
248     TPR1 = TRFALL
249     WRITE(6+2101)
250     READ(5+2102) XSAC, NREQS, QNRQSEND OF REQUESTS/TRANS AT SWITH
251     WRITE(6+2103)
252     READ(5+2104) NTERMS, TIMREQ, MPNC
253     WRITE(6+2105)
254     READ(5+2104) MPlot QMPlot=1 IF PLOT IS NEEDED
255     C
256     C PRE-CALCULATE CPU TURNAROUND TIME
257     C
258     CALL CWAITC
259     C
260     C SUM UP TOTAL TRAFFIC FOR PRELOADED SYSTEM TERMINATIONS IN REGIONS
261     C WHICH DO NOT ALLOW ANY INSERTIONS OF OTHER SYSTEM TERMINATIONS
262     C
263     TPR2=0
264     DO 77 N=1, N1
265     NK=SVR(N)
266     IF(NK .LE. 0) GOTO 77 QMRT PRELOADED
267     IF(IACTN(NK)=0) GOTO 77 INSERTIONS ARE ALLOWED
268     TRP2=TRAPTH(N) + TRAFIT(N) + TPRP
269     CONTINUE
270     CONTINUE
271     IF(IMPR(NL1)=0) GOTO 76 GONE REGION CASF
272     ANR1=ANR1+1.
273     CONTINUE
274     CONTINUE
275     TPR1=TPR1-TPR2
276     IF(NR1 .EQ. 1) GOTO 76 GONE REGION CASF
277     C
278     C DETERMINE LOWER LIMIT FOR AVERAGE REGIONAL TRAFFIC
279     C
280     ZFAC=1
281     IF(ANR1 .EQ. 0.) GOTO 340
282     TPR=TPR1/ANR1
283     GOTO 350
284     340 CONTINUE

A-5
DO 909 NREG=1:NRI
AMAXD=0.
II=0
IF((NUMPR(NREG)) .NE. 0) GOTO 5000
C ASSIGN SYSTEM TERMINATIONS TO A REGION WITHOUT ANY PRELOADING
C
DO 400 NI=1,N1
IF(SVR(NI) .NE. 0) GOTO 400
ADIST=DIST(NSCI+NI)
IF(ADIST .LE. AMAXD) GOTO 400
AMAXD=ADIST
C UPDATE LONGEST DIST. FROM NCC
II=NI
C UPDATE FARDEST SYS. TFAMN.
400 CONTINUE
NS1=II THE FARDEST SYSTEM TERMINATION
TRF5=TRF5 + TRAFD(NS1) + TRAFIT(NS1)
SVR(NS1)=NRFG
NUMPR(NREG)=NUMPR(NREG)+1
IF(TRF5 .GT. TPR1) GOTO 707
GOTO 7021
5000 CONTINUE
IF(IACTN(NREG)=1) EQ. 1) GOTO 909
C SUM UP TRAFFIC IN THIS REGION
C
DO 702 I=1,N1
IF(SVR(I) .NE. NRFG) GOTO 702
TRFS=TRFS+TRAFD(I)+TRAFIT(I)
ADIST=DIST(NSCI+I)
IF(ADIST .GT. AMAXD) IFF=1
702 CONTINUE
CALL FINDD51,N52)
IF(N52 .EQ. 0) GOTO 900
SVR(N52)=NRFG
NUMPR(NREG)=NUMPR(NREG)+1
TRFS=TRFS+TRAFD(N5P)+TRAFIT(N52)
IF(NREG .EQ. 0) NR1 GOTO 7021
TRFS=TRFS+TRP+TPR1 GOTO 707
GOTO 7021
707 CONTINUE
TPR1=TPR1-TPR5 UPDATE REMAINING TRAFFIC
ANR1=ANR1-1
TPR1=TPR1/ANR1 UPDATE AVERAGE TRAFFIC PER REGION
TPRL=TPR1*(1. -ZETA1) UPDATE LOWER LIMIT
909 CONTINUE
GOTO 703
726 CONTINUE
C
C ONE REGION CASE
DO 727 NN=1,N1
SVR(NN) = 1
727 CONTINUE
NUMPR(1) = N1
703 CONTINUE
C
C SELECT REGIONAL SWITCHING CENTER
C
DO 500 J=1,N1
WCASE = 1.0E12
NMRR = 0
DO 505 K=1,N1
IF(SVR(K) .NE. J) GO TO 505
NMRR = NMRR + 1
SVR(NMRR)=INXPT(K)
505 CONTINUE
500 CONTINUE
C
C PRINT OUT PID AND NAMES FOR SYSTEM TERMINATIONS IN THE REGION J
C
WRITE(IWT,101A) J,(NUMPR(I), (NUMP(I),I=1,4), I=1,NMRR)
C
C PRINT OUT INDICES OF SYSTEM TERMINATIONS IN THE REGION J
C
WRITE(IWT,1028) (NUMR(I),I=1,NMRR)
C
IF(NRSC(J) .NE. 0) GO TO 501 "SELECTED"
DO 520 K=1,NMRR
NN1 = NUMR(K)            "ASSUMED RSC"
SUMT = 0.0
DO 530 L=1,NMRR
NN2 = NUML(L)
SUMT = SUMT + (TRAFFN(NN2) + TRAFFT(NN2)) * DST(NN2,NN1)
530 CONTINUE
IF(SUMT .LT. WCASE) GO TO 520
WCASE = SUMT
NRSC(J) = NN1
520 CONTINUE
501 CONTINUE
C
WRITE(IWT,1003) INDXPT(NN1), J
IF(IACTH(J,2) .EQ. 1) "OPTIMIZATION IS REQUIRED"
CALL RGMET(J,NMRR,NUMR,160,NMRR)
500 CONTINUE
C
C GENERATE INTER-REGION ORIGIN-DESTINATION MATRIX
C
C INITIALIZATION
C
IF(NN1 .LE. N) GOTO 551
DO 902 K1=1,N7
KKK=BASE(K1)
902 CONTINUE
C
A-7
IF(KKK EQ. 0) WRITE(6,7777) K
NBASF(K1)=VR(KKK)
CONTINUE
DO 609 K1=1,NR1
DO 609 K2=1,NR1
TRM(K1,K2)=0.
TRM(K2,K1)=0.
CONTINUE
DO 900 J=1,NR1
NMAR = 0
DO 905 K=1,NR1
IF(SVR(K) .NE. J) GO TO 905
NM2=NBASE(KK) ORGENTAL TRAFFIC FOR KK'S DATA PAGE
TRM(J,NM2)=TRAFF(K,J,2+KK)+TRM(J,NR1) QUITING TRAFFIC
TRM(NM2,J)=TRAFF(K,J,1+KK)+TRM(NR1,J) QUITING TRAFFIC
CONTINUE
CONTINUE
DO 905 K=1,NR1
NM2=NR5C(J1)
NM3(J,J1) = DIST(NR1,J1)
CONTINUE
CONTINUE
NTURN=NRI/10+1
DO 535 LL=1,NTURN
LL=LL-1*10+1
LUL=10
IF(LU .LT. NR1) LU=NR1
WRITE(IWT,1030) NR1,NR1,(K,K=LL,LU)
DO 1022 I=1,NR1
WRITE(IWT,1021) I,(TRM(I,J),J=LL,LU)
CONTINUE
CONTINUE
DO 535 LL=1,NTURN
LL=LL-1*10+1
LUL=10
IF(LU .LT. NR1) LU=NR1
WRITE(IWT,1030) NR1,NR1,(K,K=LL,LU)
DO 1022 I=1,NR1
WRITE(IWT,1021) I,(TRM(I,J),J=LL,LU)
CONTINUE
CONTINUE
C CALL TNOP(S,NLIMT,TAM)
74 FORMAT(//40X,TOTAL TRAFFIC=,F9.2)
75 FORMAT(//55X,TOTAL SYSTEM TRAFFIC=,F9.2)
220 FORMAT(1ASSUME NUMBER OF REGIONS=)
* / 1 ENTER NR AND STRIKE RETURN KEY
735 FORMAT(I3)
88A FORMAT(1X,*TYPE IN PRELOADED SYSTEM TRFHN. AND PSC WITH")
1 /1X,FORMAT(I1,X,A4,5)
800 FORMAT(I1,X,A4,5)
804 FORMAT(I10X,I1,X,A4,5)
3 FORMAT(I1,X,A4,5 DISTANCE ITEMS ARE OVERS?END)
210 FORMAT(I1,X,A4,5 ASSUME A SYSTEM CENTROID")
1 / ENTER CONF FOR NSEC AND STRIKE RETURN KEY
734 FORMAT(A4)
4013 FORMAT(" THE GIVEN SYSTEM CONF. CENTROID IS NOT OK, RETYPE IT")
77-53, Vol. IV

456 2101 FORMAT(*X*,1X,'INPUT TOTAL NO. OF TRANSACTIONS AND NO. OF ACCESS ***
457 1 *AT THE SWITCHER****/1X,'ENTER WITH FA.5 AND 13 UNDER XSAC/SEC')
458 2102 FORMAT(F6.5,13)
459 2103 FORMAT(*X*.*READ IN LIMITS ON NO. OF SYS. TERM. ON A INF***/
460 1 *RESPONSE TIME REGD AND NO. OF PROCESSORS WITH FORMAT ***
461 2 */1X,F5.2,12*)
462 2104 FORMAT(F3*5.2,12)
463 2105 FORMAT(*X*.*IF PLOTTING IS REQUIRED: TYPE 1 WITH FORMAT 1X**)
464 110 FORMAT(*X*.*THE TRAFFIC MATRIX(PPS) *')
465 111 FORMAT(*X*.*REAP IN LIMITS ON NO. OF SYS. TERMS *')
466 112 FORMAT(*X*.*A 3*F5.2 *)
467 113 FORMAT(*X*.*TERMS A 3*F5.2*)
468 114 FORMAT(*X*.*TOTAL TRAFFIC ORIGINATED FROM SYS. TERMINAL ***
469 115 FORMAT(*X*.*TOTAL TRAFFIC ORIGINATED TO SYS. TERMINAL ***
470 116 FORMAT(*X*.*H'Y DATA BASE IS RTN AS A SYSTEM TERMINAL ***
471 117 FORMAT(*X*.*INITIAL INTERREGION DISTANCE MATRIX ***
472 118 FORMAT(*X*.*THE NEXT SYSTEM TERMINATION M WHICH IS CLOSEST TO N ***
473 119 FORMAT(*X*.*WHERE M HAS NOT BEEN ASSIGNED TO ANY REGION YET ***
474 120 FORMAT(*X*.*CURRENT ACTION INDICIES FOR EACH REGION ***
475 121 CONTINUE
476 STOP
477 SUBROUTINE FINDM(N,M)
478 C ***************
479 C FIND THE NEXT SYSTEM TERMINATION M WHICH IS CLOSEST TO N
480 C WHERE M HAS NOT BEEN ASSIGNED TO ANY REGION YET
481 C ***************
482 AMIN=20000.
483 M=0
484 DO 70A K=1,N
485 IF(SW(K).NE.0) GOTO 70B
486 ADIST=DIST(N,K)
487 TF(ATIST .GE. AMIN) GOTO 70B
488 AMIN= ADIST
489 M=K
490 CONTINUE
491 RETURN
492 SUBROUTINE CREADK
493 C ***************
494 C READ IN ACTIONS REGARDING INSPECTIONS OF SYSTEM TERMINATIONS
495 C TO PRELOADED REGIONS AND REGIONAL NETWORK OPTIMIZATIONS
496 C ***************
497 WRITE(*,6,94)
498 6 94 FORMAT(' TYPE IN ACTION INDICIES FOR EACH REGION ***
499 1 */1X,*1ST ELEMENT: 1= INSERTION TO THIS PRELOADED REGION IS OK ***
* / * 2ND ELEMENT 1 = OPTIMIZATION IS NEEDED */
514 DO 200 NN1=1,NN1
515 DO 200 NN2=1,2
516 IACTH(NN1,NN2)=0
517 CONTINUE
518 WRITE(6,206)
519 206 FORMAT(* TYPE IN REGION INDEX AND ACTION NUMBER NEEDED, */
520 * / * WITH FORMAT 212 AND END IT WITH A 0 0 */
521 CONTINUE
522 READ(5,201) NREG, NCODE
523 FORMAT(2I12).
524 IF(NREG .LE. 0) GOTO 255
525 IF(NREG .GT. NCODE .OR. NCODE .GT. 2) GOTO 260
526 IACTH(NREG,NCODE)=1
527 GOTO 255
528 260 CONTINUE
529 WRITE(6,202)
530 202 FORMAT(* PLEASE RTYPE THE INPUT */
531 GOTO 255
532 CONTINUE AND MORE INPUT
533 RETURN
534 SUBROUTINE CREA(N1)
535 C
536 C
537 C FUNCTIONS OF THIS SUBROUTINE ARE TO
538 C 1. RECEIVE TOTAL NO. OF SYSTEM TERMINATIONS, DATA BASES AND CITIES
539 C 2. RECEIVE CITY LOCATIONS (V A H)
540 C 3. RECEIVE PID NO., SYS. TERM, NAMES, AND MAPPINGS AND TRAFFICS
541 C
542 C
543 WRITE(6,11)
544 11 FORMAT(* TYPE IN NO. OF SYS. TERMINATIONS, DATA BASES AND CITIES */
545 * / *WITH FORMAT 315 */
546 READ(5,10) NIN7,NCITY, N NUMBER OF SYSTEM TERMINATIONS
547 WRITE(6,12) NIN7, NCITY
548 10 FORMAT(2I10)
549 12 FORMAT(2X)
550 * CITIFS */ * TYPE IN DATA BASE LOCATIONS WITH FORMAT 6(IV* A4) */
551 READ(5,15) (NBASE(I),I=1,N7)
552 15 FORMAT(6(1V* A4))
553 WRITE(6,16) N7, (NBASE(I)+I=1,N7)
554 16 FORMAT(6I1)
555 WRITE(6,17) N7
556 17 FORMAT(6I1)
557 FORMAT(* DATA BASES ARE AT * A(2Y* A4) */
558 WRITE(6,18) N7
559 18 FORMAT(6I1)
560 FORMAT(* TYPE IN CITY V H WITH FORMAT (4Y* A5,2X,15))
561 READ(5,19) (IVERT(I),I=1,NCITY)
562 19 FORMAT(33X15)
563 WRITE(6,20) N7
564 20 FORMAT(6I1)
565 FORMAT(* TYPE IN PID NO., NAME, MAPPING ANR */
566 * / * TRAFFIC */ / * WITH FORMAT 14, 1X, A9, 14, 6F, A2 */
567 READ(5,801) (NAMFS(I),J=1,N7), PAN, MAPADP(1), (TRAFD(I,K,L),
568 * K=1,2,L=1,17))
569 INXPT(I)=I
570 CONTINUE
571 801 FORMAT(A4,1X,346,A4,12,14,4F10.2)
572 CONTINUE
573 RETURN
574 SUBROUTINE CREA(N2)
CREATE A RATE APPLICATION MATRIX IRATEJINP=N2)

WRITE(6+83)

83 FORMAT(' TYPE IN NO. OF RATE STRUCTURES UNDER*
84 * // CONSIDERATION WITH FORMAT 13*)
85 RFAD (5+50) N2
86 WRITE(6+84)

84 FORMAT(' TYPE IN RATE APPLICATION TO EACH COMPO.*',
85 * // WRT EACH SYS. SYR. WITH FORMAT 1012*)
86 DO 11 I=1+N2
87 READ (5+100) (IRATEJ(J),J=1,N2)
88 11 CONTINUE
89 WRITE(6+71)

71 FORMAT(' READ IN TRAFFIC DENSITY TYPE AND RATE STRUCTURE*,
72 * // FOR EACH CITY WITH FORMAT A011*)
73 READ(5+72) ((TRANP(I,J),J=1,2),I=1+NCTY)
74 72 FORMAT((A011))
75 50 FORMAT (13)
76 100 FORMAT (1012)
77 RETURN
78 SUBROUTINE CREATE(N3)
79 C
80 4 FORMAT ' TYPE IN NO. OF RATE STRUCTURES UNDER*
81 RFAD (5+50) N3
82 WRITE(6+86)

86 FORMAT(' TYPE IN NAME, CAPACITY, UTILIZATION FACTORS AVAIL. FOR * ',
87 * // EACH LINE TYPE WITH FORMAT A6+1X+16,1X+F3,2P(1X,*))
88 DO 12 I=1+N3
89 READ(5+100) LINAME(I), LINCAP(I), UTIL1, UTIL2, UTIL3, TDUPY(I)
90 12 CONTINUE
91 50 FORMAT (13)
92 100 FORMAT(A6+1X+16,1X+F3,2P(1X,1))
93 RETURN
94 SUBROUTINE CRFADD(N4)
95 C
96 87 FORMAT(' TYPE IN NO. OF DEVICES AND NAMES FOR EACH LINE TYPE*,
97 * // WRT EACH LINE TYPE WITH FORMAT 13/10(A6+1X,*)')
98 RFAD (5+50) N4, (NAMEHW(I),I=1,N4)
99 WRITE(6+88)

88 FORMAT(' TYPE IN INST. AND RECURR. COSTS WRT*,
89 * // RATE STRUCTURE, LINE TYPE, DEVICES, TRAFFIC DENSITY*,
90 * // AND IMPRINTING M0DE WITH FORMAT 2F9,2/2F9,2*)
91 C
A-11
**77-53, Vol. IV**

627 DO 13 IRATE=1,N2
628 DO 13 ILINE=1,N3
629 DO 13 IDNSTY=1,N4
630 CONTINUE
631 READ(5,100)((AINSTC(IRATE,ILINE,IDNSTY),J=1,2),I=1,9)
632 CONTINUE
633 13 FORMAT((2F9.2/2F9.2))
634 50 FORMAT(13/10(A6,1X))
635 RETURN
636 SUBROUTINE CREAD
637 ***************
638 C
639 C
640 C CREATE A MATRIX OF BASIC INSTALLATION AND RECURRING COSTS FOR
641 C LINES. COST MAY OR MAY NOT BE A LINEAR FUNCTION OF DISTANCE
642 C
643 C
644 WRITE(6,89)
645 A9 FORMAT(9 TYPE IN INST. COSTS FOR LINES WRT *:
646 1 /// RATE LINE, DENSITY, AND DURALINKING MODF **,
647 2 /// WITH FORMAT 4F9.2**)
648 WRITE(6,90)
649 90 FORMAT(9 TYPE IN INDEX FOR LINEARITY OF LINES RECUP. COST**
650 1 /// FUNCTION WITH I=LINEAR AND NONLINEAR OTHERWISE**
651 2 /// WITH FORMAT 11 FOR EACH LINE TYPE**)
652 WRITE(6,91)
653 91 FORMAT(9 TYPE IN RECUP. COSTS WITH FORMAT 4F9.2 IF LINEAR **
654 1 /// WITH FORMAT 10FA,3/10FA,3 IF NONLINEAR, **
655 2 /// IF NONLINEAR, USE 10FA,2**)
656 DO 14 IRATE=1,N2
657 DO 14 ILINE=1,N3
658 READ(5,200) INDEX
659 IFLAG(IRATE,ILINE)=INDEX QLINF COST LINEARITY INDICATOR
660 DO 14 IDNSTY=1,3
661 READ(5,100)((AINSTC(IRATE,ILINE,IDNSTY,J),J=1,2),I=1,9)
662 IF (INDEX,JF.I) GO TO 3
663 C LINEAR COST FUNCTION
664 READ(5,100)((RECLN(IRATE,ILINE,IDNSTY,J),J=1,2),I=1,9)
665 GO TO 14
666 C CONTINUE A NONLINEAR COST FUNCTION
667 READ(5,401)((RECLN(IRATE,ILINE,IDNSTY,J),J=1,16),I=1,9)
668 CONTINUE
669 CONTINUE
670 CONTINUE
671 CONTINUE
672 CONTINUE
673 CONTINUE
674 C
675 C
676 C FIND LOCAL INDEX FOR SYSTEM TERMINATION WITH YD NL
677 C
678 C
679 LOCAL=0
680 IF(NL.FG.0) RETURN
681 DO 400 NH=1,N1
682 IF(ININDEX(NL).FG.NL) GOTO 400
683 CONTINUE

A-12
RETURN

LOCAL=NN
RETURN

SUBROUTINE OVERFL(J,K)

***************
C
***************
C
IF(IOVER1.GE.NMAX) GOTO 8000
CALL PACK (J,511,1,DSTNCE)
IVRD(IOVER1+1)=J
IVRD(IOVER1+2)=K
IOVER1=IOVER1+3
RETURN

8000 CONTINUE
WRITE(6,8001)
FORMAT(2X,'THE OVERFLOW TABLE HAS BEEN FULLY LOADED**',
+15X,'PLEASE INCREASE ITS SIZE**')
STOP

SUBROUTINE CREATE
C
***************
C
C RECEIVE DATA FOR RESPONSE TIME CALCULATION
C
C 4MSLIN(NPO+2)=INPUT MSG LENGTH AS A FUNCTION OF TYPE AND PRIORITY
C MSOUT(NPO+2)=OUTPUT MSG LENGTH AS A FUNCTION OF TYPE AND PRIORITY
C AMSL(T)=AVERAGE MSG LENGTH FOR EACH TYPE
C 1=POLLING 2=NAK RESPONSE 3=INPUT MSG WITH PRIORITY 1
C 4=INPUT MSG WITH PRIORITY 2
C 5=OUTPUT MSG WITH PRIORITY 1
C 6=OUTPUT MSGS WITH PRIORITY 2
C 7=ALL MSGS
C TIXMT(NP3)=AVERAGE TRANSMISSION TIME FOR ABOVE ITEMS
C RATPRI(NPO+2)=INPUT MSG DISTRIBUTION AND OUT-GOING MSG RATIO BY PRIORITY
C N+1.1 = PERCENT OF OUTPUT MSG SENT TO WITH PRIORITY 1 IF ITS TYPE IS N
C N+1.2 = PERCENT OF OUTPUT MSG WHOSE DESTINATION IS OUTSIDE OF
C N+1.3 = PERCENT OF OUTPUT TRAFFIC DISTRIBUTION AS A FUNCTION OF TYPE AND PRIORITY
C N+1.4 = OUTPUT TRAFFIC DISTRIBUTION AS A FUNCTION OF TYPE AND PRIORITY
C
C ****************
C DIMENSION MSLIN(NPO+2),RATIO(NPO+2),RATIO(NPO+2),
C 1=MSLIN(NPO+2),MSRHM(NPO+1),
C 2=MSL(NP3),NPL(NP3),NPL(NP3),NPL(NP3),NPLH(NP3),NPLH(NP3),
C 3=MNH(NP3),TAMDH(NP3),TAM(NP3)
C WRITE(6,771)
C 771 FORMAT(1X,'TYPE IN NPL=',NPLH,NPLH,'NAK=',NPLH)
C 730 * /+ TAMD, TAD IN FORMAT (514,'2F7.5')
C 731 READ(5,771) (NPL(I),NAK(I),NPLH(I),NPLH(I),
C * MHI,TAMHI,TAPN(I),I=1,N3)
C 732 WRITE(6,772) NPL(NP3),NPL(NP3),NPL(NP3),NPL(NP3),NPLH(NP3),NPLH(NP3),
C * MHI,TAMHI,TAPN(I),I=1,N3)
C 733 FORMAT(1X,'NPL=',NPLH,'NAK=',NPLH,'NPLH=',NPLH,'NPLH=',NPLH)
C 734 READ(5,772) (NPL(I),NAK(I),NPLH(I),NPLH(I),NPLH(I),NPLH(I),
C 735 /+ TAMD, TAD IN FORMAT (514,'2F7.5')
C 736 NPLH,NPLH,'NAK=',NPLH,'NPLH=',NPLH,'NPLH=',NPLH)
C 737 WRITE(6,772)
C 772 FORMAT(1X,'TYPE IN NO. OF MSG TYPES, AND TRAFFIC STATISTICS**
C A-13
1: /// SUCH AS MSGNAME, MSGLIN, MSGOUT, RATIO WITH, 
2: /// FORMAT '4/(A6.2(2I4.F6.3))' 
3: READ(5,77) NTP
4: READ(5,179) (MSGNAME(I),MSGLIN(I),MSGOUT(I),)
5: * RATIO(I,J),I=1,J=1,NTP)
6: ❖ FORMAT('4/(A6.2(2I4.F6.3))') 
7: READ(5,81) CFHI AVG
8: ❖ FORMAT(F7.4)
9: ❖ C
10: ❖ C CALCULATE AVERAGE MSG LENGTH
11: ❖ C
12: DO 61 I=3,7
13: AMSL(I)=0.
14: 61 CONTINUE
15: DO 50 I=1,4
16: ASIM(I)=0.
17: 50 CONTINUE
18: DO 62 J=1,NTP
19: J1=J+6
20: J2=J+6
21: AMSL(J1)=AMSLL(J1)+MSGLIN(J1)*RATIO(I,J)
22: AMSL(J2)=AMSLL(J2)+MSGOUT(J1)*RATIO(I,J)
23: ASUM(J)=ASUM(J)+RATIO(I,J)
24: 62 CONTINUE
25: 66 CONTINUE
26: 67 CONTINUE
27: 68 CONTINUE
28: 69 CONTINUE
29: 70 CONTINUE
30: 71 CONTINUE
31: 72 CONTINUE
32: 73 CONTINUE
33: 74 CONTINUE
34: IF(ASUM(4).GT.0.0) GOTO 69
35: 75 CONTINUE
36: 76 CONTINUE
37: 77 CONTINUE
38: 78 CONTINUE
39: 79 CONTINUE
40: 80 CONTINUE
41: WRITE(10,105) (AMSLL(I),I=3,7)
42: 105 FORMAT(' /// SIZE AVG. INPUT MSG WITH PRIO 1=*,F6.1, CHARS',',
43: 1 /// SIZE AVG. OUTPUT MSG W/ PRIO 1=*,F6.1, CHARS',',
44: 2 /// SIZE AVG. OUTPUT MSG W/ PRIO 2=*,F6.1, CHARS',',
45: 3 /// OVERALL AVG. MSG =*,F6.1, CHARS',',
46: 4 /// OVERALL AVG. MSG =*,F6.1, CHARS')
47: DO 65 K=1,N3
48: AMSLL(I)=NPL(K)
49: AMSL(K)=NKL(K)
50: 65 CONTINUE
51: 66 CONTINUE
52: BSUM=ASUM(3)+ASUM(4)
53: RETURN
SUBROUTINE CWAITC

C

C PPF-CALCULATE CPU WAIT TIME

C

C

RHOCPU=XSAC*CPUAVG/MPROC

WRITE(100,ASO) RHOCPU

IF(RHOCPU .LE. .R) GOTO 851

IF(RHOCPU .LE. .R) GOTO 851

IF(RHOCPU .LE. .R) GOTO 851

STOP

RETURN

END
**77-53, Vol. IV**

5192#STACOM(1), RNHFET/8777

1 C SUBROUTINE RNHFET(JPREG, NPREG, NUMP, 100, NUMS)
2 C
3 C
4 C DEVELOP A REGIONAL MULTIPORT NETWORK, STARTING WITH A STAR
5 C NETWORK AND THEN OPTIMIZE IT BY ESI-ADM WILLIAMS METHOD, GIVEN
6 C THE FOLLOWING AUGMENTS:
7 C
8 C JPREG= THE INDEX FOR THE REGION UNDER CONSIDERATION
9 C NPREG= THE NUMBER OF SYSTEM TERMINATIONS IN REGION JPREG
10 C NUM= AN ARRAY THAT CONTAINS INDICES FOR ALL SYSTEM
11 C TERMINATIONS IN REGION JPREG
12 C IGO= 1 IF NETWORK OPTIMIZATION IS TO BE PERFORMED
13 C
14 C NOTE: NODE AND SYSTEM TERMINATION ARE EXCHANGEABLE
15 C
16 C
17 C PARAMETER NP1=130, NP2=1, NP3=4, NP4=3
18 C PARAMETER NP=4
19 C PARAMETER NMT=100, MW=4, NPC=360
20 C PARAMETER NP6=(NPC*NPC/2-NPC+1)/4+1
21 C COMMON/CONST/ N1,H1,P,H2,N4,M5,N7,AcT
22 C * /REF/INP(1), TRAF(NP1:2+NP2), DSNCF(NP6), MAPADP(NP1)
23 C * /LINCH/ LINMIX(NP1), LTNCAP(NP3), UTIL17(NP3)
24 C * /INF/ IRATF(JP2,NP2), IRAND(NP1:2), IFLAG(NP2,NP3)
25 C * /ET/ SR(HP1), RSC(MW), NUMD(MW), TRAF(NP1), TRAFIT(NP1)
26 C * /NAME/NAMET(NP1:4), LNAME(NP3), NAMEHW(NP4)
27 C * /SUM/ ASUM(N1), SUM
28 C * /M/ S(16)
29 C * /ROUND/ NTERMSP,TIMEPO, MPKCH, MPLOT
30 C * /ADD/ IADD(NP1), KCHO, KAND
31 C * /T/ RSH(4), PSTIM
32 C DIMENSION COSTEW(NP1:4), IAPRAY(NP1:5), APAY(NP1:2)
33 C DIMENSION TMRSP(NP1), TRFSUM(NP1:2), TIMMCST(NP1)
34 C DIMENSION MLINE(NP1:3), LDMY(NP1), NHP1(1)
35 C DIMENSION IOST(JP1), LNSHM(NP4, NP1), HNMP(NP4, NP1)
36 C DIMENSION ICSTHW(NP1), NMP(NP1:2), ICSTLN(NP1:2), ICSTLN(NP1:2), TCHT(NP1:2)
37 C DIMENSION ISUB(NP1), NMP(NP1:3), NMP(NP1)
38 C DIMENSION S(HJ1), JCHAR(2)
39 C DIMENSION NUMR(1)
40 C DIMENSION RSH(NP1)
41 C DIMENSION RSH(NP1)
42 C EQUIVALENCES (JCHAR, ICHAR)
43 C DATA JANCH(1), / 
44 C
45 C RSPTIME=0.
46 C IPOINT=0.
47 C INTEGER TCOST1, TCOST2, COST, COSTF
48 C
49 C INITIALIZATION COST ARRAY, INDEPENDENT OF LINF TYPE
50 C DO 399 K1=1:NP1
51 C DO 399 K2=1:NP2
52 C ICSTLN(K1,K2)=0
53 C ICSTHW(K1,K2,K3)=0
54 C CONTINUE
55 C NN1=RSC(1) GLOBAL INDEX FOR RSC
56 C
57 C
C FIND THE LOCAL RSC INDEX IN THE REGION ARRAY
57
58
59
60
61
62
63
64
C
65
C BUILD A STAR NETWORK
66
C
67
C CALL STAREW
68
C
69 C PRINT OUT STAR NETWORK
70
C
71
C CALL SUMPRN(NOREGN,1) IF(160.LT.0) GOTO 979
72
C
73 C DEVELOP A MULTIDROP NETWORK UTILIZING THE
74 C ESAU-WILLIAMS ALGORITHM
75 C
76
77 MAXSAV=0
78 MAXM=0
79 MAXL=0
80 MAXK=0
81 MAXI=0
82 MAXIN=0
83 MAXNOL=0
84 LINNEW=0
85 RSPMAX=0.
86 RHOMAX=0.
87 ICHAR=1
88 ITALLY=0
89 JTALLY=0
90 KCHG=1
91 KADD=1
92 IOK=0
93 INTRY=0
94 C ESSWIL
95 WRITE(*,933) ITALL,ITALLY
96 FORMAT(*,'TRYLNK HAS BEEN ACCEPTED FOR ',I9,' TIMES',
97 ' ',/2X,'UPNETW HAS BEEN ACCEPTED FOR ',I9,' TIMES')
98 CONTINUE
99 RETURN
100 C SUBROUTINE STAREW
101 ***************
102 C
103 C FORM THE INITIAL REGIONAL STAR NETWORK, IARRAY, AND FIND ITS
104 C COST, COSTW
105 C NOREGN=NUMBER OF SYSTEM TERMINATIONS IN THE REGION
106 C
107 ***************
108 C INTEGRATE COST
109 DO 110 K3=1, NOREGN
110 110 IARRAY(K3+K4)=0
111 CONTINUE
112
113 KK=NMR(K3)

A-17
114      IARRAY(K3;1)=IAND(KK)
115      IARRAY(K3;5)=IRSC  G LOCAL INDEX FOR RSC
116      ARRAY(K3;1)=TRFIN(KK)
117      ARRAY(K3;2)=TRFOUT(KK)
118      TIMRSP(K3)=0.
119  100  CONTINUE
120      IARRAY(IRSC;1) = NOREGN = 1 GND. OF NODES UNDER RSC
121      NM=0. 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(KS;1) = NM
133          IARRAY(NM+4) = KS
134          200  CONTINUE
135      C
136      C DETERMINE LINE TYPE FOR CENTRAL LINKS TO RSC
137      C
138      DO 550 NONF=1, NOREGN
139          IF(NODE .EQ. IRSC) GOTO 555
140          TRFIN=ARRAY(NODE;1)+6.5
141          TRFOUT=ARRAY(NODE;2)+6.5
142          NN2=NUMR(NODE)
143          DSTD=DIST(NODE)*NN2
144          COSTW=NODF1*DSTD
145          COSTW(NODE;4) =DSTD
146      550  CONTINUE
147      C
148      C TAKE A FIRST GUESS FOR LINE CONFIGURATION
149      C
150          COST=0
151          RH=0.
152          MDOR=IARRAY(NONF;1)+1
153          CALL LINNUM(TRFIN,TRFOUT,LDIMMY,LINOLD,0,RHO)
154  781  CONTINUE
155      C
156      C COMPUTE INITIAL RESPONSE TIME
157      C
158          IKONT=0
159          DO 783 NL=1,N3
160              IF(LINNUM(1).NE. 0) IKONT=IKONT+1
161          783  CONTINUE
162          AINTRF=TRFIN
163          OUTTFR=TRFOUT
164          IF(IKONT .EQ. 1 .AND. LDIMMY(LINOLD) .EQ. 1) GOTO 772
165      C
166      C RESPONSE TIME CALCULATION NEEDS MODIFICATION
167      C
168          ACAP=0.
169          DO 771 NL=1,N3
170              ACAP=ACAP+INCAP(NL)*LDIMMY(NL)
171  772  CONTINUE
LDUMMY(NL)=0
CONTINUE
LDUMMY(LINOLD)=1
AINTRF=TRFIN+LINCAP(LINOLD)/ACAP OPERROR TRAFFIC
OUTRF=TRFOUT+LINCAP(LINOLD)/ACAP
CONTINUE
CALL RSPNSE(AINTRF,OUTRF,LINOLD,NODE,10K)
IF(TOK=EQ.1) GOTO 775
IF(LINOLD=EQ.N3) GOTO 774
LDUMMY(LINOLD)=N
LINOLD=LINOLD+1
GOTO 775
CONTINUE
NLL=0
N33=N3-1
DO 776 I=1,N33
IF(LDUMMY(I)=EQ.0 ) GOTO 776
NLL=I
GOTO 780
CONTINUE
LDUMMY(NLL)=0
LDUMMY(NLL+1)=LDUMMY(NLL+1)+1
CONTINUE
CALL RHOFUN(TRFIN,TRFOUT,LDUMMY,LINOLD,RHINT+RHO)
GOTO 781
CONTINUE
TIMESP(NODE)=RSPTIM
KCHAF=2
CALL ISINPUT(IRSC,NODE+2,COST)
RMOF(NODE)=RHO
COSTEW(NODE+1)=COST
COSTFW(NODE+2)=LINOLD
DO 499 NL=1,N3
NLINES(NODE+NL)=LDUMMY(NL)
DO 499 NM=1,N4
ICSTLN(NODE+NM)=ICSTLN(NODE+NM)+1
NLKCHL(NL+NM)=NLKCHL(NL+NM)+1
ICSTHW(NODE+NM+1)=ICSTHW(NODE+NM+1)+1
CONTINUE
JTRAF=TRFIN+TRFOUT
JTRAF=JTRAF/UTILIZ(LINOLD)
COSTFW(NODE+3)=JTRAF/LINCAP(LINOLD)+1
GOTO 555
CONTINUE
C ASUMING TRAFFIC AT IRSC IS TAKEN CARE OF AUTOMATICALLY
C
DO 498 NL=1,N3
NLINES(NODE+NL)=0
CONTINUE
C
555 COSTFW(NODE+1)=0
C
556 COSTFW(NODE+2)=0
C
557 COSTFW(NODE+3)=0
C
A-19
77-53, Vol. IV

228 COSTFW(NODE+4)=0
229 RHOF(NODE)=0.
230 550 CONTINUE
231 RETURN
232 SUBROUTINE ISUMUP(L1,L2,LT,IC)
233 ***************
234 C C CALCULATE COST BETWEEN NODES L1 AND L2 AND ADD IT TO
235 C TOTAL COST IC WHERE LT=LINE TYPE
236 C C ***************
237 C
238 C
239 LL1=NUM(L1)
240 LL2=NUM(L2)
241 CALL ICOSTFW(LL1,LL2,LNKC,W1,NKCLN)
242 KK=N3
243 IF(LT.LT.0) KK=1
244 DO 211 LINTYP=1, KK
245 LTYPE=LINTYP
246 IF(KK.EQ.1) LTYPE=LT
247 IC=IC+LNKC(W1,LTYPE,NKCLN)
248 IF(KK.EQ.1) 222 I2=1, N4
249 IF(KK.EQ.1) 222 I2=1, N4
250 IC=IC+LNKC(W1,LTYPE,NKCLN)
251 222 CONTINUE
252 211 CONTINUE
253 211 CONTINUE
254 RETURN
255 SUBROUTINE ESWIL
256 ***************
257 C C TRY AGAIN TO OPTIMIZE THE NETWORK
258 C C ***************
259 C
260 C
261 560 CONTINUE
262 K=ARRAY(RSC+2) QFIRST SUBNET UNDER RSC
263 KNEXT=ARRAY(K+3) QNEXT SUBNET UNDER RSC
264 IF(KNEXT.EQ.0) 60 60 TO 599 ONLY ONE SUBNET
265 560 CONTINUE
266 570 CONTINUE
267 L=ARRAY(RSC+2) QSUBNET IS TO PF LINKED TO L=SUBNET
268 570 CONTINUE
269 IF(L .NE. K) GOTO 575
270 L=ARRAY(L+3)
271 IF(L .NE. 0) GOTO 660
272 575 CONTINUE
273 K1=NUM(K)
274 DFF=DIST(MIN1,K1)
275 C C TRST TOTAL NO. OF TERMINALS IF K AND L ARE COMBINED
276 C
277 C IMTRY=0 INDICATION OF ENTRY TO TRYLNK
278 LINE=COSTFW(K+2)
279 IF(LINCAP(LINE)=0) 599 GOTO 599 AND MULTIDROPPING ON 599
280 NODE=ARRAY(K+1)+ARRAY(L+1)+2
281 IF(NODE .GT. NTERMS) GOTO 599 GOTO 599 MANY TERMINALS
282 M=L
283 K=K

A-20
580 CONTINUE
581 MINNUM(M)
582 DTRY=DIST(K1+M)/2.
583 IF(DTRY .GT. DREF) GOTO 140
584 CALL TPLYNK(K+K1+L/M) & M IS THE INSERTION NODE
585 IF(10K .EQ. 0) GOTO 585
140 CONTINUE
292 M=NXTNOD(K+M) (NEXT NODE UNDER M ON L-SURFET
293 IF(M .NE. 0) GOTO 580 GTO MORE NODES UNDER M ON L-SURFET
294 K=NXTNOD(K+K1) (START WITH NEXT NODE ON K-SURFET
295 IF(K .EQ. 0) GOTO 585
296 K=NUNUM(K)
297 MFL
298 GOTO 580
585 CONTINUE
300 L=IARRAY(L+3) (NEXT SUCCESSOR
301 IF(L .NE. 0) GOTO 570
302 660 CONTINUE
303 K=IARRAY(K+3)
304 IF(K .NE. 0) GOTO 560 GOTO AN END YET, REPEAT THE SEARCH
305 CALL ALL POSSIBLE COMBINATIONS HAVE BEEN TRIED
307 CALL IF(MAXSAV .LE. 0) GOTO 599 (NO NEED TO GO FURTHER
309 CALL C UPDATE NETWORK BASED ON UP-TO-DATE MAXIMUM COST SAVING
311 C PARAMETERS
312 C JTALLY=JTALLY+1
313 C CALL UPNETW
314 C C REINITIALIZATION
315 C RSPMAX=0.
316 MAXSAV=0
317 MAXK=0
318 MAXL=0
319 MAXM=0
320 MAXK1=0
321 MAXLIN=0
322 MAXN=0
323 MAXS=0
324 MAXT=0
325 MAXW=0
326 MAXV=0
327 MAXP=0
328 GOTO 5000
329 599 CONTINUE
330 C PRINT OUT COSTS FOR THE OPTIMIZED MULTIDROP NETWORK
331 C CALL MUTDRP
333 C PRINT OUT THE OPTIMIZED MULTIDROP NETWORK
336 C CALL NPTRPRT
338 IF(MPLOT .NE. 1) GOTO 50
339 CALL CALPLT
340 50 CONTINUE
341 RETURN
SUBROUTINE TRYLINK(KL, KIL, LL, ML)

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

C
C TRY TO ELIMINATE CENTRAL LINK KL AND LINK IT TO THE SUBNETWORK
C LL THROUGH SYSTEM TERMINATIONS KIL AND ML.

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

A-22
77-53, Vol. IV

399 133 GOTO 134
400 134 CONTINUE
401 ITEMP=0
402 KADD=0
403 KCHG=2
404 CALL ISUMUP(1RSC•KL+LINNEW•ITEMP)
405 MCOST=MCO5TK-ITEMP
406 134 CONTINUE
407 INTRY=1 FLAG THAT INDICATES AN ENTRY TO TRYLNK
408 JSAV=MCOSTW(1L+1)+COSTEw(KL+1)-(MCOST+MCO5TK)
409 719 CONTINUE
410 COSTKM=0
411 KADD=0
412 CALL ISUMUP(KL•KL+LINNEW•COSTKM)
413 ISAV=JSav+COSTKM
414 IF (ISAV •LE. MAXSAV) GO TO 132
415 RSPMAX=RSPMAX+ISAV
416 MAXSAV=ISAV
417 MAX=KL
418 MAX=KL
419 MAX=ML
420 MAX=ML
421 MAXL=LINNEW
422 MAX=NL
423 MAX=NL
424 132 CONTINUE
425 RETURN
426 SUBROUTINE LCOST(INA+NB+TCOST)
427 ***************
428 ***************
429 C FIND COST FOR A SUBNETWORK, NA=BEGINNING NODE FOR THE SUBNET
430 C TO BE EVALUATED.
431 C NA=1 WHEN COST FOR CENTRAL LINK NA IS TO BE INCLUDED
432 C NA=0 WHEN COST FOR CENTRAL LINK NA IS NOT TO BE INCLUDED
433 C
434 C
435 INTEGER TCOST
436 TCOST=0
437 KCHG=2
438 CALL ISUMUP(INA+LINNEW•TCOST)
439 C START COMPUTING SUBNET COST
440 C
441 C
442 JS=ARRAY(INA+2)  @ FIRST SUCCESSOR
443 IF (JS •EQ. 0) GOTO 400
444 CONTINUE
445 JS=ARRAY(INA+5)
446 CALL ISUMUP(JPA+JS+LINNEW•TCOST)
447 JS=ARRAY(INA+JS+LINNEW•TCOST)
448 IF (JS •EQ. 0) GOTO 400  @ CALL IT AN END
449 GO TO 300
450 400 CONTINUE
451 IF (INA •EQ. 1) RETURN
452 ITEMP=0
453 KADD=0
454 CALL ISUMUP(INA+LINNEW•ITEMP)
455 TCOST=TCOST+ITEMP

A-23
RETURN

FUNCTION NXTNOD(L1,M1)

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

C FIND THE NEXT NODE IN THE SUBNET L1 WHICH M1 BELONGS TO.

C IN THE PROCESS, IF THE NEXT NODE IS L1, 0 IS RETURNED.

C OTHERWISE THE NEXT NODE IS RETURNED.

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

NXTNOD=0

MM=M1

KSON=ARRAY(MM+2)

IF (KSON.EQ.0 .AND. MM.EQ. L1) RETURN 0A SINGLE NODE

IF (KSON.EQ.0) GO TO 1

0 NO SUCCESSOR

NXTNOD=KSON

RETURN

CONTINUE

LOOK FOR HIS NEXT BROTHER

KPRO=ARRAY(MM+3)

IF (KPRO.EQ.0) GO TO 2

1 NO MORE SUCCESSIONS WITH SAME PREDECESSOR

RETURN

CONTINUE

GO TO HIS FATHER

MM=ARRAY(MM+5)

IF (MM.NE. L1) GO TO 1

2 BACK TO THE BEGINNING

RETURN

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

UPDATE IARRAY AND COSTEW BASED ON MAXIMUM SAVING

PARAMETERS OBTAINED

UPDATE TRAFFIC AND NO. OF TERMINALS FOR L-SUBNET

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

NOK=ARRAY(MAXK+1)+1

IARRAY(MAX1)=IARRAY(MAX1)+NOK

IARRAY(MAX1)=IARRAY(MAX1)+ARRAY(MAX1)

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

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

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

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

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

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

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

UPDATE THE COSTEW

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

COSTEW(MAX1)=MAXLIN

COSTEW(MAX1)=MAXNOL

COSTEW(MAX1)=MAXNO

COSTEW(MAX1)=MAXNO

COSTEW(MAX1)=MAXNO

MAXK=NUMR(MAX)

MAXK=NUMR(MAX)

MAXK=NUMR(MAX)

MAXK=NUMR(MAX)

MAXK=NUMR(MAX)

MAXK=NUMR(MAX)

COSTEW(MAX1)=COSTEW(MAX1)+COSTEW(MAX1)+DIST(MAXKIN+MAXKP)
77-53, Vol. IV

513 * -DIST(MAXN+NN1)
514 RHOF(MAXL)=RHOMAX
515 COSTW(MAXL)=0
516 C
517 C UPDATE MULTIDROPPED-LINE RESPONSE TIMF
518 C
519 C TMRSP(MAXL)=RSPMAX
520 CONTINUE
521 91 KIPA=ARRAY(MAXK+5)  GREATEST KI'S PREDECESSOR
522 MSUN=ARRAY(MAXK+2)  KN'S 1ST SUCCESSOR
523 CALL LKOFF(MAXKI)  CREATE KI AS A SUCCESSOR OF KIPA
524 IARRAY(MAXK+2)=MAXKI
525 IARRAY(MAXKI+5)=MAXM
526 IARRAY(MAXKI+3)=MSUN
527 IF(MSON+NE+0) IARRAY(MSON)=MAXKI
528 IARRAY(MAXKI+4)=0
529 MAXM=MAXKI
530 MAXKI=KIPA
531 IF(MAXM+NE+MAXKI) GOTO 91
532 RETURN
533 FUNCTION JCOSTA(N+KREF)
534 C  *******************
535 C
536 C FIND PARTIAL SUM FOR ICSI1K
537 C
538 C  *******************
539 JCOSTA=0
540 DO 777 KI=1,KREF
541 JCOSTA=JCOSTA+ICST(KI,N)
542 CONTINUE
543 RETURN
544 FUNCTION JCOSTR(N,M,KREF)
545 C  **************
546 C
547 C FIND PARTIAL SUM FOR ICSITHW
548 C
549 C  **************
550 JCOSTR=0
551 DO 77A KK=1,KREF
552 JCOSTR=JCOSTR+ICSTHW(KK,N)
553 CONTINUE
554 RETURN
555 SUBROUTINE NETPRT
556 C  **************
557 C
558 C PRINT OUT CONFIGURATION OF THE MULTIDROP NETWORK
559 C
560 C  **************
561 DO 196 KK=1,NP1
562 IBLANK(KK)=JBLANK
563 CONTINUE
564 NI=NUMR(1RSC)
565 WRITE(IWI,197) NI2
566 197 FORMAT(I1,1X,REGIONAL CENTER= '1A4/16X,1SUBNETWORK',16X)
567 * *BEGINS AT',/')
568 KP=1
569 ISON=ARRAY(IRSC+2)
77-53, Vol. IV

570  IPOINT=ISON
571  ISON=NUMR(ISON)
572  WRITE(INT+192) (IRLANK(I),I=1,KN),ISON
573  C
574  C LOOK FOR ITS FIRST SUCCESSOR
575  C
576  190 CONTINUE
577  ISON=IARRAY(IPOINT+2) CURRENT NEXT INDEX
578  IF(ISON.EQ.0) GOTO 191 AND MORE SAME
579  KP=KP+1 GA LEVEL DEFLER
580  R=ISON=NUMR(R=ISON)
581  WRITE(INT+192) (IRLANK(I),I=1,KN),ISON
582  192 FORMAT(1X,2F4.0))
583  IPOINT=ISON
584  GOTO 190
585  191 CONTINUE
586  C
587  C LOOK FOR NEXT SUCCESSOR WITH THE SAME PREDECESSOR
588  C
589  IARR=IARRAY(IPOINT+3)
590  IF(IARR.EQ.0) GOTO 193
591  JARR=NUMR(IARR)
592  WRITE(INT+192) (IRLANK(I),I=1,KN),IARR
593  IPOINT=JARR
594  GOTO 190
595  193 CONTINUE
596  C
597  C NEXT LEVEL UP
598  C
599  KP=KP+1
600  IPOINT=IARRAY(IPOINT+5)
601  IF(KP.EQ.0) GOTO 194 AND Nack TO GO FURTHER
602  GOTO 191
603  194 CONTINUE
604  RETURN
605  SUBROUTINE CONVERT(ICOST)
606  ***************
607  C
608  C CONVERT A NUMBER INTO ITS FIELD EQUIVALENT
609  C
610  C ***************
611  JCHAR(I)=JRLANK
612  JCHAR(2)=JRLANK
613  IF(ICOST.EQ.0) GOTO 916
614  ENCONE(198,JCHAR) ICOST
615  19A FORMAT(18)
616  916 CONTINUE
617  RETURN
618  SUBROUTINE SUMPRTN(REF+NN)
619  C ***************
620  C
621  C SUM UP COSTS AND PRINTS
622  C
623  C ***************
624  TCOST1=0
625  TCOST2=0
626  DO 779 K=1,NNREF
ITCOST(K;1)=ICSTLN(K;1)

ITCOST(K;2)=ICSTLN(K;2)

DO 7791 K=1,N4

ITCOST(K;1)=ITCOST(K;1)+ICSTHW(K;K;1)

ITCOST(K;2)=ITCOST(K;2)+ICSTHW(K;K;2)

CONTINUE

TCOST1=ITCOST1+ITCOST(K;1)

TCOST2=ITCOST2+ITCOST(K;2)

CONTINUE

KCost=TCOST1+TCOST2

C

C PRINT OUT COST

C

NTURN=NREF/10+1

ITEM=MOD(NREF;10)

IF(ITEM.EQ.0) NTURN=NTURN-1

LPAGE=1

DO 919 KW=1;NTURN

KW=10*(KW-1)+1

KWU=10+KW

IF(KWU.GT.NREF) KW=NREF

IF(NN.EQ.0) GOTO 879

IF(LPAGE.NE.1) GOTO 9033

WRITE(IWTr9031) KW

9031 FORMAT('1X;140X;REGIONAL STAR NETWORK AND ITS COSTS;--1)1

GOTO 9035

9033 CONTINUE

WRITE(IWT+9034) KW

9034 FORMAT('1X;140X;REGIONAL STAR NETWORK AND ITS COSTS;--1)1

9035 CONTINUE

WRITE(IWT+9032) (NUMRR(I);I=KWL,KWU)

9036 FORMAT('1X;140X;SYSTEM TERMINAL;--1)1

WRITE(IWT+9033)

903 FORMAT('1X;140X;NO. OF LINES REG;--1)1

DO 1903 NJ=1,N3

IF(LINMIX(NJ).EQ.0) GOTO 1903

WRITE(IWT+904) LINAME(NJ), (NLINES(K,NJ);K=KWL,KWU)

904 FORMAT('1X;140X;1X;14X;10(1X,1X))

1903 CONTINUE

WRITE(IWT+9036) (RHOF(NJ);NJ=KWL,KWU)

9036 FORMAT('1X;140X;LINE UTILIZATION;--1)1

WRITE(IWT+906) (DSTN(C);C=KWL,KWU)

906 FORMAT('1X;140X;DISTNCF FROM SRC;--1)1

GO TO 806

879 CONTINUE

IF(LPAGE.NE.1) GOTO 8033

WRITE(IWT+8033) KW

8031 FORMAT('1X;140X;FINAL MULTIDROP NETWORK AND ITS COSTS;--1)1

GOTO 8035

8033 CONTINUE

WRITE(IWT+8034) KW

8034 FORMAT('1X;140X;FINAL MULTIDROP NETWORK AND ITS COSTS;--1)1

8035 CONTINUE

WRITE(IWT+803) (1;I=KWL,KWU)

803 FORMAT('1X;140X;SURFNET NO.;--1)1

DO 1803 NF=KWL,KWU

ID=NSUR(N)

A-27
77-53, Vol. IV

MSUB(N)=NUMARR(ID)
LSUB(N)=IARRAY(ID)+1

1800 CONTINUE
WRITE(IWT+106) (MSUB(N), N=KWL,KWU)
1806 FORMAT(5X*BEGINNING NODE**11X*10(F4.3,1X))
WRITE(IWT+1807) (LSUB(N), N=KWL,KWU)
1807 FORMAT(3X*NO. OF TERM**12X*10(IN1,1X))
WRITE(IWT+811)
811 FORMAT(3X*NO. OF LINES)
DO 1808 NJ=1,N3
IF(LININV(NJ).EQ.0) GOTO 1808
WRITE(IWT+904) LINAME(NJ) *(NLINES(K,NJ), K=KWL,KWU)
1808 CONTINUE
WRITE(IWT+908) (RHOF(NJ), N=KWL,KWU)
8036 FORMAT(3X*LINE UTILIZATIONADOR.1X*10(F4.3,1X))
WRITE(IWT+908) (IDST(N), N=KWL,KWU)
808 FORMAT(3X*TOTAL MILEAGE**12X*10(IN1,1X))
701 806 CONTINUE
702 DO 1101 N=KWL,KWU
703 N=
704 IF(NN+F0.0) INSGSUB(N)
705 TRFSUM(N+1)=ARRAY(ID+1)
706 TRFSUM(N+2)=ARRAY(ID+2)
707 TIMOUT(N)=TIMSP(ID)
708 1101 CONTINUE
709 WRITE(IWT+1102) (TRFSUM(N+1)+1=KWL,KWU)
710 1102 FORMAT(3X*TRAFFIC**/3X*LINE TO CPU**11X*10(F8.3,1X))
711 WRITE(IWT+1103) (TRFSUM(N+2)+1=KWL,KWU)
712 1103 FORMAT(3X*CPU TO LINE**/10X*10(F8.3,1X))
713 WRITE(IWT+1104) (TIMOUT(N)+1=KWL,KWU)
714 1104 FORMAT(3X*LINE RESPONSE TIME**7X*10(F8.3,1X))
715 WRITE(IWT+907)
716 907 FORMAT(3X*SUBTOTAL**/11X*INST. COSTS)
717 COST=JCOSTA(K1,1,RF)
718 IF(KW.NE.1) COST=0
719 CALL CONVERT(COST)
720 WRITE(IWT+908) (JCHAR(L), L=1,2),(ICSTLN(NODE), NODE=KWL,KWU)
721 908 FORMAT(5X*LINES**,8X,A6,A2,1X,10(F4.3,1X))
722 DO 1909 K=1,N4
723 COST=JCOSTB(K1,1,RF)
724 IF(KW.NE.1) COST=0
725 CALL CONVERT(COST)
726 WRITE(IWT+909) NAMEhw(K),(JCHAR(L), L=1,2),(ICSTHW(NODE), NODE=KWL,KWU)
727 + NODE=KWL,KWU)
728 909 FORMAT(5X*A6,7X,A6,A2,1X,10(IN1,1X))
729 1909 CONTINUE
730 WRITE(IWT+910)
731 910 FORMAT(5X*ANNUAL RECUR, COST)
732 COST=JCOSTA(2,RF)
733 IF(KW.NE.1) COST=0
734 CALL CONVERT(COST)
735 910 FORMAT(5X*A6,7X,A6,A2,1X,10(IN1,1X))
736 DO 1911 K=1,N4
737 COST=JCOSTB(K,RF)
738 IF(KW.NE.1) COST=0
739 CALL CONVERT(COST)
740 WRITE(IWT+909) NAMEhw(K),(JCHAR(L), L=1,2),(ICSTHW(NODE), NODE=KWL,KWU)

A-28
* NODE=KWL+KWU

CONTINUE

WRITE(IWT+912)

FORMAT(IX,TOTAL COST)

IF(KW+NE.1) TCOST=0

CALL CONVRT(TCOST)

WRITE(IWT+913) (JCHAR(L),L=1,2), (ITCOST(K,1),K=KW+KWU)

IF(KW+NF.1) TCOST2=0

CALL CONVRT(TCOST2)

WRITE(IWT+914) (JCHAR(L),L=1,2), (ITCOST(K,2),K=KWL+KWU)

913 FORMAT(4X,INST, COST, 4X, A6, A2, 1X, 10I9, 1X)

914 FORMAT(4X, RECUR, COST, 3X, A6, A2, 1X, 10I9, 1X)

919 CONTINUE

WRITE(IWT+695) K COST

695 FORMAT(/, 25X, TOTAL COST=, 18)

RETURN

SUBROUTINE MULTDRP

C **START**

DO 590 NL=1,N3

LDUMMY(NL)=0

590 CONTINUE

IBR=IARRAY(IRSC+2) @FIRST SUCCESSOR

K=1

699 CONTINUE

IF(IBR+EQ.0) GOTO 698

NK=NUMR(IBR)

NK1=N1

NK2=N2

N=IRK(K1)=IPRO

LINE=COSTW(IBR+2)

LDUMMY(LINE)=COSTFW(IBR+3)

JS=IARRAY(IBR+2)

IF(JS+EQ.0) GOTO 694

DO 592 NK=1,2

ICSTLN(K1+NM)=0

DO 592 NK=1,N4

ICSTHN(K1+NK+NM)=0

592 CONTINUE

DO 594 NL=1,N3

LDINES(K1+NL)=LDUMMY(NL)

596 CONTINUE

KCHG=2

312 CONTINUE

CALL COSTJ(LDUMMY,NK1+NK2+LNCM,1, NKCHN)

DO 595 NL=1,N3

DO 595 NM=1,2

ICSTLN(K1+NM)=ICSTLN(K1+NM)+LNCM(NL,NM)

DO 595 NK=1,N4

ICSTHN(K1+NK+NM)=ICSTHN(K1+NK+NM)+LNCN(NL,NK,NM)

595 CONTINUE

IF(JS+EQ.0) GOTO 311

NK2=NUMR(JSON) @GLOBAL INDEX FOR NEXT NODE.
77-53, Vol. IV

798  NK=IARRAY(JSON+N) GFDECFSSOR
799  NK1=NUM(NKI) GLOBAL INDEX FOR PRDFCCESSOR
800  JSON=NUM(loc(IARO+JSON))
801  GOTO 312
802  311  CONTINUE
803  LDUMMY(LINE)=0
804  GOTO 591
805  694  CONTINUE
806  C
807  C USE PREVIOUS DATA
808  C
809  DO 597 NL=1+3
810      NINES(KI+NL)=NINES(IARO+NL)
811  597  CONTINUE
812  DO 598 NM=1+3
813      ICSTLN(KI+NM)=ICSTLN(IARO+NM)
814  598  CONTINUE
815  ICSTHN(KI+NK+NM)=ICSTHN(IARO+MK+NM)
816  L99  CONTINUE
817  LDUMMY(LINE)=0
818  591  CONTINUE
819  INST(KI)=COSTEW(IAR0+4)
820  RHOP(KI)=RHOF(IARO) ASHUFFLING RHO'S DUE TO RF-INDEXING
821  IARO=IARRAY(IARO+3)
822     KI=KI+1
823  GOTO 699
824  698  CONTINUE
825  NOSUB=KI-1
826  CALL SUMPRAT(NOSUB+0)
827  RETURN
828  SUBROUTINE CALPLT
829  **************
830  C
831  C PLOT A MULTIDROP NETWORK
832  C
833  C  **************
834     KP=1
835     IPOINT=IRSC
836     CALL TRSFRM(2)
837     ISON=IARRAY(IARO+2) QFIRST SUCCESSOR
838     IPOINT=ISON
839     CALL TRSFRM(1)
840  C
841  C LOOK FOR ITS FIRST SUCCESSOR
842  C
843  190  CONTINUE
844     ISON=IARRAY(IPOINT+2) QFIRST SUCCESSOR
845     IF(ISON .EQ. 0) GOTO 191
846     KP=KP+1
847     IPOINT=ISON
848     CALL TRSFRM(1)
849  GOTO 190
850  191  CONTINUE
851  C
852  C LOOK FOR ITS NEXT SUCCESSOR
853  C
854     IARO=IARRAY(IPOINT+3)
855  IPOINT=IARRAY(IPOINT+5) GO TO ITS PREDECESSOR
856  CALL TRSFPM(2)
857  IF(IBRO .EQ. 0) GOTO 193
858  IPOINT=IBRO
859  CALL TRSFPM(1)
860  GOTO 190
861  193 CONTINUE
862  C
863  C GO BACK TO ITS PREDECESSOR
864  C
865  KPK=KPK+1
866  IF(KPK .EQ. 0) GOTO 194
867  GOTO 191
868  194 CONTINUE
869  CALL TRSFPM(3)
870  RETURN
871  SUBROUTINE TRSFPM(LK)
872  C ***************
873  C C FIND GLOBAL MADADR INDEX FOR V-H COORDINATES AND PID NO5
874  C
875  C
876  C
877  DATA IP/0/
878  IF(LK .EQ. 3) GOTO 666
879  LKI=NUMR(IPOINT) GLOBAL INDEX
880  IDD=MAPADR(LKI) MAPADR INDEX FOR LK1
881  IF(IDD .EQ. IP) RETURN
882  IP=IDD
883  666 CONTINUE
884  CALL PLOTPT(IDD,LK)
885  RETURN
886  SUBROUTINE LNKOFF(MP)
887  C ***************
888  C C DELETE MP AS A SUCCESSOR OF NONE PA
889  C
890  C
891  C
892  IFRONT=IARRAY(MP+4) THE SUCCESSOR REFERE MP
893  IRACK =IARRAY(MP+3) THE SUCCESSOR AFTER MP
894  IF(IRFRONT .NE. 0) GOTO 92
895  MP2=IARRAY(MP+5)
896  IARRAY(MP+2)= IRACK 1ST SUCCESSOR UNDER NEW MPA
897  GOTO 99
898  92 CONTINUE
899  IARRAY(IFRONT+3)=IRACK
900  99 CONTINUE
901  IF(IRACK .EQ. 0) RETURN
902  IARRAY(IRACK+4)=IFRONT
903  RETURN
904  SUBROUTINE RSPTST(KKK,LLE,LINMAX,1OK)
905  C ***************
906  C C TEST RESPONSE TIME SATISFIED WHEN 1OK=1
907  C
908  C
909  C
910  MDROP=IARRAY(LLE+1)+IARRAY(KKK+1)+2
911  TRFIN=ARRAY(LLE+1)+ARRAY(KKK+1)
912  TRFOUT=ARRAY(LLE+2)+ARRAY(KKK+2)
913  CALL RSPNSE(TRFIN,TRFOUT,LINMAX,MDROP,1OK)
914  RETURN
915  END

SPRT STACOM.INOP/0777
SUBROUTINE IRNOP(NR, LIMIT, TRM)

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

SUBPROGRAM FOR THE INTER-REGION NETWORK OPTIMIZATION

LIMIT: MINIMAL NUMBER OF PATHS NEEDED PER REGIONAL SWITCHING CENTER

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

PARAMETER NP1 = 130, NP2 = 1, NP3 = 4, NP4 = 3
PARAMETER NP5 = 360, NP6 = (NPC + NPC/2 - NPC + 1)/4 + 1
PARAMETER NP7 = 4, INT = 100, MW = 4
COMMON/CONST/N1, N2, N3, MW, NP7, C, N7, NCIT
COMMON/INCHR/LINMIX(NP3) + LINCAP(NP3) + UTILIZ(NP3)
COMMON/LINCA(NP3) + NAME(NP3) + NAMEH(NP3)
COMMON/NAMEXPT(NP1) + NAMEF(NP1) + NAMEH(NP1)
COMMON/LINCAF(NP1) + TRAF(NP1) + TRAF1(NP1)
COMMON/LINCAF(NP1) + TRAF(NP1) + TRAF1(NP1)
COMMON/LINCAF(NP1) + TRAF(NP1) + TRAF1(NP1)

DIMENSION NETSUM(NP3, 2) + ORNET(MW, MW, NP3)
DIMENSION NLINK(NP3) + LNKCHW(NP3 + NP4) + LNKCLN(NP3, 2)
21 INTEGER SUMCST
22 INTEGER ORNET
23 DIMENSION TRM(MW, MW) + TR(MW, MW)
24 INTEGER ORICST + ORICST1 + ORICST2
25 INTEGER DIVTRJ(MW) + DIVTRI(MW)
26 INTEGER TRM(MW + MW) + NETCF(NM + MW, NP3) + LINEQ(NP3)
27 C + LINAD(NP3) + LINAD(NP3) + LINAD(NP3)
28 C + LINEQ(NP3) + LINEA(NP3) + LINEF(NP3)
29 DIMENSION RHOF2(MW, MW)
30 DIMENSION TRM(MW, MW)
31 EQUIVALENCE (LINEQ, LINEQ) + (LINEQ, LINEG)
32 C
33 C RESET UTILIZATION FACTOR TO .5
34 C
35 DO 70 NN1 = 1, N3
36 UTILIZ(NN1) = .5
37 70 CONTINUE
38 C
39 C COMPUTE ORNET(MW, MW, N3) FOR INITIAL TOPOLOGY WHERE N3 IS
40 C THE NUMBER OF CHARGEABLE ITEMS
41 C
42 ORICST = 0
43 ORICST1 = 0
44 ORICST2 = 0
45 NN1 = NN1 - 1
46 C DO 203 NN1 = 1, N3
47 DO 203 NN2 = 1, 2
48 NETSUM(NN1, NN2) = 0 GOST SUM
49 203 CONTINUE
50 C
51 C MODIFY DUALISING MODE FROM HALF TO FULL. DUPLX
52 C
53 DO 667 K1 = 1, N3
54 DUPLX(K1) = 2
55 667 CONTINUE
56 DO 101 I1 = 1, NN1

A-32
77-53, Vol. IV

57  NLINK(I)=NR1  ORIG LINKS AT THE BEGINNING
58   I=I+1
59  DO 102  J=I+1,NR
60   II=NRSC(I)
61   JJ=NRSC(J)
62   ATRMAX=AMAX1(TR(I,J), TR(J,I))  DASSUMING FULL DUPLEX
63  CALL LINNUM (ATRMAX,0,LINFO,LINUP,0,PHO)
64   RHOF2(I,J)=PHO
65   RHOF2(J,I)=PHO
66  CALL ICOSTJILINE(I,J, LNKCHW, LNK)()
67  DO 104  NN=1,N3
68   ORINET(I,J,NN)=LINFO(NN)
69   ORINET(J,I,NN)=LINFO(NN)
70  DO 105  NI=1,N2  @ LINE COST
71   NETSUM(NN+NM)=NETSUM(NN+NM)+LNK(NN,NN)
72  DO 106  NK=1,N4  @ HARDWARE COSTS
73   NETSUM(NN+NM)=NETSUM(NN+NM)+LNKCHW(NN,NK,NM)
74  106  CONTINUE
75  105  CONTINUE
76  104  CONTINUE
77  103  CONTINUE
78  102  CONTINUE
79  101  CONTINUE
80  100  CONTINUE
81   ORINET(K1,K1,NN)=0
82  107  CONTINUE
83  CALL OUTPUT(I)
84   ITALL=0
85  999  CONTINUE
86   MAXSAV=0
87  DO 777  I=1,NR1
88   IF (NLINK(I) .LE. LIMIT) GO TO 777
89   I=I+1
90  DO 788  J=I+1,NR
91   IF (NLINK(J) .LE. LIMIT) GO TO 788
92   IN=NTEST(ORINET+I,J)
93   IF (IN .EQ. 0) GO TO 788  AND LINK TO BE DELETED
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=I+1,NR1
101   LI=L+1
102  DO 138  M=I+1,NR
103   IYTEST(I, J, J, L, M)
104   IF (IY .EQ. 1) GO TO 110  OR NEXT STEP NOT TO BE TESTED
105   IN=NTEST(ORINET+L,M)
106   IF (IN .EQ. 1) GO TO 138
107  110  CONTINUE
108 DO 137  N=I+1,NR
109   IYTEST(I, J, J, L, N)
110   IF (IY .EQ. 1) GO TO 137
111   IN=NTEST(ORINET+L,N)
112   IF (IN .EQ. 0) GO TO 137
113  133  CONTINUE
A-33
114    rv=ITEST(I,J,N,N)
115    IF(rv .EQ. 1) GO TO 137
116    INTEST(INF,INFT,N,N)
117    IF (INF .EQ. 1) GO TO 138
118    137 CONTINUE
119    GO TO 788
120  138 CONTINUE
121  139 CONTINUE
122    SNVIT=0
123    SNVIT=0
124    CALL TRFDIV(IFLOP)
125    IF (IFLOP .EQ. 1) GO TO 201
126    CALL MINAD(I1AD+MINCST)
127    GO TO 202
128    201 CONTINUE
129    CALL NETWKC(MINCST)
130    202 CONTINUE
131    ISA=ORICST-MINCST
132    IF (MAXSAV .GE. ISA) GO TO 78A
133    MAXSAV=ISA
134    IFLIP=IFLOP G IFFLIP=GLOBAL INDICATOR
135    IF (IFLOP .EQ. 1) GO TO 204
136    JMAX=IAD
137    IMAX=I
138    JMAX=J
139    DO 666 NN=1,N3
140    ILNAD(NN)= LINADT(NN) CHANGE OF LINE Rgf.
141    JLINAD(NN)= LINADJ(NN)
142  666 CONTINUE
143   204 CONTINUE
144    DO 331 K1=1,NR
145    DO 331 K2=1,NR
146    TRRM(K1,K2)=TRR(K1+K2)
147   331 CONTINUE
148   788 CONTINUE
149   777 CONTINUE
150    IF (MAXSAV .LE. 0) GO TO 9999
151    CALL NETUP(IFLOP+JMAX+IMAX+JMAX)
152    ITALL=ITALL+1
153   9999 CONTINUE
154    109 FORMAT(1X:' THIS NETWORK HAS BEEN UPDATED FOR T TIMES')
155    WRITE(6,109) ITALL
156    DO 81 I=1,N3
157    DO 81 J=1,2
158    NETSUM(I,J)=0
159    81 CONTINUE
160    DO 91 I=1,NR1
161    NETSUM(I,J)=0
162    K=1+1
163    DO 92 K=1,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
SUBROUTINE OUTPRTSN

C **PRINT OUT INTERREGIONAL NETWORK CONFIGURATION AND ITS COSTS**

C

C

C

DO 110 I=1,N3
ORICS1=ORICS1+NETSUM(I+1)
ORICS2=ORICS2+NETSUM(I+2)
110 CONTINUE

NTURN=NR/10+1
DO 201 L=1,NTURN
LL=NTURN+10+1
LU=NTURN+10
IF(LU.GT.NR) LL=NR
IF(N.LT.2) GOTO 2100
WRITE(LWT,2002) (J,J=LL,LU)
2002 FORMAT(* //10X,**INITIAL INTERREGIONAL NETWORK CONFIGURATION**, + //20X,10(5X,I3,2X))
203 GOTO 2101
2100 CONTINUE
WRITE(LWT,2003) (J,J=LL,LU)
2003 FORMAT(* //10X,**FINAL OPTIMAL INTERREGIONAL NETWORK CONFIGURATION**, + //20X,10(5X,I3,2X))
206 CONTINUE
DO 2108 M=1,N3
WRITE(LWT,2004) (LINAME(M,J),J=1,LU)
2004 FORMAT(15X,J14.14)
DO 210A N=1,N3
WRITE(LWT,2008) LINAME(M),ORINFT(M,K),M=1,N3
2008 FORMAT(* //15X,**LINE UTILIZATION**,I10,L8.8,I16.16)
210A CONTINUE
2108 CONTINUE
WRITE(LWT,2201) (RHOP2(I,J),J=1,LU)
2101 FORMAT(* //20X,**MAXIMUM INTERREGIONAL COSTS**,I4,3X,1F4.3,2X)
2103 CONTINUE
WRITE(LWT,2206) (J,J=1,LU)
2201 FORMAT(* //17X,**INST. COST**,7X,**RECV. COST**,9X,**SHTOTAL**)
DO 2205 K=1,N3
ISUM=NETSUM(K)+NETSIM(K,N)
WRITE(LWT,2207) K,LINAME(K),ISUM
DO 2207 I=1,LU
WRITE(LWT,2208) LINAME(I),ISUM
2207 FORMAT(* //15,3X,A6,5X,I6,11X,I6,10Y,1R)
220A CONTINUE
2209 CONTINUE
WRITE(LWT,2206)

ORICST=ORICS1+ORICS2
WRITE(IWT,200) ORICS1,ORICS2,OPICST
200 FORMAT((//9X,TOTAL,17X,10X,17X,10X,1A))
RETURN
FUNCTION ITEST(I,J,K,L)

ITYEST=0
IF (I.EQ. K .AND. J.EQ. L) ITEST=1
IF (I.EQ. L .AND. J.EQ. K) ITEST=1
RETURN
FUNCTION NTEST(NET,I,J)

RETURN

C ITEST DIRECT LINE CONNECTIVITY BETWEEN I & J.
C
C
C DIMENSION NET(MW+MW,NP3)
DO 103 II=1,NB
IF (NET(I,J,II) .GT. 0) GO TO 10A
103 CONTINUE
NET=0 . NO CONNECTION
RETURN
10A NET=1
RETURN . YES, THERE IS A CONNECTION
SHROUTINE TRFDIV(IFLOP)

C

C DIVERT TRAFFIC BETWEEN I AND J THROUGH OTHER MSCS.
C
C
C
C IT RETURNS WITH IFLOP=1 WHEN SUCCESSFUL OR IFLOP=0.
C IT ALSO CREATES TEMPORARY MATRICES TRA AND NETCF.
C
C SNIVT=0, QTOTAL TRAFFIC DIVERTED (I TO J)
SNIVT=0, QTOTAL TRAFFIC DIVERTED (J TO I)

DO 205 K=1,NR
DIVRI(K)=0. Q TRAFFIC DIVERTED THRU REGION K (I TO J)
DIVRJ(K)=0. Q TRAFFIC DIVERTED THRU REGION K (J TO I)
CONTINUE
DO 220 II=1,NR
IF (II.EQ.I) OR (II.EQ.J) GO TO 220
IC1=ITEST(ORINET(I),II)
IC2=ITEST(ORINET(J),II)
IF (IC1.EQ.0 .OR. IC2.EQ.0) GOTO 220
C
C DIVERT I TO J TRAFFIC THRU II
C
C DIVRI(I)=,.
DIVRI(J)=,.

CALL LINTRF(I,J,A)
DELTA=A-TPR(I,J)
IF (DELTA.EQ.0.0) GO TO 160
C
C
DIVTR=DELT
CALL LINTRF(II,J+1)
IF (DELT.EQ.0.0) GO TO 160
DIVTR(II)= AMIN(DIVTR,DIVTR)
IF ((DIVTR(II)* SDIVTI),GT,TR(1,I,J))GO TO 140
SDIVTI= SDIVTI+ DIVTR(II)
GO TO 160
C
IF (FLTR.LE.0.0) GO TO 180
DIVTR(II)= TR(I,J)+ SDIVTI
SDIVTI= TR(I,J)
C
DIVERT J TO I TRAFFIC THRU II
C
CONTINUE
CALL LINTRF(J+II,J+II+1)
DELTAE A-TR(J,J+II)
IF (DELT.EQ.0.0) GO TO 220
DIVTR(J+II)= AMIN(DIVTR,DIVTR)
IF ((DIVTR(J+II)* SDIVTJ),GT,TR(J,J+II))GO TO 140
SDIVTJ= SDIVTJ+ DIVTR(J+II)
GO TO 220
C
DIVTR(J+II)= TR(J+II,J)+ SDIVTJ
SDIVTJ= TR(J+II,J)
C
CONTINUE
CALL LINTRF(J+II,J+II+1)
DELTAE A-TR(J+II,J+II)
IF (DELT.EQ.0.0) GO TO 220
DIVTR(J+II)= AMIN(DIVTR,DIVTR)
IF ((DIVTR(J+II)* SDIVTJ),GT,TR(J+II,J))GO TO 140
SDIVTJ= SDIVTJ+ DIVTR(J+II)
GO TO 220
C
DIVTR(J+II)= TR(J+II,J)+ SDIVTJ
SDIVTJ= TR(J+II,J)
C
CONTINUE
IF (FLTR.LE.0.0) GO TO 340
CONTINUE
IFLOPS=0
GO TO 360
IFLOPS1
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
TIR(K1,K2)=TR(K1,K2)
DO 191 K3=1+NR
NETCNF(K1,K2,K3)=ORINF(K1,K2,K3)
191 CONTINUE
DO 190 K1=1+NR
NETCNF (I,J+K1)=0
NETCNF (J+I,K1)=0
NETCNF (I+I,K1)=0
NETCNF (J+J,K1)=0
190 CONTINUE
TH(I,J)=0
TH(J,I)=0
DO 380 IK=1rNR
IF (I.EQ. IK .OR. J.EQ. IK) GO TO 380
TRR(I+IK)= TR(I+IK) + DIVTR(IK)
TRR(J+IK)= TR(J+IK) + DIVTR(JK)
TRR(I+IK)= TR(I+IK) + DIVTR(JK)
ATRMAX=ATRMAX + TRR(I+IK)+TRR(IK+J)
BTRMAX=BTRMAX + TRR(J+IK)+TRR(JK+I)
350 INRNSC(I)
351 INR2=NRNSC(I)
352 CALL LINNUM1(ATRMAX,0,LINEQ,'LINUP,0,RHO)
353 RHOF2(I+IK)=RHO
354 RHOF2(K+J)=RHO
355 INR1=NRNSC(J)
356 CALL LINNUM1(BTRMAX,0,LINEQ,'LINUP,0,PFO)
357 RHOF2(J+IK)=RHO
358 RHOF2(K+J)=RHO
359 DO 430 NR1=1,N3
430 CONTINUE
380 CONTINUE
380 CONTINUE
RETURN
367 SUBROUTINE LINTRF(I,J,A)
369 C ***************
370 C CONVERT LINES INTO TRAFFIC CAPACITIES BETWEEN NODES I AND J.
371 C ***************
373 C
374 DO 100 IR=I+1,N3
375 A=AS+ ORINET(I+J,IR)*LINCAP(IR)*UTILI(IR)
376 CONTINUE
377 RETURN
378 SUBROUTINE NETWCF(SUMCST)
379 C ***************
380 C FIND TOTAL INTERREGIONAL NETWORK COST,SUMCST,BASED ON SPECIFIC
381 C CONFIGURATION NETCNF
382 C ***************
383 INTEGER SUMCST
384 SUMCST=0
385 DO 420 I=1,NR1
386 IR=I+1
387 DO 400 IK=IR+1
400 IC=ICFST(NETCNF,IP,IK)
388 IF (IC.EQ.0) GO TO 400
389 IC=INRSC(IR)
390 J=INRSC(K)
391 JJ=INRSC(IK)
392 DO 150 I=1,N3
393 LINEQ(I+III)= NETCNF(IR+IK+J)
394 CONTINUE
395 CALL TCOSTJ(LINEQ+IIJ,J,JNCHW,JNCLN)
396 DO 501 J=1,N3
397 DO 510 J=1,N3
398 DO 520 J=1,N4
A-38
77-53, Vol. IV

399 SUMCST=SUMCST+LNKCHW(J1,J3,J2)
400 520 CONTINUE
401 SUMCST=SUMCST+LNKCHW(I1,I2)
402 510 CONTINUE
403 SUMCST=SUMCST+LNKCHW(I1,J2)
404 501 CONTINUE
405 SUMCST=SUMCST+LNKCHW(I1,I2)
406 RETURN
407 SIROUTINF MINAD(IAD,MINCST)
408 C
409 ***************
410 C
411 C CAPACITY INCREASE IS REQUIRED WHEN TFLOP=0, AND THE CAPACITY AT
412 C MAXIMUM COST SAVINGS.
413 C
414 ***************
415 DIMENSION LINDJ(NP3),LINDJ(NP3)
416 MINCST=0
417 RTFJ=TR(J,J)+SUMT(J) REMAINING TRAFFIC FROM J TO I
418 RTFJ=TR(J,I)+SUMT(I) REMAINING TRAFFIC FROM I TO J
419 DO 500 II=1,NR
420 IF(IJ.EQ.I0.OR.IJ.EQ.J1) GO TO 500
421 IF(ITRRI(I,J),GE.0. OR. TRR(I,J),GE.0.1) GO TO 500
422 C
423 C DETERMINE DELTA COST FOR INCREASE CAPACITY IN ALTERNATE ROUTE.
424 C
425 C
426 AII=TRR(I,I)+RTFJ
427 AII=TRR(I,J)+RTFJ
428 AMX=AMAX1(AII,AII)
429 IDR1=NRCN(I)
430 CALL LINNUM(A,0,LINEAR,LINNP,0,EHO)
431 DO 511 NN=1,NR
432 LINDJ(NN)=LINEAR(NN)-NETCF(I,J,NN)
433 NETCF(I,J,NN)=LINEAR(NN)
434 NETCF(I,J,NN)=LINEAR(NN)
435 511 CONTINUE
436 C
437 C
438 C
439 C
440 BJI=TRR(J,I)+RTFJ
441 AMX=AMAX1(BJI,BJI)
442 IDR2=NRCN(J)
443 CALL LINNUM(BM0,0,LIMGB,LINJP,0,EHO)
444 DO 511 NN=1,NR
445 LINDJ(NN)=LIMGB(NN)-NETCF(J,I,NN)
446 NETCF(J,I,NN)=LIMGB(NN)
447 NETCF(J,I,NN)=LIMGB(NN)
448 511 CONTINUE
449 CALL NETWKC(SUMCST)
450 C
451 IF(SUMCST,GT,MINCST) GO TO 120
452 DO 207 NN=1,NR
453 LINAD(NN)=LINDJ(NN)
454 LINAD(J1)=LINDJ(NN)
455 207 CONTINUE
456 11AD="I"
MINCST = SUMCST

CONTINUE

C
C RESET TO INITIAL NETWORK CONFIGURATION FOR NEXT TRY

DO 250 NN = 1, N3

NETCNF(I+I+NN) = NETCNF(I+I+NN) + LIMDN(1)

NETCNF(J+J+NN) = NETCNF(J+J+NN) + LIMDN(1)

NETCNF(J+J+NN) = NETCNF(J+J+NN) + LIMDN(1)

NETCNF(I+I+NN) = NETCNF(I+I+NN) + LIMDN(1)

250 CONTINUE

TRP(I:IIAD) = TRP(I:IIAD) + RTPFI

TRP(IIAD:J)TRP(IIAD:J) + RTPFI

TRP(J:IIAD) = TRP(J:IIAD) + RTPFJ

TRP(IIAD:J) = TRP(IIAD:J) + RTPFJ

RETURN

C SUBROUTINE NETUP(IFLIP, IIAAD, I, J)

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

C C UPDATE THE INTRREGIONAL NETWORK WHEN THERE IS SOME SAVINGS

C

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

IF (IFLIP .EQ. 1) GO TO 700

C C UPDATE THE NETWORK TRAFFIC MATRIX AND

C C UPDATE THE OPTIMAL INTRREGIONAL NETWORK

DO 99 NN = 1, N3

ORINET(I:IIAD+NN) = ORINET(I:IIAD+NN) + LIMDN(1)

ORINET(IIAD:J+NN) = ORINET(IIAD:J+NN) + LIMDN(1)

ORINET(J:IIAD+NN) = ORINET(J:IIAD+NN) + LIMDN(1)

ORINET(I:IIAD+NN) = ORINET(I:IIAD+NN) + JLIMDN(1)

99 CONTINUE

700 CONTINUE

DO 701 NN = 1, N3

ORINET(I+J+NN) = 0

ORINET(J+I+NN) = 0

701 CONTINUE

C C RESET TRAFFIC MATRIX TR(INR:NR)

C

DO 980 IR = 1, NR

DO 990 IK = 1, NR

TR(IR:IK) = TRM(IR:IK)

980 CONTINUE

990 CONTINUE

C C UPDATE TOTAL COST FOR OVERALL NETWORK

C

ORICST = ORICST - MAXSAV

C C UPDATE NLINK MATRIX

C

NLINK(I) = NLINK(I) - 1

NLINK(J) = NLINK(J) - 1

RETURN

END

APRT STA.COM, I=CO$TJ/0777

A-40
SUBROUTINE ICOSTJ(ILINEU,I,J,LNKCHM,LNKCLM)

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

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

PARAMETER NP1=130,NP2=1,NP3=4,NP4=3,NP5=360
PARAMETER NP6=2
DIMENSION LINSTAT(NP3), LNKCHM(NP3,NP4+2), LNKCLM(NP3,NP2)
COMMON/LINCHM,LINSTAT(NP3),LNKCHM(NP3,NP4+2),LNKCLM(NP3,NP2)
/CONST/N1,N2,N3,N4,N7,NCYT
/RCOST/AINSTC(NP2,NP3,NP4+2),RFPC(NP2,NP3,NP4+2)
/AMSTLN(NP2,NP3,NP4+2), RECLN(NP2,NP3,NP4+2)
/STRUC/N1,N2,N3,N4,N7,NCYT
ADV, IAND, IAND(NP1),KCHG,KAND, GTERMNTLS WITH SAME V-H
/REF/RFENDIF(NPC),TRAFD(NP1,2,NP7),DSTMCE(NP6),MAPADR(NP1)

INITIALIZATION
II=MAPADR(I)
IJ=MAPADR(J)
IAND=IAND(IJ)
DO 100 NL=1+NL
100 CONTINUE

KRATE= IRAND(I)+1 N RATE STRUCTURE TYPE FOR NODE I
KRF= IRAND(I)+1 N RATE STRUCTURE TYPE FOR NODE J
KDEN= IRAND(I)+2 N TRAFFIC DENSITY TYPE FOR NODE I
KDSN= IRAND(I)+2 N TRAFFIC DENSITY TYPE FOR NODE J
KDEC= KDENSTY+1 N DENSITY TYPE, 2=N-L AND 1=L-L
K=KDENS(J) 03=H-H, 2=H-L AND 1=L-L
DST = DIST(I,J) N DISTANCE BETWEEN NODES I AND J

ENDIST

ITIP2 = GPRIME COST FOR H/W UNIT

IF DST LE 0.5) ITIP3 = GRESS COST FOR ADDITIONAL UNIT
K P= IARTF(KRATE,KRATE,J) N ACTUAL RATE STRUCTURE TO BE USED
DO 1 IL= 1+NL

INPX = IDUPLX(IL) N DUPLEx ING NODE 1H AND 2F

MNV = LINES(IL) N NUMBER OF LINES REQUIRED

MV1=MNV*IANDIN*KAND

CALCULATE COSTS FOR NON-LINE TYPE CHARGES
C
C IF (MNV=0) GO TO 1 N NO LINES ARE REQUIRED
DO 2 IV=1+NL N HIGH DENSITY RATE

C INSTALLATION COSTS FOR NON-LINE TYPE CHARGES
LNKCHM (IL,IV+1) = AINSTC(KR,IL,IV,KK,INPX,ITIP3)*KCHG*MNV

A-41
145      1  * ANNUAL RECURRING COSTS FOR NON-LINE TYPE CHARGES
146      2  LNKCML(IL,IV) = (RECRC(KR,IL,IV,KK,TDPX,ITIP)*KCHG*NDV' + RECRC(KR,IL,IV,KK,TDPX,2)*NDV)*12.
147      2  CONTINUE
148      2  CALCULATE LINE COSTS
149      2  LIN= IFLAG(KR,IL)  @ LINEAR IF 1 AND NONLINEAR OTHERWISE
150      2  ANNUAL LINE INSTALLATION COST
151      2  AN=1.
152      2  LNKCML(IL+1) = ANSTLN(KR,IL,KK,TDPX,2)*AN*NDV.
153      2  IF (LIN.NE.1) GO TO 41
154      2  LINEAR LINE RECURRING COST FUNCTION
155      5  BN=NST/RECRLN(KR,IL,KK,TDPX,1)  
156      5  LNKCML(IL+2) = RECRLN(KR,IL,KK,TDPX,2)*AN*NDV*12.
157      5  GO TO 32
158      1  CONTINUE
159      1  NONLINEAR LINE RECURRING FUNCTION
160      3  DO 10 NON1=1,8
161      3  NON2=2*NON
162      3  NON1=NON2-1
163      3  COST=RECRLN(KR,IL,KK,TDPX,NON2)
164      3  DT=RECRLN(KP,IL,KK,TDPX,NON1)
165      3  IF (DST.GT.DT) GO TO 51
166      3  LNKCML(IL+2) = COST*DST*NDV*12+LNKCML(IL,2)
167      5  CONTINUE
168      5  LNKCML(IL,2) = COST*DST*NDV*12+LNKCML(IL,2)
169      3  DST=DST-DT
170      1  CONTINUE
171      1  CONTINUE
172      1  CONTINUE
173      1  KCHG=1
174      1  KAPD=1
175      100  RETURN
176      101  END

CPRNT STACOM.RHOFUN/0777
**SUBROUTINE RHOFUN(T1,T2,LINEQU,LNLMT,RLHOLIN,RHO)**

**CALCULATE LINE UTILIZATION**

T1 = LINT TO SWITCHER TRAFFIC
T2 = SWITCHER TO LINT TRAFFIC
LNLMT = HIGHEST LINT TYPE
LINEQU = LINE CONFIGURATION

**PARAMETER NP3=4**

COMMON/LINCHR/LINIX(NP3),LINCAP(NP3),UTILIZ(NP3)

*/CONST/ N1+N2+N3+NH+N7=NMCITY
*/SUM/ASUM(4)+RSUM
*/XMT/ TIMXMT(7)+NP3+WAIT(6)
*/MSLA/ AMSL(7)
DIMENSION LINEQU(1),RLHOLIN(1)
RHO=0
CAP=0
DO 8 N=1,N3
CAP=CAP+LINEQU(N)*LINCAP(N)
8 CONTINUE
CAP=CAP/LINCAP(LNLMT)/CAP NORMALIZATION FACTOR
X SAC1=CAP+ASUM(4)/(ASUM+AMSL(5)+R)* OUTPUT WITH PRI TO 1
X SAC2=0
IF(AMSL(6) .GT. 0) GOTO 1201
X SAC3=CAP+ASUM(4)/(ASUM+AMSL(6)+R) OUTPUT WITH PRI TO 2
1201 CONTINUE
X SAC1=CAP+T1/AMSL(4)+R INPUT TRAFFIC IN TRANS
RLHOLIN(1)=X SAC1+TIMXMT(5,LNLMT)
RLHOLIN(2)=X SAC2+TIMXMT(6,LNLMT)
RLHOLIN(3)=X SAC3+TIMXMT(4,LNLMT)
RHO=RLHOLIN(1)+RLHOLIN(2)+RLHOLIN(3)
RETURN
END
SUBROUTINE LINNUM(T1,T2,LINEQU,LNLMT,JFLAG,RHO)

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

C FIND LINE CONFIGURATION BASED ON THE GIVEN TRAFFIC AND
C APPLICABLE LNF TYPE
C
C JFLAG= 1 FOR MULTIDROP LINE CASE
C
C T1= LINE TO SWITCHER TRAFFIC
C
C T2= SWITCHER TO LINE TRAFFIC
C
C
***************

PARAMETER NP3=4
COMMON/LINCHR/ LINMIX(NP3)*LINCAP(NP3)*UTILIZ(NP3)
* /CONST/ N1*N2*N3*N4*N7*N8*N9+N10*/

INTEGER TRAF
DIMENSION LINEQU(1,RHOLIN(N3))

TRAF=T1+T2
DO 1 I=1,N3
LINEQU(I)=0
1 CONTINUE
LNLMT=0
CALL REFER
CALL LINNUM(LNLMT)

C SET UP INITIAL LINE CONFIGURATION
C
IF(JFLAG .EQ. 1) GOTO 10
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(NLMT)+1)
7 CONTINUE
70 CONTINUE
CALL RHOFUN(T1,T2,LINEQU,LNLMT,RHOLIN,RHO)
IF(RHO .LT. UTILIZ(LNMT)) GOTO 150
IF(LNLMTU.NE.N3) GOTO 72
IF(JFLAG.NE.1) GOTO 75
LINEQU(N3)=LINEQU(N3)+1 0 NEEDED TO BE MODIFIED
GOTO 70
73 CONTINUE
DO 2 N=1,N3
IF(LINEQU(N) .NE. 0) GOTO 20
CONTINUE
20 CONTINUE
NL=NL
IF(NL.EQ.N3) GOTO 74
LINEQU(NL)=0
22 CONTINUE
NL=NL+1
IF(LINMIX(NL),EQ.0) GOTO 22
LINEQU(NL)=LINEQU(NL)+1

A-44
57  74  GOTO 70
58  74  CONTINUE
59  LINEGU(I)=1
60  GOTO 70
61  72  CONTINUE
62  LINEGU(LNLMT(I))=0
63  LNLMT(I)=LNLMT(I)+1
64  IF(LINMIX(LNLMT(I))=F.G.0) GOTO 79
65  LINEGU(LNLMT(I))=1
66  GOTO 70
67  150  CONTINUE
68  LNLMT=LNLM'TU
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)=F.G.0) GOTO 14
77  LNLMT=NN
78  IF(LINCAP(NN)=G.T. LTRAF) GOTO 15
79  14  CONTINUE
80  15  CONTINUE
81  RETURN
82  END

OPRT STACON.PACK/0777
5192B*STACOM(1),PACK/0777

1    COMPILER (FLD=ABS)
2    SUBROUTINE PACK(I,K,L,IA)
3    **************
4    C
5    C RETRIEVE/STORE DATA FROM/INTO ARRAY IA
6    C L=1 FOR STORING AND L=2 FOR RETRIEVAL
7    C K= DISTANCE DATA CONCERNED
8    C
9    C  **************
10    DIMENSION IA(1)
11    I0=(L-1)/4  THE WORD LOCATION
12    IR=L-I0+4  THE QUARTER CONCERNED
13    IQ=I0+1
14    IS=(IR-1)*4
15    IF(L<=15) GOTO 10
16    C
17    C RETRIEVE IT(9 BITS) BEGINNING AT IS-TH BIT OF THE IR-TH WORD
18    C
19    K=FLD(15,I0,IA(I0))
20    RETURN
21    C
22    C STORE IT(9 BITS) BEGINNING AT IS-TH BIT OF THE IR-TH WORD
23    C
24    CONTINUE
25    FLD(IS,9,IA(I0))=K
26    RETURN
27    END

@RTR STACOM,DIST/0777
FUNCTION DIST(I,J)
***************
C FIND DISTANCE BETWEEN I AND J
C ***************
PARAMETER NPI=130+NPC=360
PARAMETER NP6=(NPC*NPC/2+NPI+1)/4+1
PARAMETER NP7=4
COMMON /REF/REF(NPC), TRAFF(NP1+2,NP7),
* DSTNCE(NP6), MAPADR(NPI)
INTEGER DISTANCE
DIST=0.
IF(I.EQ.J) RETURN
II=MAPADR(I) ACTUAL CITY INDEX
JJ=MAPADR(J)
IF(I.EQ.JJ) RETURN
I,J,L=LINK(I,J)
CALL PACK(I,J,L,DIST,2,DISTCF)
DIST=DIST
IF(DIST .NE. 511) RETURN
DIST=DFRMR(I,L)
RETURN
END

PRINT L, LINK/0777
FUNCTION LINK(J,K)

C FIND THE RELATIVE LOCATION FOR (J,K) COMBINATION
C WHICH IS THEN USED FOR FINDING DISTANCE BETWEEN SYSTEM
C TERMINATIONS J AND K

PARAMETER NP1=130,NPC=360
PARAMETER NP6=(NPC+NPC/2-NPC+1)/4+1
PARAMETER NP7=4,NP8=4
COMMON /CONST/ N1,N2,N3,N4,N7,NCITY
1 /REF/IREF(NPC)+TRAFF(NP1+2,NP7),DISTCF(NP6),MAPADR(NP1)
INTEGER DISTNCE
LINK=0
IF(J.GT.NCITY.OR.K.GT.NCITY.OR.K.EQ.J) RETURN
JJ=J
KK=K
IF(J.GT.K) GOTO 1
JJ=K
KK=J
1 CONTINUE
LINK=(JJ-1)*NCITY + KK - IREF(JJ)
RETURN
END
FUNCTION RECOVR(I)

C RETRIEVE OVERFLOW DISTANCE DATA FROM IVRD

C

PARAMETER NPC=360,NPD=10,NPC
COMMON /OVER/ IVRD(NPD+2),IOVER1
DO 10 N=1,IOVER1
10 IF(I.EQ. IVRD(N+1)) GOTO 20
11 CONTINUE
12 WRITE(6,99) I
13 99 FORMAT(1X,\'NO OVERFLOW DATA HAS BEEN FOUND \',
14 * FOR LOCAL INDEX \',I2\')
15 STOP
16 CONTINUE
17 RECOVR=IVRD(N+2)
18 RETURN
19 END

51928*STACOM(1),RECOVR/0777

A-49
SUBROUTINE PLOTPT(L1,L3)

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

SUBROUTINE FOR MOVING CALCOMP PEN WITH OR WITHOUT PEN DOWN

L3=1 FOR MOVING WITH PEN DOWN

=2 FOR MOVING WITH PEN UP

=3 FOR CLOSING THE PLOTTER

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

PARAMETER NPC=360

COMMON/VH/VERT(NPC),HORIZONTAL(NPC)

DIMENSION IBUF(1000)

DATA IP/0/, GFLAG FOR PLOTS CALL

DATA X/1.2566/

IF(IP .NE. 0) GOTO 50

CALL PLOTS

IF(L3 .EQ. 3) GOTO 100  GPLOTTING IS TO BE CLOSED

AV=VERT(L1)

AH=HORIZONTAL(L1)

BV=AV*COS(X)+AH*SIN(X)

BH=AV*SIN(X)-AH*COS(X)

BH=(BH+RH)/301.

IF(L3 .EQ. 2) GOTO 80

CALL SYMBOL(BH,BV,0.0254,0.0,2) PEN IS DOWN

IF(IP .NE. 2) GOTO 80

CALL PLOT(RH,BV,2) PEN IS UP

IP=1 GPLOTS CALL IS NOT NEEDED ANY MORE

RETURN

100 CONTINUE

CALL PLOT(10.,0.,999)

RETURN

END

SUPPORT RESPONSE/0777
SUBROUTINE RSPNSE(T1, T2, LINTYP, N, ICK)

C CALCULATE MEAN RESPONSE TIME FOR THE PROPOSED MULTIDROP LINE

C WAIT(6)=TIMED DELAYS DUE TO

C 1=WAIT FOR POLLING   2=WATT FOR I/O   3=INPUT XMT TIME

C 4=CPU TURNAROUND   5=OUTPUT QUEUE   6=OUTPUT XMT TIME

C

PARAMETER NMP=4
COMMON/RESP, RHO,LINTYP, RSPTIME
1 /XMT/ TIMXMT(NMP)+ WAIT(6)
2 /BOUND/ NTERMS,TIMPRD,PROC,PLOT
3 /CONST/N1,N2,N3,N4,N7,N9,N10
DIMENSION LDUMMY(NMP)
10 DO 10 N=1+N3
LDUMMY(N)=0
10 CONTINUE
LDUMMY(LINTYP)=1
10 CONTINUE
10 IOK=0
20 CALL RHOFUN(T1, T2, LDUMMY, LINTYP, RHO, LINTYP)
30 RHO=LINTYP=1, RHO
40 IF(RHO LINTYP<0.5) RETURN
50 WAIT(1)=0.5+0.5/(1-RHOLINTYP/(1-RHOLINTYP))
60 WAIT(2)=0.5+0.5/(1-RHOLINTYP/(1-RHOLINTYP))
70 RSPTIME=0.
80 DO 10 J=1,6
90 RSPTIM=WAT(J)+RSPTIME
100 CONTINUE
11 IF RSPTIME GT TIMREQ RETURN
12 IOK=1
13 RETURN
15 END

QPRR L INPUT/0777
## 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>Definition</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>
</tbody>
</table>

B-2
UNIVAC

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

Vertical Horizontal (V-H) Coordinates

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