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:

Executive Summary

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

Requirements Analysis and Design of Texas Criminal Justice Telecommunications Network

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

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

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

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

It presents the results of one phase of research carried out jointly by the Jet Propulsion Laboratory, California Institute of Technology, and the States of Texas and Ohio. The project is sponsored by the Law Enforcement Assistance Administration, Department of Justice, through the National Aeronautics and Space Administration (Contract NAS7-100).
A users' guide is provided in this volume for the network design software developed during the State Criminal Justice Telecommunications (STACOM) project sponsored by the Law Enforcement Assistance Administration (LEAA).

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

1 INTRODUCTION .............................................. 1-1
1.1 PURPOSE AND SCOPE ........................................ 1-1
1.2 SUMMARY .................................................. 1-1
1.2.1 The STACOM Program .................................... 1-1
1.2.2 State Criminal Justice Communication Network and its Optimization ........................................ 1-2
1.2.3 Functions Performed by the STACOM Program ............. 1-2
1.2.4 Operational Procedure .................................... 1-4
1.2.5 Aborting and Recovering a Run ............................ 1-5
2 THE STACOM PROGRAM ......................................... 2-1
2.1 INTRODUCTION .............................................. 2-1
2.1.1 State Criminal Justice Information System ................ 2-1
2.1.2 State Digital Communication Network ...................... 2-1
2.1.3 A STACOM Communication Network ......................... 2-2
2.1.4 Communication Network Configurations .................... 2-3
2.1.5 Network Optimization ..................................... 2-3
2.1.6 The STACOM Program and its Purposes .................... 2-4
2.1.7 Functions Performed by the STACOM Program ............. 2-4
2.2 MAIN FEATURES ............................................. 2-5
2.2.1 Structure ............................................... 2-7
2.2.2 Response Time Algorithm ................................ 2-12
2.2.3 Flexibility ............................................. 2-13
2.2.4 Programming Language .................................... 2-14
2.2.5 Operating System Requirements ........................... 2-14
2.2.6 Functional Limitations ................................... 2-15
2.3 INPUT ..................................................... 2-15
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1</td>
<td>Data Requirements</td>
<td>2-15</td>
</tr>
<tr>
<td>2.4</td>
<td>PROCESSING LOGIC</td>
<td>2-18</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Traffic Calculation</td>
<td>2-18</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Distance Calculation</td>
<td>2-19</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Formation of Regions</td>
<td>2-21</td>
</tr>
<tr>
<td>2.4.4</td>
<td>Selection of Regional Switchers</td>
<td>2-24</td>
</tr>
<tr>
<td>2.4.5</td>
<td>Formation of Regional Star Networks</td>
<td>2-27</td>
</tr>
<tr>
<td>2.4.6</td>
<td>Optimization of Regional Networks</td>
<td>2-29</td>
</tr>
<tr>
<td>2.4.7</td>
<td>Formation of an Interregional Network</td>
<td>2-37</td>
</tr>
<tr>
<td>2.4.8</td>
<td>Optimization of an Interregional Network</td>
<td>2-37</td>
</tr>
<tr>
<td>2.5</td>
<td>OUTPUT</td>
<td>2-41</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Printer</td>
<td>2-41</td>
</tr>
<tr>
<td>2.5.2</td>
<td>CalComp Plot</td>
<td>2-43</td>
</tr>
<tr>
<td>2.6</td>
<td>SYSTEM CONFIGURATION</td>
<td>2-44</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Hardware</td>
<td>2-44</td>
</tr>
<tr>
<td>3</td>
<td>PROGRAM OPERATIONS</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1</td>
<td>INTRODUCTION</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>ENVIRONMENT</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Hardware</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Software</td>
<td>3-2</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Functional Limitations</td>
<td>3-2</td>
</tr>
<tr>
<td>3.3</td>
<td>RUN DESCRIPTION</td>
<td>3-3</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Initialization and Setup</td>
<td>3-3</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Run Options</td>
<td>3-4</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Control Instruction and Sequences</td>
<td>3-9</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Program Listing</td>
<td>3-11</td>
</tr>
<tr>
<td>3.4</td>
<td>SAMPLE RUN</td>
<td>3-11</td>
</tr>
</tbody>
</table>
### 77-53, Vol. IV

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4.1</td>
<td>Run Stream</td>
<td>3-11</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Input</td>
<td>3-12</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Output</td>
<td>3-24</td>
</tr>
</tbody>
</table>

**REFERENCES**

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1</td>
</tr>
</tbody>
</table>

**APPENDIXES**

<table>
<thead>
<tr>
<th>Letter</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>STACOM PROGRAM LISTING</td>
<td>A-1</td>
</tr>
<tr>
<td>B</td>
<td>GLOSSARY</td>
<td>B-1</td>
</tr>
</tbody>
</table>

**Figures**

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

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

Tables

2-1 Examples of Line Configurations obtained by
    Subroutine LINNUM ---------------------------------------- 2-29
3-1 Formats for Input Data ------------------------------------ 3-5
3-2 Input Data for the Example Run --------------------------- 3-13
3-3 Printer Output from the Example Run ---------------------- 3-25
3-4 Unit 6 Printer Output from the Example Run --------------- 3-33
SECTION 1
INTRODUCTION

1.1 PURPOSE AND SCOPE

The STACOM (STAtes 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-6 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 (STAtes 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 @@@ after interrupting the line communication by pressing the BREAK key, will terminate a program execution at any time. On the other hand, the EXEC-9 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.
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:

\[ A \]

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

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

- \( c(f_1) \) = 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 \( I_A \), \( I_B \), \( I_C \), \( I_D \), and \( I_E \) which are internal cardinal numbers for system terminations A, B, C, D, and E.

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

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

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

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

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

(1) MAIN Routine

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

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

(2) RGNNET Routine

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

(3) IRNOP Routine

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

(4) LINNUM Routine

This routine provides an estimated line configuration required to satisfy a given traffic load and is mainly called upon by routine RNNET. 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 termina:s, drops, modem and others can be used. Furthermore, under the Multischedule Private Line (MPL) tariffs given by AT&T for interstate communication lines, the monthly line charge between any two terminals is now a function of both the inter-city distance and the traffic densities of both terminal cities. The STACOM program has been implemented in such a way that it can take line-cost figures based on MPL tariffs or other tariffs.

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

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

(3) **Number of Terminals per Multidrop Line.**

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

(4) **Average Terminal Response Time.**

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

2.2.4 Programming Language

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

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

2.2.5 Operating System Requirements

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

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

As an example, the following is a complete list of EXEC-8 control statements which need to be prepared or typed in after the run card for properly executing the STACOM program.
2.2.6 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 \( N_1 \). In anticipation of possible multiple data bases at different locations, the number of data bases (designated internally as \( N_7 \)) informs the program that each system termination has \( N_7 \) 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. \( N_7 \) 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 \( N_7 \) data bases are needed in order to properly execute the STACOM program.

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

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

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

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

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

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

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

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

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

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

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

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

2.4 **PROCESSING LOGIC**

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

2.4.1 **Traffic Calculation**

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

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

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

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

and

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

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

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

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

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

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

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

\[
\text{Distance} = \sqrt{\frac{(569)^2 + (38)^2}{10}} = \sqrt{\frac{323761 + 1444}{10}} = \sqrt{32521} = 181 \text{ miles}
\]
When a specific location in the United States is not designated with specific V and H coordinates, it is normally assigned with the same V and H coordinates as the closest location.

Following the procedures as given above, the distance between any given two system terminations is calculated and stored in arrays 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:
   \[ L = I^{*}\text{NPC} + J - \Delta(I) \]
   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
   \[ L1 = \left\lfloor \frac{(L-1)}{4} \right\rfloor + 1 \]
   \[ S1 = (L-1) \text{ Modulo } 4 + 1 \]
where \( \lfloor x \rfloor \) = the integer part of \( x \) and
\[
D1 = \begin{cases} 
D & \text{if } D < 511 \\
511 & \text{if } D \geq 511 
\end{cases}
\]

(5) Store \( D1 \) in segment \( S1 \) of entry \( L1 \) of table DSTNCE.
(6) If \( D \geq 511 \), store \( L \) and \( D \) in next available space of table IVRD.

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

(1) Calculate \( L \), \( L1 \) and \( S1 \) as described above.
(2) Retrieve the content \( D1 \) in segment \( S1 \) of entry \( L1 \) of table DSTNCE. If \( D1 < 511 \), it is the distance.
(3) If \( D1 = 511 \), retrieve the second element of the row of table IVRD, whose first element contains value \( L \). The retrieval value is the distance.

2.4.3 Formation of Regions

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

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

\[
TPR = \frac{TPR1}{ANR1}
\]

with

\[
TPR1 = \sum_{1 \leq i \leq N1} [TRAFIT(i) + TRAFDN(i)]
\]

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

\[
ANRI = NR1 - \text{[number of preloaded regions which do not allow any insertions]}
\]
Figure 2-8. Flow Chart for Formations of Regions
When the number of regions is 1, all of the system terminations are assigned to the region and no other region formation process is performed. Otherwise, the program starts assigning system terminations to regions (in a cardinal order) which allow their entries.

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

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

\[
TPR1 = TPR1 - TPFS
\]

\[
TPR = TPR1/(ANR1 - 1.)
\]

\[
TPRL = 0.9 \times TPR
\]

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 TRFS is again tested against TPRL to determine if other additions are possible.

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

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

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

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

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

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

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

2.4.4 Selection of Regional Switchers

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

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

\[
SUMT = \sum_{i=1}^{NMBR} [TRAFFN(i) + TRAFFIT(i)] \times DIST(i,K)
\]

2-24
where

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

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

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

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 = NMBR \cdot (TRAFFDN(i) + TRAFFTN(i)) \cdot DIST(i, K)
NMBR = NUMBER OF SYSTEM TERMINATIONS IN THE REGION
DIST(i, K) = DISTANCE BETWEEN SYSTEM TERMINATIONS i AND K

NMBR = NMBR + 1
SUMT = SUMT + WCASE

START

WCase = 1 \times 10^{12}

IS RSC FOR THIS REGION PRESELECTED?

Y

K = 1

N

INCREMENT K BY 1

CALCULATE SUMT

IS SUMT > WCASE?

Y

WCase = SUMT
NRSC = K

N

IS K > NMBR?

Y

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

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

2.4.6 Optimization of Regional Networks

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

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

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

The following section describes the logic implemented in this program.

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

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

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

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 $K_1$, 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.

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

![Figure 2-15. Relationship among K, L, KI, and M Parameters](image)
The program first calculates the distance \textit{DREF} between system termination \textit{K} and the region switching center for each \textit{K}, and then the \textit{DTRY} which is half of the distance between system terminations \textit{KI} and \textit{M} for each combination.

If \textit{DTRY} is greater than \textit{DREF}, the program skips the process of calling on subroutine \textit{TRYLNK}. Otherwise, it calls on subroutine \textit{TRYLNK}. The purpose of subroutine \textit{TRYLNK} is to estimate the possible cost saving resulting from eliminating central link \textit{K}, and integrating sub-networks \textit{K} and \textit{L} by connecting system terminations \textit{KI} of \textit{K} and \textit{M} of \textit{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 \textit{K}, \textit{L}, \textit{KI}, and \textit{M}. A detailed description of functions performed by subroutine \textit{TRYLNK} is given in Paragraph 2.4.6.3. After all possible combinations for \textit{K}, \textit{L}, \textit{KI}, and \textit{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.

\textbf{2.4.6.3 Function Performed by Subroutine \textit{TRYLNK} for a Given Set of Values \textit{K}, \textit{KI}, \textit{L}, and \textit{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 sub-networks \textit{K} and \textit{L} needs to handle. It then estimates the required line configuration, \textit{LDUMMY}, by calling subroutine \textit{LINNUM} which has been described in Paragraph 2.4.5.1. Based on \textit{LDUMMY}, the program estimates the average response time and tests it against the user-provided response time limit by calling subroutine \textit{RSPNSE}. If the estimated response time is not satisfied, the program updates the line configuration \textit{LDUMMY} to the next higher line type and repeats the process of estimating its average response time and testing it against the given constraint. This process ends when either there is a satisfied line configuration or it is not possible to upgrade any further.

When a satisfied line configuration is obtained, the program continues to estimate its cost saving, based on the assumed integrated sub-network. If the resulting cost saving is better than the up-to-date best cost saving, it replaces all of the maximum saving parameters, which are used to keep tracking the up-to-date best network changes; it then returns to its calling routine. If there is no line configuration satisfying the response time constraint, the process stops and the program returns to its calling routine.
Figure 2-16. Flow Chart for Subroutine TRYLNK
2.4.6.4 Functions Performed by Subroutine RSPNSE. Figure 2-17 shows the flow chart of the subroutine RSPNSE. This subroutine calculates six items of delays: polling, message transmission time from a terminal to the central switches, input buffer queue time, service time, output buffer queue time, and returned message transmission time from the central switcher to the same terminal.

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

Figure 2-17. Flow Chart for Subroutine RSPNSE
2.4.6.5 **Network Updates by Subroutine UPNETW.** If there is a positive cost saving after trying all possible combinations for parameters K, KL, 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 ensure 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 ensure that it has three links to other RSCs and J is different from I. Another test is made to insure that I and J are connected to each other. If any of these conditions are not met, RSC node J + 1 is selected and these three tests are repeated.

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

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

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

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

2.5 OUTPUT

The STAGOM 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 STAGOM program. The amount of printout data depends upon the number of system terminations operated and also upon the number of functions executed in each specific run.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.5.1.10 **Initial Interregional Network.** If formation and optimization of the interregion network is required, the program will perform these functions and print its initial and optimized network. For each pair of ASCs, 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, NPI is a parameter variable which is used to make the number of system terminations allowed in a system a variable. A choice of NPI as 105, for example, dictates that the STACOM program can only be used in systems where 105 or less system terminations are under consideration. Any run which has a number of system terminations greater than 105 will result in either an abnormal run termination or a normal run termination with unwanted output.

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

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

3.3 RUN DESCRIPTION

3.3.1 Initialization and Setup

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

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

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

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

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

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

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

### 3.3.2 Run Options

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

<table>
<thead>
<tr>
<th>Item No.</th>
<th>Item Description</th>
<th>Names of Internal Variables/Arrays</th>
<th>Number of Cards Needed</th>
<th>Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>No. of regions under consideration</td>
<td>NR1</td>
<td>1</td>
<td>(I3)</td>
</tr>
<tr>
<td>2.</td>
<td>No. of system terminations, no. of data bases, and no. of distinctive cities under consideration.</td>
<td>N1, N7, NCITY</td>
<td>1</td>
<td>(3I5)</td>
</tr>
<tr>
<td>3.</td>
<td>IDs for data bases</td>
<td>NBASE(N7)</td>
<td>1</td>
<td>(3(1X,A4))</td>
</tr>
<tr>
<td>4.</td>
<td>V-H Coordinates for cities</td>
<td>IVERT(NCITY), IHORZ(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(N1,H), IADD(N1), MAPADR(N1), TRAFD(N1,2,N7)</td>
<td>a. N1 if N7≤2 b. 2N1 if N7&gt;2</td>
<td>(A4,1X,3A6,A4, I2,14,4F10.2/6F10.2)</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)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>b. 2xN2(2+N3x3x2) b. (4F9.2/I1/10F8.3/3x2) if non linear</td>
<td>b. (4F9.2/I1/10F8.3)</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) TAMD(N3), TAPD (N3)</td>
<td>N3</td>
<td>(5I4,2F7.5)</td>
</tr>
<tr>
<td>17.</td>
<td>No. of message types</td>
<td>NTYP</td>
<td>1</td>
<td>(I4)</td>
</tr>
<tr>
<td>18.</td>
<td>Message name, input message length, output message length, input percentage and output percentage with priority 1 and 2</td>
<td>MSGNAM (NTYP), MSLIN (NTYP) MSGOVT(NTYP), RATIOI(NTYP,2) RATIO(NTYP,2)</td>
<td>NTYP</td>
<td>(A6,2(I4,2F6.3))</td>
</tr>
<tr>
<td>19.</td>
<td>Average CPU service time per transaction</td>
<td>CPUAVG</td>
<td>1</td>
<td>(F7.4)</td>
</tr>
<tr>
<td>20.</td>
<td>Preloading system terminations and/or preselecting regional centers</td>
<td>NCODE, NSTATE, NREQQ</td>
<td>Variable</td>
<td>(I1,1X,A4,A5)</td>
</tr>
<tr>
<td>21.</td>
<td>System Center 1</td>
<td>NSCC1</td>
<td>1</td>
<td>(A4)</td>
</tr>
</tbody>
</table>
### Table 3-1. Formats for Input Data
(Continuation 3)

<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>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 #E6 after interrupting the line communication by pressing the BREAK key will terminate a program execution at any time. On the other hand, the EXEC-$ may abort a program execution when certain serious violations occur during its execution, e.g., number of punch cards exceeding the limit on the run card.

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

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

3.4 SAMPLE RUN

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

3.4.1 Run Stream

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

```plaintext
@RUN J6L,J6G3YL,51928,20,90/1000
@ASG,UP 100
@SYM,P PUNCH$,G9PLTF
@QT LEE.STACOM
@ADD LEE.DATA
@BRKPT 100
@FREE 100
@SYM,U 100,T4
@FIN
```

The first control statement is a run request which specifies its run ID as J6L, identifies its account number as J6G3YL, 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>5192A*STACOM101.INPUT/0777</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 25 4 358</td>
</tr>
<tr>
<td>1 P A L E S T I N E</td>
</tr>
<tr>
<td>5 A N D R E W S</td>
</tr>
<tr>
<td>6 L U P K I N</td>
</tr>
<tr>
<td>7 R O C K P O R T</td>
</tr>
<tr>
<td>8 A R C H E R C I T Y</td>
</tr>
<tr>
<td>9 J O U R D A N T O N</td>
</tr>
<tr>
<td>10 B E L L V I L L E</td>
</tr>
<tr>
<td>11 M U L S H O E</td>
</tr>
<tr>
<td>12 S E Y M O U R</td>
</tr>
<tr>
<td>13 B E E V I L L E</td>
</tr>
<tr>
<td>14 B E L T O N</td>
</tr>
<tr>
<td>15 F O R T H O O D</td>
</tr>
<tr>
<td>16 H A R K E R H E I G H T S</td>
</tr>
<tr>
<td>17 K I L L E E N</td>
</tr>
<tr>
<td>18 N O L A N V I L L E</td>
</tr>
<tr>
<td>19 T E M P L E</td>
</tr>
<tr>
<td>20 A L A M O H E I G H T S</td>
</tr>
<tr>
<td>21 F T S A M H O U S T O N</td>
</tr>
<tr>
<td>22 L E O N V A L L E Y</td>
</tr>
<tr>
<td>23 S A N A N T O N I O</td>
</tr>
<tr>
<td>24 U N I V E R S A L C I T Y</td>
</tr>
<tr>
<td>25 C L I F T O N</td>
</tr>
<tr>
<td>26 M E R I D I A N</td>
</tr>
<tr>
<td>27 T E X A R K A N A</td>
</tr>
<tr>
<td>28 A L V I N</td>
</tr>
<tr>
<td>29 A N G E L T O N</td>
</tr>
<tr>
<td>30 C L I F T E</td>
</tr>
<tr>
<td>31 P R E F E R</td>
</tr>
<tr>
<td>33 P E R L A N D</td>
</tr>
<tr>
<td>34 B R Y A N</td>
</tr>
<tr>
<td>35 C O L L E G E S T A T I O N</td>
</tr>
<tr>
<td>36 A L P H I NE</td>
</tr>
<tr>
<td>37 F A L F U R R I A S</td>
</tr>
<tr>
<td>38 B R O W N W O O D</td>
</tr>
<tr>
<td>39 C A L D W E L L</td>
</tr>
<tr>
<td>40 P O R T L A V A C A</td>
</tr>
<tr>
<td>41 B R O W N S V I L L E</td>
</tr>
<tr>
<td>42 H A R L I N G E N</td>
</tr>
<tr>
<td>43 P O R T I S A R E L</td>
</tr>
<tr>
<td>44 S A N B E N I T O</td>
</tr>
<tr>
<td>45 L I N D E N</td>
</tr>
<tr>
<td>46 D I N M I T</td>
</tr>
<tr>
<td>47 A N A H U A C</td>
</tr>
<tr>
<td>48 J A C K S O N V I L L E</td>
</tr>
<tr>
<td>49 C H I L D R E S S</td>
</tr>
<tr>
<td>50 M O R T O N</td>
</tr>
<tr>
<td>51 R O B E R T L E E</td>
</tr>
<tr>
<td>52 C O L E M A N</td>
</tr>
<tr>
<td>53 F R I S C O</td>
</tr>
<tr>
<td>54 M C K I N N E Y</td>
</tr>
<tr>
<td>55 P L A N O</td>
</tr>
<tr>
<td>56 W E L L I N G T O N</td>
</tr>
</tbody>
</table>

3-13
Table 3-2. Input Data for the Example Run
(Continuation 1)

<p>| | | | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<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>57</td>
<td>COLUMBUS</td>
<td>COLORADO</td>
<td>9032</td>
<td>3740</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>
<tr>
<td>58</td>
<td>NEW BRAUNFELS</td>
<td>COMAL</td>
<td>9185</td>
<td>4018</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>
<tr>
<td>59</td>
<td>COMANCHE</td>
<td>COMANCHE</td>
<td>8735</td>
<td>4275</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>
<tr>
<td>60</td>
<td>GAINESVILLE</td>
<td>COOKE</td>
<td>8285</td>
<td>4162</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>
<tr>
<td>61</td>
<td>COPPERAS COVE</td>
<td>CORTELL</td>
<td>8844</td>
<td>4080</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>
<tr>
<td>62</td>
<td>GATESVILLE</td>
<td>CRANE</td>
<td>8771</td>
<td>4070</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>
<tr>
<td>63</td>
<td>CRANE</td>
<td>CRAMER</td>
<td>8173</td>
<td>4096</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>
<tr>
<td>64</td>
<td>OZONA</td>
<td>CROCKETT</td>
<td>9144</td>
<td>4692</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>
<tr>
<td>65</td>
<td>DALHART</td>
<td>DALLAS</td>
<td>8129</td>
<td>4289</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>
<tr>
<td>66</td>
<td>ADDISON</td>
<td>DALLAS</td>
<td>8956</td>
<td>4016</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>
<tr>
<td>67</td>
<td>CEDAR HILL</td>
<td>DALLAS</td>
<td>8945</td>
<td>4017</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>
<tr>
<td>68</td>
<td>DALLAS</td>
<td>DALLAS</td>
<td>8436</td>
<td>4034</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>
<tr>
<td>69</td>
<td>DESOTO</td>
<td>DALLAS</td>
<td>8478</td>
<td>4030</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>
<tr>
<td>70</td>
<td>DUNCANVILLE</td>
<td>DALLAS</td>
<td>8469</td>
<td>4044</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>
<tr>
<td>71</td>
<td>FARMERS BRANCH</td>
<td>DALLAS</td>
<td>8414</td>
<td>4062</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>
<tr>
<td>72</td>
<td>GARLAND</td>
<td>DALLAS</td>
<td>8405</td>
<td>4018</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>
<tr>
<td>73</td>
<td>GRAND PRAIRIE</td>
<td>DALLAS</td>
<td>8485</td>
<td>4066</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>
<tr>
<td>74</td>
<td>HIGHLAND PARK</td>
<td>DALLAS</td>
<td>8436</td>
<td>4034</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>
<tr>
<td>75</td>
<td>IRVING</td>
<td>DALLAS</td>
<td>8440</td>
<td>4064</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>
<tr>
<td>76</td>
<td>LANCASTER</td>
<td>DALLAS</td>
<td>8470</td>
<td>4013</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>
<tr>
<td>77</td>
<td>MESQUITE</td>
<td>DALLAS</td>
<td>8426</td>
<td>4000</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>
<tr>
<td>78</td>
<td>RICHARDSON</td>
<td>DALLAS</td>
<td>8399</td>
<td>4035</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>
<tr>
<td>79</td>
<td>SEAGOVILLE</td>
<td>DALLAS</td>
<td>8447</td>
<td>3980</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>
<tr>
<td>80</td>
<td>UNIVERSITY PARK</td>
<td>DALLAS</td>
<td>8436</td>
<td>4034</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>
<tr>
<td>81</td>
<td>LAKE SA</td>
<td>DAWSON</td>
<td>8779</td>
<td>4919</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>
<tr>
<td>82</td>
<td>HEREFORD</td>
<td>DEAF SMITH</td>
<td>8379</td>
<td>4143</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>
<tr>
<td>83</td>
<td>DENTON</td>
<td>DENTON</td>
<td>8379</td>
<td>4143</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>
<tr>
<td>84</td>
<td>LEWISVILLE</td>
<td>DENTON</td>
<td>8399</td>
<td>4093</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>
<tr>
<td>85</td>
<td>DFW</td>
<td>DENTON</td>
<td>8372</td>
<td>4127</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>
<tr>
<td>86</td>
<td>CURRO</td>
<td>DFW</td>
<td>8372</td>
<td>4127</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>
<tr>
<td>87</td>
<td>SPUR</td>
<td>DFW</td>
<td>8372</td>
<td>4127</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>
<tr>
<td>88</td>
<td>EASTLAND</td>
<td>DUVAL</td>
<td>9542</td>
<td>3988</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>
<tr>
<td>89</td>
<td>EASTLAND</td>
<td>EASTLAND</td>
<td>8649</td>
<td>4352</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>
<tr>
<td>90</td>
<td>ECTOR</td>
<td>ECTOR</td>
<td>8982</td>
<td>4030</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>
<tr>
<td>91</td>
<td>ENNIS</td>
<td>ELLIS</td>
<td>8514</td>
<td>3970</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>
<tr>
<td>92</td>
<td>WAXAHACHIE</td>
<td>ELLIS</td>
<td>8517</td>
<td>4011</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>
<tr>
<td>93</td>
<td>EL PASO</td>
<td>EL PASO</td>
<td>9243</td>
<td>5655</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>
<tr>
<td>94</td>
<td>STEPHENVILLE</td>
<td>ERATH</td>
<td>8645</td>
<td>4232</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>
<tr>
<td>95</td>
<td>MARLIN</td>
<td>FALLS</td>
<td>8739</td>
<td>3931</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>
<tr>
<td>96</td>
<td>BONHAM</td>
<td>FANNIN</td>
<td>8234</td>
<td>3996</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>
<tr>
<td>97</td>
<td>RORY</td>
<td>FISHER</td>
<td>8679</td>
<td>4646</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>
<tr>
<td>98</td>
<td>FLOYDADA</td>
<td>FLOYD</td>
<td>8486</td>
<td>4902</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>
<tr>
<td>99</td>
<td>FAIRFIELD</td>
<td>FREESTONE</td>
<td>8602</td>
<td>3320</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>
<tr>
<td>100</td>
<td>PEARL</td>
<td>FRIO</td>
<td>9374</td>
<td>4179</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>
<tr>
<td>101</td>
<td>RICHMOND</td>
<td>FT BEND</td>
<td>9009</td>
<td>3508</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>
<tr>
<td>102</td>
<td>ROSENBERG</td>
<td>FT BEND</td>
<td>9009</td>
<td>3508</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>
<tr>
<td>103</td>
<td>SEMINOLE</td>
<td>GAINES</td>
<td>8522</td>
<td>5840</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>
<tr>
<td>104</td>
<td>FRIENDSWOOD</td>
<td>GALVESTON</td>
<td>8969</td>
<td>3449</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>
<tr>
<td>105</td>
<td>GALVESTON</td>
<td>GALVESTON</td>
<td>8965</td>
<td>3397</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>
<tr>
<td>106</td>
<td>HISTOCOCK</td>
<td>GALVESTON</td>
<td>8975</td>
<td>3419</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>
<tr>
<td>107</td>
<td>LA MARQUE</td>
<td>GALVESTON</td>
<td>8975</td>
<td>3420</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>
<tr>
<td>108</td>
<td>LEAGUE CITY</td>
<td>GALVESTON</td>
<td>8967</td>
<td>3468</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>
<tr>
<td>109</td>
<td>TEXAS CITY</td>
<td>GALVESTON</td>
<td>8975</td>
<td>3429</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>
<tr>
<td>110</td>
<td>POST</td>
<td>GARZA</td>
<td>8650</td>
<td>4645</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>
<tr>
<td>111</td>
<td>FREDERICKSBURG</td>
<td>GILLESPIE</td>
<td>9079</td>
<td>4194</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>
<tr>
<td>112</td>
<td>GOLIAD</td>
<td>GOLIAD</td>
<td>9301</td>
<td>3807</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>
<tr>
<td>113</td>
<td>GONZALEZ</td>
<td>GONZALEZ</td>
<td>9137</td>
<td>3848</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>
<tr>
<td></td>
<td>PANAMA</td>
<td>GRAY</td>
<td>8148</td>
<td>4852</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>
Table 3-2. Input Data for the Example Run  
(Continuation 2)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>114</td>
<td>DENISON</td>
<td>115</td>
<td>SHERMAN</td>
<td>116</td>
</tr>
<tr>
<td>117</td>
<td>KILGORE</td>
<td>118</td>
<td>LONGVIEW</td>
<td>119</td>
</tr>
<tr>
<td>120</td>
<td>SEGUIN</td>
<td>121</td>
<td>PLAINVIEW</td>
<td>122</td>
</tr>
<tr>
<td>123</td>
<td>SPARKMAN</td>
<td>124</td>
<td>GUANAH</td>
<td>125</td>
</tr>
<tr>
<td>126</td>
<td>SILSBEE</td>
<td>127</td>
<td>DAYTON</td>
<td>128</td>
</tr>
<tr>
<td>129</td>
<td>DEEP PARK</td>
<td>130</td>
<td>GALENA PARK</td>
<td>131</td>
</tr>
<tr>
<td>132</td>
<td>HUMBLE</td>
<td>133</td>
<td>JARINTO CITY</td>
<td>134</td>
</tr>
<tr>
<td>135</td>
<td>KATY</td>
<td>136</td>
<td>LA PORTE</td>
<td>137</td>
</tr>
<tr>
<td>140</td>
<td>SPRING VALLEY</td>
<td>141</td>
<td>TOHATT</td>
<td>142</td>
</tr>
<tr>
<td>147</td>
<td>HASKELL</td>
<td>148</td>
<td>SAN MARCOS</td>
<td>149</td>
</tr>
<tr>
<td>150</td>
<td>ATHENS</td>
<td>151</td>
<td>DONNA</td>
<td>152</td>
</tr>
<tr>
<td>153</td>
<td>Hidalgo</td>
<td>154</td>
<td>MCALLEN</td>
<td>155</td>
</tr>
<tr>
<td>156</td>
<td>MISSION</td>
<td>157</td>
<td>PHARR</td>
<td>158</td>
</tr>
<tr>
<td>159</td>
<td>HILLSBORO</td>
<td>160</td>
<td>LEVELLAND</td>
<td>161</td>
</tr>
</tbody>
</table>
| 162 | BURG | 163 | COMMERCE | 164 | GRAF 
| 165 | BORGER | 166 | JACKSBORO | 167 | EDNA |
| 168 | BEAUMONT | 169 | NENDELAND | 170 | PORT ARTHUR |

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

<table>
<thead>
<tr>
<th></th>
<th>City</th>
<th>County</th>
<th>Zip Code</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>171</td>
<td>ALICE</td>
<td>WELLS</td>
<td>9535</td>
<td>3845</td>
</tr>
<tr>
<td>172</td>
<td>BURLESON</td>
<td>JOHNSON</td>
<td>8522</td>
<td>4103</td>
</tr>
<tr>
<td>173</td>
<td>CLFURNE</td>
<td>JOHNSON</td>
<td>8563</td>
<td>4112</td>
</tr>
<tr>
<td>174</td>
<td>ANSON</td>
<td>JONES</td>
<td>8647</td>
<td>4563</td>
</tr>
<tr>
<td>175</td>
<td>STAMFORD</td>
<td>JONES</td>
<td>8603</td>
<td>4562</td>
</tr>
<tr>
<td>176</td>
<td>KARNES CITY</td>
<td>KARNES</td>
<td>9264</td>
<td>3915</td>
</tr>
<tr>
<td>177</td>
<td>KAUFMAN</td>
<td>KAUFMAN</td>
<td>8442</td>
<td>3936</td>
</tr>
<tr>
<td>178</td>
<td>TERRELL</td>
<td>KAUFMAN</td>
<td>8410</td>
<td>3943</td>
</tr>
<tr>
<td>179</td>
<td>BOFRINE</td>
<td>KENDALL</td>
<td>9168</td>
<td>4123</td>
</tr>
<tr>
<td>180</td>
<td>JAYTON</td>
<td>KENT</td>
<td>8589</td>
<td>4718</td>
</tr>
<tr>
<td>181</td>
<td>KEMPVILLE</td>
<td>KERR</td>
<td>9143</td>
<td>4226</td>
</tr>
<tr>
<td>182</td>
<td>JUNCTION</td>
<td>KIMBLE</td>
<td>9097</td>
<td>4373</td>
</tr>
<tr>
<td>183</td>
<td>KINGSVILLE</td>
<td>KLEBERG</td>
<td>9566</td>
<td>3801</td>
</tr>
<tr>
<td>184</td>
<td>BENJAMIN</td>
<td>KNOX</td>
<td>8472</td>
<td>4619</td>
</tr>
<tr>
<td>185</td>
<td>PARTS</td>
<td>LAMAR</td>
<td>8173</td>
<td>3977</td>
</tr>
<tr>
<td>186</td>
<td>LITTLEFIELD</td>
<td>LAMB</td>
<td>8558</td>
<td>5040</td>
</tr>
<tr>
<td>187</td>
<td>OLTON</td>
<td>LAMB</td>
<td>8400</td>
<td>6046</td>
</tr>
<tr>
<td>188</td>
<td>LAMPAASAS</td>
<td>LAMPAAS</td>
<td>8875</td>
<td>4137</td>
</tr>
<tr>
<td>189</td>
<td>HALLETTSVILLE</td>
<td>LAVACA</td>
<td>9114</td>
<td>3780</td>
</tr>
<tr>
<td>190</td>
<td>YOAKUM</td>
<td>LAVACA</td>
<td>9117</td>
<td>3914</td>
</tr>
<tr>
<td>191</td>
<td>CENTERVILLE</td>
<td>LEMON</td>
<td>8682</td>
<td>3768</td>
</tr>
<tr>
<td>192</td>
<td>CLEVELAND</td>
<td>LIBERTY</td>
<td>8001</td>
<td>3540</td>
</tr>
<tr>
<td>193</td>
<td>LIBERTY</td>
<td>LIBERTY</td>
<td>8A35</td>
<td>3463</td>
</tr>
<tr>
<td>194</td>
<td>MEYLA</td>
<td>LIMESTONE</td>
<td>8635</td>
<td>3460</td>
</tr>
<tr>
<td>195</td>
<td>GEORGE WEST</td>
<td>LIVE OAK</td>
<td>9419</td>
<td>3910</td>
</tr>
<tr>
<td>196</td>
<td>LUBBOCK</td>
<td>LUBBOCK</td>
<td>8508</td>
<td>4962</td>
</tr>
<tr>
<td>197</td>
<td>SLATON</td>
<td>LUBBOCK</td>
<td>8616</td>
<td>4916</td>
</tr>
<tr>
<td>198</td>
<td>TAHOKA</td>
<td>LYNN</td>
<td>8660</td>
<td>4924</td>
</tr>
<tr>
<td>199</td>
<td>MADISONVILLE</td>
<td>MADISON</td>
<td>8780</td>
<td>3733</td>
</tr>
<tr>
<td>200</td>
<td>JEFFERSON</td>
<td>MARION</td>
<td>8267</td>
<td>3618</td>
</tr>
<tr>
<td>201</td>
<td>BAY CITY</td>
<td>MATAGORDA</td>
<td>9135</td>
<td>3578</td>
</tr>
<tr>
<td>202</td>
<td>EELE PASS</td>
<td>MAVERICK</td>
<td>9505</td>
<td>4370</td>
</tr>
<tr>
<td>203</td>
<td>BRADY</td>
<td>MCCULLOCH</td>
<td>893A</td>
<td>4344</td>
</tr>
<tr>
<td>204</td>
<td>BELLMEAD</td>
<td>MCLENNAN</td>
<td>8706</td>
<td>3903</td>
</tr>
<tr>
<td>205</td>
<td>BEVERLY HILLS</td>
<td>MCLENNAN</td>
<td>8706</td>
<td>3903</td>
</tr>
<tr>
<td>206</td>
<td>WACO</td>
<td>MCLENNAN</td>
<td>8706</td>
<td>3903</td>
</tr>
<tr>
<td>207</td>
<td>WOODWAY</td>
<td>MCLENNAN</td>
<td>8706</td>
<td>3903</td>
</tr>
<tr>
<td>208</td>
<td>HONDO</td>
<td>MEDINA</td>
<td>9285</td>
<td>4174</td>
</tr>
<tr>
<td>209</td>
<td>MENARD</td>
<td>MENARD</td>
<td>9011</td>
<td>4417</td>
</tr>
<tr>
<td>210</td>
<td>MILTOWN</td>
<td>MIDLAND</td>
<td>8934</td>
<td>4897</td>
</tr>
<tr>
<td>211</td>
<td>CAMPRON</td>
<td>MILAN</td>
<td>8355</td>
<td>3910</td>
</tr>
<tr>
<td>212</td>
<td>ROCKDALE</td>
<td>MILAN</td>
<td>8877</td>
<td>3808</td>
</tr>
<tr>
<td>213</td>
<td>COLORADO CITY</td>
<td>MITCHELL</td>
<td>8781</td>
<td>4706</td>
</tr>
<tr>
<td>214</td>
<td>BOWIE</td>
<td>MONTAIGUE</td>
<td>8351</td>
<td>4275</td>
</tr>
<tr>
<td>215</td>
<td>MONTAGUE</td>
<td>MONTAIGUE</td>
<td>8323</td>
<td>4611</td>
</tr>
<tr>
<td>216</td>
<td>CONROE</td>
<td>MONTGOMERY</td>
<td>8832</td>
<td>3610</td>
</tr>
<tr>
<td>217</td>
<td>DUNAS</td>
<td>MOORE</td>
<td>8141</td>
<td>5144</td>
</tr>
<tr>
<td>218</td>
<td>DANGERFIELD</td>
<td>MORRIS</td>
<td>8240</td>
<td>3704</td>
</tr>
<tr>
<td>219</td>
<td>NACOGDOCHES</td>
<td>NACOGDOCHES</td>
<td>8515</td>
<td>3669</td>
</tr>
<tr>
<td>220</td>
<td>CONSCANA</td>
<td>NAVARRO</td>
<td>8553</td>
<td>3921</td>
</tr>
<tr>
<td>221</td>
<td>SWEETWATER</td>
<td>NOLAN</td>
<td>8737</td>
<td>4632</td>
</tr>
<tr>
<td>222</td>
<td>CORPUS CHRISTI</td>
<td>NUECES</td>
<td>9475</td>
<td>3739</td>
</tr>
<tr>
<td>223</td>
<td>RONSTOWN</td>
<td>NUECES</td>
<td>9496</td>
<td>3786</td>
</tr>
<tr>
<td>224</td>
<td>PEPPYTON</td>
<td>OCHILREEF</td>
<td>7962</td>
<td>4987</td>
</tr>
<tr>
<td>225</td>
<td>VEGA</td>
<td>OLDHAM</td>
<td>8292</td>
<td>5177</td>
</tr>
<tr>
<td>226</td>
<td>ORANGE</td>
<td>ORANGE</td>
<td>8746</td>
<td>3281</td>
</tr>
<tr>
<td>227</td>
<td>MINERAL WELLS</td>
<td>PARKER</td>
<td>8520</td>
<td>4261</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run
(Continuation 4)

<table>
<thead>
<tr>
<th>Code</th>
<th>Town</th>
<th>County</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>228</td>
<td>Carthage</td>
<td>Panola</td>
<td>31°56'</td>
<td>-97°46'</td>
</tr>
<tr>
<td>229</td>
<td>Weatherford</td>
<td>Parker</td>
<td>32°21'</td>
<td>-97°21'</td>
</tr>
<tr>
<td>230</td>
<td>Farnell</td>
<td>Parker</td>
<td>31°45'</td>
<td>-97°52'</td>
</tr>
<tr>
<td>231</td>
<td>Friona</td>
<td>Pecos</td>
<td>32°37'</td>
<td>-96°54'</td>
</tr>
<tr>
<td>232</td>
<td>Fort Stockton</td>
<td>Pecos</td>
<td>32°07'</td>
<td>-97°54'</td>
</tr>
<tr>
<td>233</td>
<td>Amarillo</td>
<td>Potter</td>
<td>32°26'</td>
<td>-97°76'</td>
</tr>
<tr>
<td>234</td>
<td>Canyon</td>
<td>Randall</td>
<td>33°17'</td>
<td>-98°75'</td>
</tr>
<tr>
<td>235</td>
<td>Clarksville</td>
<td>Red River</td>
<td>34°17'</td>
<td>-99°09'</td>
</tr>
<tr>
<td>236</td>
<td>Pecos</td>
<td>Reeves</td>
<td>91°36'</td>
<td>31°11'</td>
</tr>
<tr>
<td>237</td>
<td>Big Lake</td>
<td>Regan</td>
<td>90°62'</td>
<td>47°23'</td>
</tr>
<tr>
<td>238</td>
<td>Hearne</td>
<td>Robertson</td>
<td>88°02'</td>
<td>38°46'</td>
</tr>
<tr>
<td>239</td>
<td>Rockwall</td>
<td>Rockwall</td>
<td>83°48'</td>
<td>30°49'</td>
</tr>
<tr>
<td>240</td>
<td>Ballinger</td>
<td>Runnels</td>
<td>88°55'</td>
<td>44°08'</td>
</tr>
<tr>
<td>241</td>
<td>Henderson</td>
<td>Rusk</td>
<td>84°20'</td>
<td>36°40'</td>
</tr>
<tr>
<td>242</td>
<td>Aransas Pass</td>
<td>San Patroco</td>
<td>94°37'</td>
<td>37°00'</td>
</tr>
<tr>
<td>243</td>
<td>Gregory</td>
<td>San Patroco</td>
<td>94°55'</td>
<td>37°31'</td>
</tr>
<tr>
<td>244</td>
<td>Inglewood</td>
<td>San Patroco</td>
<td>94°47'</td>
<td>37°11'</td>
</tr>
<tr>
<td>245</td>
<td>Portland</td>
<td>San Patroco</td>
<td>94°45'</td>
<td>37°31'</td>
</tr>
<tr>
<td>246</td>
<td>Stinnett</td>
<td>San Patroco</td>
<td>94°36'</td>
<td>37°77'</td>
</tr>
<tr>
<td>247</td>
<td>Eldorado</td>
<td>Schleicher</td>
<td>90°76'</td>
<td>45°47'</td>
</tr>
<tr>
<td>248</td>
<td>Snyder</td>
<td>Scurry</td>
<td>87°18'</td>
<td>47°37'</td>
</tr>
<tr>
<td>249</td>
<td>Stratford</td>
<td>Sherman</td>
<td>80°49'</td>
<td>41°04'</td>
</tr>
<tr>
<td>250</td>
<td>Tyler</td>
<td>Smith</td>
<td>84°17'</td>
<td>37°54'</td>
</tr>
<tr>
<td>251</td>
<td>Breckenridge</td>
<td>Sterling City</td>
<td>85°42'</td>
<td>44°34'</td>
</tr>
<tr>
<td>252</td>
<td>Sterlity</td>
<td>Sterling City</td>
<td>89°00'</td>
<td>46°36'</td>
</tr>
<tr>
<td>253</td>
<td>Aspermont</td>
<td>Stephenville</td>
<td>85°69'</td>
<td>46°50'</td>
</tr>
<tr>
<td>254</td>
<td>Tulia</td>
<td>Swisher</td>
<td>83°37'</td>
<td>50°16'</td>
</tr>
<tr>
<td>255</td>
<td>Arlington</td>
<td>Tarrant</td>
<td>84°72'</td>
<td>40°85'</td>
</tr>
<tr>
<td>256</td>
<td>Bedford</td>
<td>Tarrant</td>
<td>84°47'</td>
<td>40°92'</td>
</tr>
<tr>
<td>257</td>
<td>Colleyville</td>
<td>Tarrant</td>
<td>84°47'</td>
<td>41°17'</td>
</tr>
<tr>
<td>258</td>
<td>Crowley</td>
<td>Tarrant</td>
<td>85°10'</td>
<td>41°18'</td>
</tr>
<tr>
<td>259</td>
<td>Euless</td>
<td>Tarrant</td>
<td>84°47'</td>
<td>40°92'</td>
</tr>
<tr>
<td>260</td>
<td>Forrest Hill</td>
<td>Tarrant</td>
<td>84°79'</td>
<td>41°22'</td>
</tr>
<tr>
<td>261</td>
<td>Fort Worth</td>
<td>Tarrant</td>
<td>84°25'</td>
<td>40°94'</td>
</tr>
<tr>
<td>262</td>
<td>Grapevine</td>
<td>Tarrant</td>
<td>84°79'</td>
<td>41°22'</td>
</tr>
<tr>
<td>263</td>
<td>Haltom City</td>
<td>Tarrant</td>
<td>84°47'</td>
<td>41°17'</td>
</tr>
<tr>
<td>264</td>
<td>Hurst</td>
<td>Tarrant</td>
<td>84°47'</td>
<td>41°17'</td>
</tr>
<tr>
<td>265</td>
<td>Lake Worth</td>
<td>Tarrant</td>
<td>84°71'</td>
<td>41°58'</td>
</tr>
<tr>
<td>266</td>
<td>No richland Hls</td>
<td>Tarrant</td>
<td>84°47'</td>
<td>41°17'</td>
</tr>
<tr>
<td>267</td>
<td>Richland Hills</td>
<td>Tarrant</td>
<td>84°79'</td>
<td>41°22'</td>
</tr>
<tr>
<td>268</td>
<td>Southlake</td>
<td>Tarrant</td>
<td>84°25'</td>
<td>40°94'</td>
</tr>
<tr>
<td>269</td>
<td>White Settle</td>
<td>Tarrant</td>
<td>84°85'</td>
<td>41°53'</td>
</tr>
<tr>
<td>270</td>
<td>Argyle</td>
<td>Taylor</td>
<td>86°39'</td>
<td>45°13'</td>
</tr>
<tr>
<td>271</td>
<td>Sanderson</td>
<td>Terrell</td>
<td>93°33'</td>
<td>48°16'</td>
</tr>
<tr>
<td>272</td>
<td>Brownfield</td>
<td>Terry</td>
<td>87°05'</td>
<td>50°07'</td>
</tr>
<tr>
<td>273</td>
<td>Mt Pleasant</td>
<td>Titus</td>
<td>82°34'</td>
<td>37°55'</td>
</tr>
<tr>
<td>274</td>
<td>San Angelo</td>
<td>Tom Green</td>
<td>89°44'</td>
<td>48°43'</td>
</tr>
<tr>
<td>275</td>
<td>Austin</td>
<td>Travis</td>
<td>90°05'</td>
<td>40°06'</td>
</tr>
<tr>
<td>276</td>
<td>Gilmer</td>
<td>Upshur</td>
<td>83°17'</td>
<td>37°16'</td>
</tr>
<tr>
<td>277</td>
<td>Rankin</td>
<td>Upton</td>
<td>90°49'</td>
<td>48°11'</td>
</tr>
<tr>
<td>278</td>
<td>Uvalde</td>
<td>Uvalde</td>
<td>93°57'</td>
<td>42°79'</td>
</tr>
<tr>
<td>279</td>
<td>Del Rio</td>
<td>Val Verde</td>
<td>93°49'</td>
<td>44°20'</td>
</tr>
<tr>
<td>280</td>
<td>Canton</td>
<td>Van Zandt</td>
<td>84°14'</td>
<td>38°58'</td>
</tr>
<tr>
<td>281</td>
<td>Victoria</td>
<td>Victoria</td>
<td>92°45'</td>
<td>37°48'</td>
</tr>
<tr>
<td>282</td>
<td>Huntsville</td>
<td>Walker</td>
<td>87°56'</td>
<td>36°52'</td>
</tr>
<tr>
<td>283</td>
<td>Hempstead</td>
<td>Waller</td>
<td>89°23'</td>
<td>36°01'</td>
</tr>
<tr>
<td>284</td>
<td>Monahans</td>
<td>Ward</td>
<td>90°66'</td>
<td>50°55'</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run (Continuation 5)

<table>
<thead>
<tr>
<th>No.</th>
<th>Location</th>
<th>County</th>
<th>Population</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>285</td>
<td>Brenham</td>
<td>Washington</td>
<td>8942</td>
<td>3759</td>
</tr>
<tr>
<td>286</td>
<td>Laredo</td>
<td>Webb</td>
<td>9681</td>
<td>4099</td>
</tr>
<tr>
<td>287</td>
<td>Pierce</td>
<td>Wharton</td>
<td>9115</td>
<td>3649</td>
</tr>
<tr>
<td>288</td>
<td>Wharton</td>
<td>Wharton</td>
<td>9078</td>
<td>3679</td>
</tr>
<tr>
<td>289</td>
<td>Sharrock</td>
<td>Wheeler</td>
<td>8170</td>
<td>4008</td>
</tr>
<tr>
<td>290</td>
<td>Burkburnett</td>
<td>Wichita</td>
<td>8240</td>
<td>4440</td>
</tr>
<tr>
<td>291</td>
<td>Wichita Falls</td>
<td>Wichita</td>
<td>8326</td>
<td>4413</td>
</tr>
<tr>
<td>292</td>
<td>Vernon</td>
<td>Wilbarger</td>
<td>8326</td>
<td>4467</td>
</tr>
<tr>
<td>293</td>
<td>Raymondville</td>
<td>Willacy</td>
<td>9768</td>
<td>3793</td>
</tr>
<tr>
<td>294</td>
<td>Floresville</td>
<td>Wilson</td>
<td>9241</td>
<td>3979</td>
</tr>
<tr>
<td>295</td>
<td>Kermit</td>
<td>Winkler</td>
<td>9024</td>
<td>5060</td>
</tr>
<tr>
<td>296</td>
<td>Decatur</td>
<td>Wise</td>
<td>8309</td>
<td>4205</td>
</tr>
<tr>
<td>297</td>
<td>Winnsboro</td>
<td>Wood</td>
<td>8295</td>
<td>3794</td>
</tr>
<tr>
<td>298</td>
<td>Denver City</td>
<td>Yoakum</td>
<td>8781</td>
<td>5098</td>
</tr>
<tr>
<td>299</td>
<td>Graham</td>
<td>Young</td>
<td>8492</td>
<td>4365</td>
</tr>
<tr>
<td>300</td>
<td>Olney</td>
<td>Young</td>
<td>8450</td>
<td>4414</td>
</tr>
<tr>
<td>301</td>
<td>Crystal City</td>
<td>Zavala</td>
<td>9466</td>
<td>4246</td>
</tr>
<tr>
<td>302</td>
<td>Aransas Pass</td>
<td>Aransas</td>
<td>9437</td>
<td>3796</td>
</tr>
<tr>
<td>303</td>
<td>Pleasanton</td>
<td>Atascosa</td>
<td>9290</td>
<td>4027</td>
</tr>
<tr>
<td>304</td>
<td>Banera</td>
<td>Randera</td>
<td>9255</td>
<td>4190</td>
</tr>
<tr>
<td>305</td>
<td>Bastrop</td>
<td>Bastrop</td>
<td>9067</td>
<td>3690</td>
</tr>
<tr>
<td>306</td>
<td>Marble Falls</td>
<td>Burnet</td>
<td>8960</td>
<td>4115</td>
</tr>
<tr>
<td>307</td>
<td>Lockhart</td>
<td>Caldwell</td>
<td>9077</td>
<td>3954</td>
</tr>
<tr>
<td>308</td>
<td>Luling</td>
<td>Caldwell</td>
<td>9117</td>
<td>3933</td>
</tr>
<tr>
<td>309</td>
<td>Baird</td>
<td>Callahan</td>
<td>8688</td>
<td>4450</td>
</tr>
<tr>
<td>310</td>
<td>Pittsburg</td>
<td>Camp</td>
<td>8264</td>
<td>3742</td>
</tr>
<tr>
<td>311</td>
<td>Panhandle</td>
<td>Carson</td>
<td>8210</td>
<td>5004</td>
</tr>
<tr>
<td>312</td>
<td>Atlanta</td>
<td>Cass</td>
<td>8182</td>
<td>3618</td>
</tr>
<tr>
<td>313</td>
<td>Rusk</td>
<td>Cherokee</td>
<td>8515</td>
<td>3572</td>
</tr>
<tr>
<td>314</td>
<td>Henrietta</td>
<td>Clay</td>
<td>8323</td>
<td>4354</td>
</tr>
<tr>
<td>315</td>
<td>Crosbyton</td>
<td>Crosby</td>
<td>8545</td>
<td>4462</td>
</tr>
<tr>
<td>316</td>
<td>Carrollton</td>
<td>Dallas</td>
<td>8410</td>
<td>4066</td>
</tr>
<tr>
<td>317</td>
<td>Cooper</td>
<td>Delta</td>
<td>8241</td>
<td>3806</td>
</tr>
<tr>
<td>318</td>
<td>Carrizo Sprs.</td>
<td>Dimmit</td>
<td>9580</td>
<td>4290</td>
</tr>
<tr>
<td>319</td>
<td>Cico</td>
<td>Eastland</td>
<td>8662</td>
<td>4377</td>
</tr>
<tr>
<td>320</td>
<td>La Grange</td>
<td>Fayette</td>
<td>9016</td>
<td>3813</td>
</tr>
<tr>
<td>321</td>
<td>Mt Vernon</td>
<td>Franklin</td>
<td>8246</td>
<td>3801</td>
</tr>
<tr>
<td>322</td>
<td>Anderson</td>
<td>Grimes</td>
<td>8866</td>
<td>3708</td>
</tr>
<tr>
<td>323</td>
<td>Memphis</td>
<td>Hall</td>
<td>8297</td>
<td>4021</td>
</tr>
<tr>
<td>324</td>
<td>Alamo</td>
<td>Hidalgo</td>
<td>9854</td>
<td>3774</td>
</tr>
<tr>
<td>325</td>
<td>Granbury</td>
<td>Hood</td>
<td>8572</td>
<td>4178</td>
</tr>
<tr>
<td>326</td>
<td>Crockett</td>
<td>Houston</td>
<td>8634</td>
<td>3645</td>
</tr>
<tr>
<td>327</td>
<td>Stinnett</td>
<td>Hutchison</td>
<td>8117</td>
<td>5054</td>
</tr>
<tr>
<td>328</td>
<td>Jasper</td>
<td>Jasper</td>
<td>8643</td>
<td>3339</td>
</tr>
<tr>
<td>329</td>
<td>Groves</td>
<td>Jefferson</td>
<td>8789</td>
<td>3316</td>
</tr>
<tr>
<td>330</td>
<td>Port Neches</td>
<td>Jefferson</td>
<td>8789</td>
<td>3316</td>
</tr>
<tr>
<td>331</td>
<td>Cotulla</td>
<td>Lasalle</td>
<td>9476</td>
<td>4120</td>
</tr>
<tr>
<td>332</td>
<td>Girdings</td>
<td>Lef</td>
<td>8968</td>
<td>3448</td>
</tr>
<tr>
<td>333</td>
<td>Gofzech</td>
<td>Limestone</td>
<td>6671</td>
<td>3088</td>
</tr>
<tr>
<td>334</td>
<td>Llano</td>
<td>Llano</td>
<td>8970</td>
<td>4159</td>
</tr>
<tr>
<td>335</td>
<td>Newton</td>
<td>Newton</td>
<td>8460</td>
<td>3353</td>
</tr>
<tr>
<td>336</td>
<td>Bridge City</td>
<td>Orange</td>
<td>8774</td>
<td>3295</td>
</tr>
<tr>
<td>337</td>
<td>Vinor</td>
<td>Orange</td>
<td>8761</td>
<td>3334</td>
</tr>
<tr>
<td>338</td>
<td>Palo Pinto</td>
<td>Plo Pinto</td>
<td>8541</td>
<td>4291</td>
</tr>
<tr>
<td>339</td>
<td>Livingston</td>
<td>Polk</td>
<td>8716</td>
<td>3543</td>
</tr>
<tr>
<td>340</td>
<td>Refugio</td>
<td>Refugio</td>
<td>9345</td>
<td>3747</td>
</tr>
<tr>
<td>341</td>
<td>Franklin</td>
<td>Robertson</td>
<td>8766</td>
<td>3839</td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run
(Continuation 6)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>342</td>
<td>HEMPHILL</td>
<td>SARINE</td>
<td>8511</td>
<td>3413</td>
</tr>
<tr>
<td>343</td>
<td>SAN AUGUSTINE</td>
<td>SN AUGUST</td>
<td>8491</td>
<td>3471</td>
</tr>
<tr>
<td>344</td>
<td>COLORSPRING</td>
<td>SN JACINT</td>
<td>8784</td>
<td>3567</td>
</tr>
<tr>
<td>345</td>
<td>SAN SABA</td>
<td>SAN SABA</td>
<td>8886</td>
<td>4242</td>
</tr>
<tr>
<td>346</td>
<td>MATHIS</td>
<td>SAN PTOCO</td>
<td>9448</td>
<td>3440</td>
</tr>
<tr>
<td>347</td>
<td>CENTER</td>
<td>SHELBY</td>
<td>9443</td>
<td>3505</td>
</tr>
<tr>
<td>348</td>
<td>RIO GRANDE CTY</td>
<td>STARR</td>
<td>9861</td>
<td>3887</td>
</tr>
<tr>
<td>349</td>
<td>BENGROOK</td>
<td>TARRANT</td>
<td>8499</td>
<td>4140</td>
</tr>
<tr>
<td>350</td>
<td>EVERMAN</td>
<td>TARRANT</td>
<td>8505</td>
<td>4110</td>
</tr>
<tr>
<td>351</td>
<td>RIVERAOKS</td>
<td>TARRANT</td>
<td>8479</td>
<td>4122</td>
</tr>
<tr>
<td>352</td>
<td>GROVETON</td>
<td>TRINITY</td>
<td>8661</td>
<td>3605</td>
</tr>
<tr>
<td>353</td>
<td>WOODVILLE</td>
<td>TYLER</td>
<td>9664</td>
<td>3458</td>
</tr>
<tr>
<td>154</td>
<td>EL CAMPO</td>
<td>WHARTON</td>
<td>9115</td>
<td>3649</td>
</tr>
<tr>
<td>355</td>
<td>WHEELER</td>
<td>WHEELER</td>
<td>8126</td>
<td>4829</td>
</tr>
<tr>
<td>356</td>
<td>IOWA PARK</td>
<td>WICHITA</td>
<td>8527</td>
<td>4445</td>
</tr>
<tr>
<td>357</td>
<td>GEORGETOWN</td>
<td>WILLIAMSO</td>
<td>8927</td>
<td>4014</td>
</tr>
<tr>
<td>358</td>
<td>TAYLOR</td>
<td>WILLIAMSO</td>
<td>8922</td>
<td>3962</td>
</tr>
<tr>
<td>359</td>
<td>QUITMAN</td>
<td>WOOD</td>
<td>8340</td>
<td>3806</td>
</tr>
<tr>
<td>360</td>
<td>PLAINES</td>
<td>YOAKUM</td>
<td>8735</td>
<td>5105</td>
</tr>
<tr>
<td>361</td>
<td>ZAPATA</td>
<td>ZAPATA</td>
<td>9746</td>
<td>4089</td>
</tr>
</tbody>
</table>

1. AZLI MULESHOE PD: 7.35
2. AZKK MORTON SO: 9.00
3. AZKW SPUR PD: 47
4. AZKX FLOYDADA SO: 9.00
5. AZLO POST SO: 106
6. AZLA PLAINVIEW PD: 118
7. AZLR PLAINVIEW SO: 0.00
8. AZLQ LEVELLAND PD: 0.00
9. AZLC LITTLEFIELD PD: 0.00
10. AZKA LITTLEFIELD SO: 183
11. AZTI OLTON PD: 184
12. AZLQ LUR8OK DPS: 193
13. AZLK LUR8OK PD: 193
14. AZLL LUR8OK SO: 193
15. AZLR SLATON PD: 194
16. AZLJ TAHOKA PD: 195
17. AZKS TAHOKA SO: 195
18. AZLF BROWNFIELD PD: 269
19. AZLF BROWNFIELD SO: 269

77-53, Vol. IV
### Table 3-2. Input Data for the Example Run
(Continuation 7)

<table>
<thead>
<tr>
<th>Code</th>
<th>Data</th>
<th>City/Station</th>
<th>Name</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>399</td>
<td>.00</td>
<td>.00</td>
<td>AZLN</td>
<td>.00</td>
<td>.00</td>
<td>295</td>
<td>7.39</td>
<td>20.87</td>
<td>.00</td>
</tr>
<tr>
<td>400</td>
<td>.00</td>
<td>.00</td>
<td>DENVER CITY</td>
<td>295</td>
<td>7.39</td>
<td>20.87</td>
<td>.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>401</td>
<td>.00</td>
<td>.00</td>
<td>MULESHOE S.O.</td>
<td>.00</td>
<td>1.66</td>
<td>5.26</td>
<td>.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>402</td>
<td>.00</td>
<td>.00</td>
<td>NAAG CROSBYTON S.O.</td>
<td>.00</td>
<td>1.66</td>
<td>5.26</td>
<td>.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>403</td>
<td>.00</td>
<td>.00</td>
<td>NAC6 LEVELLAND S.O.</td>
<td>.00</td>
<td>1.66</td>
<td>5.26</td>
<td>.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>404</td>
<td>.00</td>
<td>.00</td>
<td>NAFS PLAINES S.O.</td>
<td>.00</td>
<td>1.66</td>
<td>5.26</td>
<td>.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>405</td>
<td>.00</td>
<td>.00</td>
<td>AAA, JUSTIN SWITCHER</td>
<td>.00</td>
<td>1.66</td>
<td>5.26</td>
<td>.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Notes
- **Code**: Identification code for each station.
- **Data**: Input data associated with each station.
- **City/Station**: City or station name.
- **Name**: Technical name of the station.
- **Value 1** to **Value 6**: Various input values associated with each station.
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>
</tr>
</thead>
<tbody>
<tr>
<td>513</td>
<td>100.</td>
<td>100.</td>
</tr>
<tr>
<td>514</td>
<td>100.</td>
<td>100.</td>
</tr>
<tr>
<td>515</td>
<td>135.</td>
<td>135.</td>
</tr>
<tr>
<td>516</td>
<td>135.</td>
<td>135.</td>
</tr>
<tr>
<td>517</td>
<td>100.</td>
<td>100.</td>
</tr>
<tr>
<td>518</td>
<td>100.</td>
<td>100.</td>
</tr>
<tr>
<td>519</td>
<td>135.</td>
<td>135.</td>
</tr>
<tr>
<td>520</td>
<td>135.</td>
<td>135.</td>
</tr>
<tr>
<td>521</td>
<td>100.</td>
<td>100.</td>
</tr>
<tr>
<td>522</td>
<td>100.</td>
<td>100.</td>
</tr>
<tr>
<td>523</td>
<td>135.</td>
<td>135.</td>
</tr>
<tr>
<td>524</td>
<td>135.</td>
<td>135.</td>
</tr>
<tr>
<td>525</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>526</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>527</td>
<td>10.</td>
<td>10.</td>
</tr>
<tr>
<td>528</td>
<td>10.</td>
<td>10.</td>
</tr>
<tr>
<td>529</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>530</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>531</td>
<td>10.</td>
<td>10.</td>
</tr>
<tr>
<td>532</td>
<td>10.</td>
<td>10.</td>
</tr>
<tr>
<td>533</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>534</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>535</td>
<td>10.</td>
<td>10.</td>
</tr>
<tr>
<td>536</td>
<td>10.</td>
<td>10.</td>
</tr>
<tr>
<td>537</td>
<td>LINEAR FOR 1200 BAUD LINE</td>
<td></td>
</tr>
<tr>
<td>538</td>
<td>900.</td>
<td>900.</td>
</tr>
<tr>
<td>539</td>
<td>1.</td>
<td>3.</td>
</tr>
<tr>
<td>540</td>
<td>900.</td>
<td>900.</td>
</tr>
<tr>
<td>541</td>
<td>1.</td>
<td>3.</td>
</tr>
<tr>
<td>542</td>
<td>900.</td>
<td>900.</td>
</tr>
<tr>
<td>543</td>
<td>1.</td>
<td>6.</td>
</tr>
<tr>
<td>544</td>
<td>LINEAR FOR 2400 BAUD LINE</td>
<td></td>
</tr>
<tr>
<td>545</td>
<td>900.</td>
<td>900.</td>
</tr>
<tr>
<td>546</td>
<td>1.</td>
<td>3.</td>
</tr>
<tr>
<td>547</td>
<td>900.</td>
<td>900.</td>
</tr>
<tr>
<td>548</td>
<td>1.</td>
<td>3.</td>
</tr>
<tr>
<td>549</td>
<td>900.</td>
<td>900.</td>
</tr>
<tr>
<td>550</td>
<td>1.</td>
<td>6.</td>
</tr>
<tr>
<td>551</td>
<td>LINEAR FOR 4800 BAUD LINE</td>
<td></td>
</tr>
<tr>
<td>552</td>
<td>900.</td>
<td>900.</td>
</tr>
<tr>
<td>553</td>
<td>1.</td>
<td>3.</td>
</tr>
<tr>
<td>554</td>
<td>900.</td>
<td>900.</td>
</tr>
<tr>
<td>555</td>
<td>1.</td>
<td>3.</td>
</tr>
<tr>
<td>556</td>
<td>900.</td>
<td>900.</td>
</tr>
<tr>
<td>557</td>
<td>1.</td>
<td>6.</td>
</tr>
<tr>
<td>558</td>
<td>1 2</td>
<td></td>
</tr>
<tr>
<td>559</td>
<td>0 0</td>
<td></td>
</tr>
<tr>
<td>560</td>
<td>3 2 0 0 0 0 8 .008 0</td>
<td></td>
</tr>
<tr>
<td>561</td>
<td>3 2 0 0 0 0 8 .008 0</td>
<td></td>
</tr>
<tr>
<td>562</td>
<td>3 2 0 0 0 0 .050 0</td>
<td></td>
</tr>
<tr>
<td>563</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>564</td>
<td>LDR 35 300 4.26 3.94 0 0 .0 0</td>
<td></td>
</tr>
<tr>
<td>565</td>
<td>TCIC 60 8615.25 14.97 0 0 .0 0</td>
<td></td>
</tr>
<tr>
<td>566</td>
<td>AOM 500 500 3.62 3.62 0 0 .0 0</td>
<td></td>
</tr>
<tr>
<td>567</td>
<td>C-CODE 300 300 .13 8.33 0 0 .0 0</td>
<td></td>
</tr>
<tr>
<td>568</td>
<td>CCH 426 459 5.78 5.78 0 0 .0 0</td>
<td></td>
</tr>
<tr>
<td>569</td>
<td>MVP 50 175 9.34 9.34 0 0 .0 0</td>
<td></td>
</tr>
</tbody>
</table>
Table 3-2. Input Data for the Example Run  
(Continuation 10)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>570</td>
<td>ING/NL</td>
<td>50 200 .45 .45 0 0 .0 .0</td>
</tr>
<tr>
<td>571</td>
<td>ADM/NL</td>
<td>300 300 .45 .0 0 0 .0 .0</td>
</tr>
<tr>
<td>572</td>
<td>NCIC</td>
<td>50 90 .0 9.21 0 0 .0 .0</td>
</tr>
<tr>
<td>573</td>
<td>DB/DCS</td>
<td>90 50 3.49 1.1 0 0 .0 .0</td>
</tr>
<tr>
<td>574</td>
<td>DB/DNT</td>
<td>90 50 .45 .45 0 0 .0 .0</td>
</tr>
<tr>
<td>575</td>
<td></td>
<td></td>
</tr>
<tr>
<td>576</td>
<td></td>
<td>2 AAAA 1 GSWITCHER ASSINGMENT WITH 11:1X:A4:15</td>
</tr>
<tr>
<td>577</td>
<td></td>
<td>3 TERMINATE SWITCHER ASSIGNMENT</td>
</tr>
<tr>
<td>578</td>
<td></td>
<td>2 AAAA STATE CENTER</td>
</tr>
<tr>
<td>579</td>
<td></td>
<td>6.43 TOTAL XASAC &amp; REG. AT THE AUSTIN SWITCHER WITH F8:5 A13</td>
</tr>
<tr>
<td>580</td>
<td></td>
<td>20 7.0 1 TERMIN/LINE/RESP. TIME/ MPROC WITH 13:FS:2:12</td>
</tr>
<tr>
<td>581</td>
<td></td>
<td>1 Q1 FOR PLOT:11:0 AND 0 FOR SKIPPING IT WITH 13</td>
</tr>
</tbody>
</table>

CPU: 787  CTP: .091  SUPS: 17.904

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

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

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

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

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

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

3.4.3 Output

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

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

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

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

<table>
<thead>
<tr>
<th>Term</th>
<th>AZLI</th>
<th>AZKK</th>
<th>AZKM</th>
<th>AZKZ</th>
<th>AZLO</th>
<th>AZLA</th>
<th>AZLD</th>
<th>AZLC</th>
<th>AZKA</th>
<th>AZLT</th>
<th>AZSK</th>
<th>A7LK</th>
<th>A7LL</th>
<th>A7LF</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAFIN</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</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>2.0</td>
<td>1.1</td>
<td>0.0</td>
<td>1.0</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>
<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>
<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>
<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>
<td>0.0</td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run  
(Continuation 1)

<table>
<thead>
<tr>
<th></th>
<th>TOTAL TRAFFIC ORIGINATED FROM CYS. TERMIN. (MTC/SFC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARLI</td>
<td>950  ARK</td>
</tr>
<tr>
<td>ARKA</td>
<td>554  ARH</td>
</tr>
<tr>
<td>ARHI</td>
<td>1.16T  ARH2</td>
</tr>
<tr>
<td>ARKS</td>
<td>1.65  ARK2</td>
</tr>
<tr>
<td>ARAS</td>
<td>224  ARAS</td>
</tr>
</tbody>
</table>

**TOTAL TRAFFIC: 41.00**

<table>
<thead>
<tr>
<th></th>
<th>TOTAL TRAFFIC DESTINATED TO CYS. TERMIN. (MTC/SFC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARLI</td>
<td>2.564  ARK</td>
</tr>
<tr>
<td>ARKA</td>
<td>7.915  ARK2</td>
</tr>
<tr>
<td>ARHI</td>
<td>2.135  ARH2</td>
</tr>
<tr>
<td>ARKS</td>
<td>2.295  ARK2</td>
</tr>
<tr>
<td>ARAS</td>
<td>1.01  ARAS</td>
</tr>
</tbody>
</table>

**TOTAL TRAFFIC: 119.30**

**TOTAL SYSTEM TRAFFIC: 174.30**
Table 3-3. Printer Output from the Example Run (Continuation 2)

| TERM | AZLI | ACLL | AZLK | AZLL | AZLH | AZLI | ACLL | AZLK | AZLL | AZLH | AZLI | ACLL | AZLK | AZLL | AZLH | AZLI | ACLL | AZLK | AZLL | AZLH |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|   1  |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |

1. **CPU UTILIZATION FOR PROCESSOR IS .707**

2. **RCS= AAAA FOR REGION 1**

**INDICES FOR SYS. TERM.:**

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

**REGS: 1 + SYS. TERM.:**

AZLI MULESHOE PD
AZLI POST 50 PD
AZLC LITTLEFIELD PD
AZLK LUBBOCK PD
AZKS TAHOKA PD
NAAS MULESHOE 4.0
AAAA AUSTIN SWITCH

AZPK MORTON 50
AZLA PLAINFIELD PD
AZLA LUBBOCK 50
AZLF BROWNFIELD 50
NAHS CROSSTOWN 5.0
NASC LEVALLAND 5.0
NAFA PLAINES 5.0

**NPCC: AAAA**
Table 3-3. Printer Output from the Example Run
(Continuation 3)

<table>
<thead>
<tr>
<th>SYSTEM TERMIN.</th>
<th>A2L1</th>
<th>A2L2</th>
<th>A2L3</th>
<th>A2L4</th>
<th>A2L5</th>
<th>A2L6</th>
<th>A2L7</th>
<th>A2L8</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. OF LINES AFO.</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>1.004</td>
<td>1.093</td>
<td>1.093</td>
<td>0.902</td>
<td>0.910</td>
<td>0.893</td>
<td>0.893</td>
<td>0.893</td>
</tr>
<tr>
<td>DISTANCE FROM NSC TRAFFIC</td>
<td>330</td>
<td>325</td>
<td>281</td>
<td>233</td>
<td>204</td>
<td>165</td>
<td>126</td>
<td>88</td>
</tr>
<tr>
<td>CPU TO LINE</td>
<td>1.589</td>
<td>1.589</td>
<td>1.589</td>
<td>1.589</td>
<td>1.589</td>
<td>1.589</td>
<td>1.589</td>
<td>1.589</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INST. COSTS</th>
<th><strong>TOTAL</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>0</td>
</tr>
<tr>
<td>MODEM</td>
<td>240</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>2540</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANNUAL RECUH. COST</th>
<th><strong>TOTAL</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>1342</td>
</tr>
<tr>
<td>MODEM</td>
<td>1304</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>2646</td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run (Continuation 4)

<table>
<thead>
<tr>
<th>SYSTEM TERMIN.</th>
<th>NAAL</th>
<th>NAAP</th>
<th>NAAR</th>
<th>NAFA</th>
<th>NAAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. OF LINES Req.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2KB</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>n</td>
</tr>
<tr>
<td>2.4KB</td>
<td>0</td>
<td>0</td>
<td>n</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>4.8KB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>DISTANCE FROM RSC TRAFFIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE TO CPU</td>
<td>.221</td>
<td>.221</td>
<td>.357</td>
<td>.204</td>
<td>.000</td>
</tr>
<tr>
<td>CPU TO LINE</td>
<td>.701</td>
<td>.701</td>
<td>1.132</td>
<td>.648</td>
<td>.000</td>
</tr>
<tr>
<td>LINE RESPONSE TIME</td>
<td>3.273</td>
<td>3.273</td>
<td>3.273</td>
<td>3.273</td>
<td>.000</td>
</tr>
<tr>
<td>INST. COSTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>SER.T.</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>n</td>
</tr>
<tr>
<td>MODERN</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>n</td>
</tr>
<tr>
<td>DROP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>ANNUAL RECEIPT. COST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINES</td>
<td>14364</td>
<td>11160</td>
<td>12780</td>
<td>12096</td>
<td>n</td>
</tr>
<tr>
<td>SER.T.</td>
<td>360</td>
<td>360</td>
<td>360</td>
<td>360</td>
<td>n</td>
</tr>
<tr>
<td>MODERN</td>
<td>526</td>
<td>526</td>
<td>526</td>
<td>526</td>
<td>n</td>
</tr>
<tr>
<td>DROP</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>n</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INST. COST</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>n</td>
</tr>
<tr>
<td>RECUR. COST</td>
<td>15492</td>
<td>12286</td>
<td>13568</td>
<td>14124</td>
<td>n</td>
</tr>
<tr>
<td>TOTAL COST</td>
<td>259782</td>
<td>259782</td>
<td>259782</td>
<td>259782</td>
<td>n</td>
</tr>
</tbody>
</table>
Table 3-3. Printer Output from the Example Run (Continuation 5)

<table>
<thead>
<tr>
<th>SUBNET NO.</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGINNING NODE</td>
<td>AZIL</td>
<td>AZKS</td>
</tr>
<tr>
<td>NO. OF TERM.</td>
<td>19</td>
<td>6</td>
</tr>
<tr>
<td>NO. OF LINES</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1,2KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4,8KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LINE UTILIZATION</td>
<td>.158</td>
<td>.617</td>
</tr>
<tr>
<td>TOTAL MILEAGE</td>
<td>617</td>
<td>385</td>
</tr>
<tr>
<td>TRAFFIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE TO CPU</td>
<td>55,691</td>
<td>5,305</td>
</tr>
<tr>
<td>CPU TO LINE</td>
<td>100,777</td>
<td>12,617</td>
</tr>
<tr>
<td>LINE RESPONSE TIME</td>
<td>4.038</td>
<td>3.416</td>
</tr>
</tbody>
</table>

**FINAL MULTIDROP NETWORK AND ITS COSTS**

<table>
<thead>
<tr>
<th>INST. COSTS</th>
<th>SUBTOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>0</td>
</tr>
<tr>
<td>SER.T.</td>
<td>280</td>
</tr>
<tr>
<td>MODEN</td>
<td>1300</td>
</tr>
<tr>
<td>DROP</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANNUAL RECURR. COST</th>
<th>SUBTOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES</td>
<td>16472</td>
</tr>
<tr>
<td>SER.T.</td>
<td>4640</td>
</tr>
<tr>
<td>MODEN</td>
<td>6864</td>
</tr>
<tr>
<td>DROP</td>
<td>3120</td>
</tr>
</tbody>
</table>

**TOTAL COST | SUBTOTAL**

| INST. COST | 1560 | 1140 | 480 |
| RECUR. COST | 31066 | 22127 | 8999 |
| TOTAL COST | 32646 | 32646 |
Table 3-3. Printer Output from the Example Run
(Continuation 6)
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
milage. Summarized costs are also provided.

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

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

ASSIGNED NUMBER OF REGIONS
ENTER NR AND STROKE RETURN KEY
TYPE IN NR OF SYS. TERMS DATA BASES AND CITIES WITH FORMAT 416
THERE ARE 25 SYS. TERMS, 4 DATA BASES, 35 CITIES
TYPE IN DATA BASE LOCATIONS WITH FORMAT 61X144
4 DATA BASES ARE AT AAAA 0000 0000 0000
TYPE IN CITY 'XH WITH FORMAT 33X152X15
TYPE IN PIP NO., NAME MAPPING ADH. AND TRAFFIC
WITH FORMAT 14X146X1468X2
TYPE IN NO. OF RATE STRUCTURES UNDER
CONSIDERATION WITH FORMAT 13
TYPE IN RATE APPLICATION TO EACH COMMON
WHICH EACH SYS. TERM, WITH FORMAT 10X2
READ IN TRAFFIC DENSITY TYPE AND RATE STRUCTURE
FOR EACH CITY WITH FORMAT AN11
TYPE IN NO. OF LINE TYPES APPLICABLE WITH FORMAT 13
TYPE IN NAME, CAPACITY, UTIL. FACTOR AVAIL. FOR
EACH LINE TYPE WITH FORMAT ASX14X14X32X211X11
TYPE IN NO. OF DEVICES AND NAMES FOR EACH LINE TYPE
WITH FORMAT 13/101412X1411
TYPE IN INST. AND RECUR. COSTS WHT
RATE STRUCTURE, LINE TYPE, DEVICE TYPE TRAFFIC DENSITY
AND DUEPLEXING MODE WITH FORMAT 2F442/2F442
TYPE IN INST. COSTS "N" INES WHT
RATES LINE, DENSITY ON DUEPLEXING NODE
WITH FORMAT 4F2
TYPE IN INST. FOR LINEARITY OF LINE RECUR. COST
FUNCTION WITH 1 LINEAR AND NONLINEAR OTHERWISE
WITH FORMAT 11 FOR EACH LINE TYPE
TYPE IN RECUR. COSTS WITH FORMAT 4F42 IF LINEAR
WITH FORMAT 10F442/10F442 IF NONLINEAR
IF NONLINEAR USE INE82
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 NEEDED
WITH FORMAT 2I2 AND END IT WITH A 0 0
TYPE IN NPL, MAX, MIN, HAKOH, NODE
TANH TD IN FORMAT (514)F27,5
TYPE IN NO. OF MSG TYPES AND TRAFFIC STATISTICS
SUCH AS MSGAM, MSLE, MSGPT, RATIO WITH
FORMAT 14/10A4,2219X2F4X3
TYPE IN PRELOADED SYSTEM TERMIN. AND NSC WITH
FORMAT 111X144AN
32X3 DISTANCE TERMS ARE OVERSIZED
ASSUME A SYSTEM CENTERED
ENTER CODE FOR HSIC AND STROKE RETURN KEY
INPUT TOTAL NO. OF TRANSACTIONS AND NO. OF ACCESS AT THE SWITCHER
ENTER WITH 04 AND 13 UNDER X5C5/5C
READ IN LIMITS ON NO. OF SYS. TERMS ON A I, 1
RESPONSE TIME READ AND NO. OF PROCESORS WITH FORMAT
13X5D2X12
IF PLOTTING IS REQUIRED TYPE 1 WITH FORMAT 13
TRYLKNK HAS BEEN ACCESSUED FOR 10052 TIMES
UPHST HAS BEEN ACCESSUED 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-url)
REFERENCES


APPENDIX A

STACOM PROGRAM LISTING

5192A*STACOM(1).MAIN/0777

C%******************************************************************************
C% STACOM PROGRAM LISTING
C%******************************************************************************
C% STACOM TOPOLOGY PROGRAM
C% JET PROPULSION LABORATORY
C% 4000 OAK GROVE DRIVE
C% PASADENA, CALIFORNIA 91105
C%******************************************************************************
C% THIS PROGRAM IS DESIGNED TO PERFOM FORMATIONS OF REGIONS, SELECTIONS
C% OF REGIONAL SWITCHING CENTERS, FORMATIONS OF INITIAL REGIONAL NETWORKS,
C% OPTIMIZATION OF REGIONAL NETWORKS USING THE FGA-WILLIAMS METHOD IF
C% REQUESTED, AND FINALLY FORMATION OF AN INTERREGION NETWORK AND ITS
C% OPTIMIZATION
C%******************************************************************************
C%******************************************************************************
C% THE TOPOLOGY 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: RNRT (REGIONAL NETWORK TERMINATION AND ITS OPTIMIZATION)
C% SUBPROGRAM-2: IRNRT (INTER-REGION NETWORK OPTIMIZATION)
C% SUBPROGRAM-3: ICOST (COSTING FUNCTION)
C% SUBPROGRAM-4: IHFUT (LINE UTILIZATION FUNCTION)
C% SUBPROGRAM-5: ILNCF (LINE CONFIGURATION BASED ON TRAFFIC)
C% SUBPROGRAM-6: IP (STORING OR RETRIEVAL DISTANCE DATA)
C% SUBPROGRAM-7: IDIST (FINDING DISTANCE BETWEEN TWO GIVEN TERMINALS)
C% SUBPROGRAM-8: IFTX (FINDING COMPRESSED INNX FOR DIST)
C% SUBPROGRAM-9: IREV (RECOVERING COMPRESSED DISTANCE DATA)
C% SUBPROGRAM-10: IPLOT (PLOTTING EACH DROP ON A MULTIDROP NETWORK)
C% SUBPROGRAM-11: IRPNSE (ESTIMATING RESPONSE TIME)
C%******************************************************************************
C%******************************************************************************
C%******************************************************************************
C% PARAMETER NP2=I30, NP2=1, NP3=4, NP4=9
C% PARAMETER NP6=NP3-2+NP2+14+1
C% PARAMETER NP7=NP3+10+NPC
C% COMMON X/E/N, SYR(NP1), NRSC(MW), NUMR(MW), TRAFD(NP1),
C% TFRAT(NP1)
C% /VH/ IVERT(NP1), THORZ(NP1)
C% */CON/S N2+3&M, N7&M, N17&M, N7&M
C% */INN/ IREAD(NP2, NP3), TREAD(NP3, NP4), IFLAG(NP2, NP3)
C% */INP/ AINRT(NP2, NP3, NP4), RNRT(NP3, NP4), RNPCT(NP2, NP3, NP4)
C% */AMNTH(NP2, NP3, NP4), RECP(NP2, NP3, NP4), RNPCT(NP2, NP3, NP4)
C% */LINC/ LINMIX(NP3), LINCAP(NP3), UTIL17(NP3)
C% */REF/ IREF(NP3), TRAFD(NP1, NP2, NP3), DSTNCE(NP6), MAPOR(NP1)
C% */VFR/ IFRD(NP2, NP3, NP4), IVER1
C% */NAME/ NAMEST(NP1), LNAME(NP3), NAMFHR(NP3)
C% */SUM/ ASUM1, ASUM
C% */XMT/ TIMMT(7, NP3), WATT(6)
C% */WSLA/ AMSL(Q)
C% */RUN/ NTERMS, FILE, PROG, NP2, NP4,
C% */ADV/ IADG(NP1), KCHG+KADD, OKCHG=FIRST DROP, KADD=JUST FOR LINE
C% INTEGER DSTNCE
C% DIMENSION IACT(NP2, NP3), INDPT(NP1)
C% DIMENSION NUMR(NP2), TRAFD(2), HRAF(NP7)

A-1
DATA ITRAFC/ITRAFINTROUT/
DIMENSION TRM(W,W), DRM(W,W), NUMR(NP), NUMR(NP,4)
INTEGER SRV
DIMENSION OUTPR(NP)
NMAX=NP, QMAX=MAX SIZE FOR OVERFLOW DISTANCE DATA TABLE
C
C SELECT NUMBER OF REGIONS
C
25 WRITE(6,220)
READ(5,N2) NR1
WRITE(6,1011) NR1
ANR1=NP1
C READ IN TRAFFIC DENSITY INDEX AND RATE STRUCTURE FOR EACH SYSTEM
C TERMINATION IN THE SYSTEM
C
CALL CREADA(N1)
C
C READ IN RATE APPLICATION MATRIX
C
CALL CREADM(N2)
C
C READ IN NAMES, CAPACITIES, UTILIZATION FACTORS AND AVAILABILITIES
C FOR LINES APPLICABLE IN THE SYSTEM
C
CALL CREADD(N3)
C
C READ IN INSTALLATION AND RECURRING COSTS FOR CHARGEABLE ITEMS
C REQUIRED FOR COMMUNICATION LINES
C
CALL CREADD(N4)
C
C READ IN INSTALLATION AND RECURRING COSTS FOR LINES
C
CALL CREADL
C
C READ IN ACTIONS TO BE PERFORMED ON EACH REGIONAL NETWORK
C 1ST ELEMENT : INSERTIONS TO PRELOADED REGIONS ARE ALLOWED
C 2ND ELEMENT : NO SUCH ACTION IS NOT ALLOWED
C 3RD ELEMENT : NETWORK OPTIMIZATION IS TO BE PERFORMED
C 4RD ELEMENT : NO OPTIMIZATION IS NEEDED
C
CALL CREADK
C
C READ IN LINE AND LINE PROTOCOL CHARACTERISTICS
C
CALL CREADR
C
C CONVERT TRAFFIC FROM CHARACTERS/MIN TO RITS/SFC
C
DO 85 K=1,2
DO 85 I=1,N1
DO 85 L=1,N7
CONTINUE
ISUM=0
77-53, Vol. IV

DO 25 I=1,NCITY
   ISUM=ISUM+1
   TRAFF(I)=ISUM
25 CONTINUE
DO 70 I=1,NI
   NUMPR(I) = 0  NO. OF SYSTEM TERMINATIONS AT EACH REGION
70 CONTINUE
WRITE(6,888)
READ(5,801) NCODE, NSTATE, NREGO
WRITE(4,804) NCODE, NSTATE, NREGO
NSTATE=LOCAL(NSTATE) FIRST CARDINAL INDEX
GO TO (RO1,RO2*240) NCODE
CONTINUE
WRITE(NSTATE) = NREGO
NUMPR(NREGO) = NUMPR(NREGO) + 1
CONTINUE
NRSC(NREGO) = NSTATE
GO TO R05
CONTINUE
DO 70 L=1,N1
   TRAFO(NL)=0.
70 CONTINUE
I0VERI=1  BCOUNTER FOR OVERSIZED TRAFFIC DATA
C CALCULATE DISTANCE DATA BETWEEN SYSTEM TERMINATIONS.
C
DO 20 J=1,NCITY
   DO 30 K=1,NCITY
      IF(J+K) S1*,30*30
30 CONTINUE
151 IS1=(IVER(J)+IVER(K))**2
152 ISQ1=IS1*(1+DRZ(J)+DRZ(K))**2
153 IF(ISQ1 EQ. 0) GO TO 29
154 SQ1=ISQ1/10.
155 N5Q1=INT(SQ1)
156 DIFF=S01-NSQ1
157 IF(DIFF .GT. 0) SQ1=NSQ1+1.
158 BDIST=SORT(SQ1)
159 KDIST=INT(BDIST)
160 DIFF=INT-KDIST
161 IF(DIFF .GT. 0) KDIST=KDIST+1
162 GOTO 29
22 CONTINUE
157 KDIST=0
23 CONTINUE
JKL=LINK(J,K)
164 IF(KDIST .LE. 510) GOTO 5
165 CALL OVERFL(JKL,KDIST)
166 GOTO 30
CONTINUE
164 CALL PACK(JKL,KDIST,1,DESTCF)
165 CONTINUE
166 CONTINUE
167 I0VERI=I0VERI-1
168 CONTINUE
WRITE(6,3) I0VERI
C TOTAL INPUT TRAFFIC BY EACH SY. TERM.
C TRFALL=0.0
175 TALLIT=0.
176 TALLDN=0.
177 DO 41 L=1,N1
178 TRAFIT(L)= 0.0
179 TRAFDN(L)= 0.0
180 DO 42 J=1,N7
181 TRAFIT(L)= TRAFIT(L) + TRAFDL+P+J
182 TRAFDN(L)= TRAFDN(L) + TRAFDL+1+J
183 CONTINUE
184 TALLDN=TALLDN+TRAFDN(L)
185 TALLIT=TALLIT+TRAFIT(L)
186 CONTINUE
187 C PRINT OUT TRAFFIC DATA BETWEEN SYSTEM TERMINATIONS
188 C
189 NTURN=N1/15 + 1
190 NRFM=MOD(N1,15)
191 IF(NREM .EQ. 0) NTURN=NTURN-1
192 WRITE(IWT+11)
193 DO 110 KK=1,1 FOR TEST ONLY
194 KK1=KK-15 + 1
195 KK2=KK*15
196 IF(KK2 .GT. N1) KK2=N1
197 WRITE(IWT+13) (INDXPT(J), J=KK, KK+1)
198 DO 99 J=1,N7
199 KK=1,2
200 DO 28 KK=KK1, KK2
201 WRITE(IWT+110) (TRAFC(KT), KT=KK1, KK2)
202 CONTINUE
203 WRITE(IWT+140) (INDXPT(NJ), NJ=1,N1)
204 WRITE(IWT+74) TALLDN
205 CONTINUE
206 C PRINT OUT TRAFFIC ORIGINATED FROM EACH SYSTEM TERMINATION
207 C
208 WRITE(IWT+1013)
209 WRITE(IWT+1001) (INDXPT(NJ), TRAFDN(NJ)+NJ=1,N1)
210 WRITE(IWT+74) TALLDN
211 CONTINUE
212 C PRINT OUT TRAFFIC DESTINATED TO EACH SYSTEM TERMINATION
213 C
214 WRITE(IWT+1014)
215 WRITE(IWT+1001) (INDXPT(NJ), TRAFIT(NJ)+NJ=1,N1)
216 WRITE(IWT+74) TALLIT
217 WRITE(IWT+75) TRFALL
218 CONTINUE
219 C PRINT OUT DISTANCE DATA BETWEEN SYSTEM TERMINATIONS
220 C
221 NTURN=N1/15 + 1
222 NRFM=MOD(N1,15)
223 IF(NREM .EQ. 0) NTURN=NTURN-1
224 NTURN=1 FOR SHORT OUTPUT
225 DO 101 KK=1, NTURN
226 CONTINUE
227
77-53, Vol. IV

```
228     KK1=(KK-1)*15 + 1
229     KK2=KK*15
230     IF(KK2 .GT. N1) KK2=N1
231     WRITE(IWT,100) (INDXPT(J),J=KK2,KK1,-1)
232     IF(J.EQ.KK1) KK1=J+1
233     DO 27 KK2=KK2+KK1,-1
234     WRITE(IWT,112) INDXPT(J),(OUTHPT(K),K=KK2,KK1,-1)
235     CONTINUE
236     WRITE(IWT,113) (OUTHPT(K),K=KK2,KK1,-1)
237     CONTINUE
238     WRITE(IWT,101)
239     WRITE(IWT,210)
240     WRITE(IWT,210)
241     400 CONTINUE
242     READ(IWT,734) NSCC1
243     WRITE(IWT,1015) NSCC1
244     NSCC1=LOCAL(NSCC1:
245     IF(NSCC1 .NE. 0) GOTO 403
246     WRITE(IWT,4013)
247     GOTO 4005
248     400 CONTINUE
249     TPR1=TRFALL
250     WRITE(IWT,2101)
251     READ(IWT,2102) XSAC, NREQ4W, ORMESKNDG OF REQUESTS/TRANS AT SWITCH
252     WRITE(IWT,2103)
253     READ(IWT,2104) NTERMS,TIMREQ,MPROC
254     WRITE(IWT,2105)
255     READ(IWT,2104) MNPLOT, QMPLOT=1 IF PLOT IS NEEDED
256     C
257     C PRE-CALCULATE CPU TURNAROUND TIME
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     C
264     TPR2=0
265     DO 77 N=1,N1
266     NK=SVRN(N)
267     IF(NK .EQ. 0) GOTO 77
268     IF(IACTN(NK).LT.0) GOTO 77
269     TPR2=TRAFFH(N)+TRAFFI(N)+TPR?
270     CONTINUE
271     DO 76 L=1,NR1
272     IF(IACTN(L).LT.0) ANR1=ANR1
273     CONTINUE
274     77 CONTINUE
275     TPR1=TPR1-TPR2
276     IF(NR1 .LT. N1) GOTO 76
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
285 TPR1 = TPR1 + TPS
286 CONTINUE
287 TPR1 = TPR1 * (1.0 - ZETA1)
288 DO 909 NREG = 1, NR1
289 TRFS = 0.0
290 AMAX = 0.0
291 II = 0
292 IF (NUMPR(NREG) .NE. 0) GOTO 5000 QNREG IS A PRELOADED REGION
293 C
294 C ASSIGN SYSTEM TERMINATIONS TO A REGION WITHOUT ANY PRELOADING
295 C
296 DO 400 NI = 1, NI
297 IF (SVR(NI) .NE. 0) GOTO 400 QN IS PRELOADED
298 IF (ADIST = 0) GOTO 400
299 IF (AMAX = 0) GOTO 400
300 AMAX = ADIST
301 II = NI QPUPDATE FARTHEST SYS. TRAFFIC.
302 CONTINUE
303 NS1 = II QTHE FARTHEST SYSTEM TERMINATION
304 TRFS = TRFS + TRAFDN(NS1) + TRAFIT(NS1)
305 SVR(NS1) = NNI
306 NUMPR(NREG) = NUMPR(NREG) + 1
307 IF (TRFS .GT. TPR1) GOTO 707
308 GOTO 7021
309 CONTINUE
310 IF (IACTN(NREG) .NE. 1) GOTO 909 QINSERTS ARE NOT ALLOWED
311 C
312 C SUM UP TRAFFIC IN THIS REGION
313 C
314 DO 702II = 1, NI
315 IF (SVR(I) .NE. NREG) GOTO 702
316 TRFS = TRFS + TRAFDN(I) + TRAFIT(I)
317 IF (ADIST .GT. AMAX) IF = 1
318 CONTINUE
319 IF (TRFS .GT. TPR1) GOTO 707 QENOUGH TRAFFIC IN THIS REGION
320 IF (NS1 = II) QTHE FARTHEST SYS. TRAFFIC. IN THE REGION
321 NS1 = II QREG. IN NREG.
322 CONTINUE
323 CALL FINDP(NS1, NS2)
324 IF (NS2 .EQ. 0) GOTO 909
325 SFN(NS2) = NREG
326 NUMPR(NREG) = NUMPR(NREG) + 1
327 TRFS = TRFS + TRAFDN(NS2) + TRAFIT(NS2)
328 IF (NREG .EQ. NR1) GOTO 7021
329 IF (TRFS .GT. TPR1) GOTO 707
330 GOTO 7021
331 CONTINUE
332 IF (TRFS .GT. TPR1) GOTO 707 QUPDATE REMAINING TRAFFIC
333 ANR1 = ANR1 - 1
334 IF (ANR1 .LE. 1) GOTO 726
335 CALL UPTR1(ANR1 - 1, TPS) QUPDATE AVERAGE TRAFFIC DPF REGIONS
336 TPR1 = TPR1 * (1.0 - ZETA1) QUPDATE LOWER LIMI
337 CONTINUE
338 GOTO 703
339 CONTINUE
340 C
341 C ONE REGION CASE
342 C
343 DO 727 NN=1,N1
344 SVR(NN) = 1
345 727 CONTINUE
346 NUMR(I) = N1
347 703 CONTINUE
348 C
349 C SELECT REGIONAL SWITCHING CENTER
350 C
351 DO 500 J=1,NN1
352 WCASE = 1.0E12
353 NUMR = 0
354 DO 505 K=1,N1
355 IF(SVR(K) .NE. J) GO TO 505
356 NUMR = NUMR + 1
357 NR(K) = NUMR
358 NUMR(NMRR)=INXPT(K)
359 DO 490 I=1,4
360 WNAME(K)=NAMEST(K,I)
361 490 CONTINUE
362 505 CONTINUE
363 C
364 C PRINT OUT PID AND NAMES FOR SYSTEM TERMINATIONS IN THE REGION J
365 C
366 WRITE(IWT,101A) J,(NUMR(I),(NUMR(I,I),I=1,4),I=1,NUMR)
367 C
368 C PRINT OUT INDICES OF SYSTEM TERMINATIONS IN THE REGION J
369 C
370 WRITE(IWT,102B) (NUMR(I),I=1,NUMR)
371 C
372 IF(NRSC(J) .NE. 0) GO TO 501 QPPE-SELECTED
373 DO 520 K=1,NMRR
374 NN1 = NUMR(K) ASSUMED RSC
375 SUMT = 0.0
376 DO 530 L=1,NMRR
377 NN2 = NUMR(L)
378 SUMT=SUMT+(TRAFFN(HH2)*TRAFFT(HH2)*DIST(HH2,NN1))
379 530 CONTINUE
380 IF(SUMT .GT. WCASE) GO TO 520
381 WCASE = SUMT
382 NRSC(J) = NN1
383 520 CONTINUE
384 C
385 IF(NN1 .LE. J) GOTO 551
386 DO 902 K1=1,N7
387 KKK=NAMEST(K1)
388 902 CONTINUE
389 500 CONTINUE
390 C
391 C GENERATE INTER-REGION ORIGIN-DESTINATION MATRIX
392 C
393 C INITIALIZATION
394 C
395 IF(NN1 .LE. J) GOTO 551
396 DO 902 K1=1,N7
397 KKK=NAMEST(K1)
398 902 CONTINUE
77-53, Vol. IV

399 IF(KKK;EQ. 0) WRITE(6, 7777) K1
400 NBASM(K1)=3VR(KKK)
401 CONTINUE
402 DO 609 K1=1, NR1
403 DO 609 K2=1, NR2
404 TRM(K1,K2)=0
405 CONTINUE
406 CONTINUE
407 DO 900 J=1, NR1
408 NMP = 0
409 DO 905 K=1, NR1
410 IF(SVR(K) .NE. J) GO TO 905
411 DO 915 KK=1, N7
412 MN2=NBASE(KK) ORIGTAL. INHX FOR KKS DATA PAGE
413 TRM(J,MM2)=TRAFF(J,KK)+TRMF(J,MM2) QUITGOING TRAFFIC
414 TRM(NN2,J)=TRAFF(J,KK)+TRMF(NN2,J) QUITCOMING TRAFFIC
415 CONTINUE
416 CONTINUE
417 DO 920 J=1, NR1
418 NMP = NRSC(J1)
419 ORN(J, J1) = DIST(NN1, NN2)
420 CONTINUE
421 CONTINUE
422 CAST=NR1/10+1
423 DO 535 L=1, CAST
424 LL=LL+10
425 LL1=L10
426 IF(L1,JT,NR1) LUN=NR1
427 WRITE(IWT;1030) NR1, NR1, (K, K=LL, LL1)
428 DO 1024 I=1, NR1
429 WRITE(IWT;1021) I, (TRM(I,J), J=LL, LL1)
430 CONTINUE
431 CONTINUE
432 DO 535 L=1, CAST
433 LL=LL+10
434 LL1=L10
435 IF(L1,JU, NR1) LU=NR1
436 WRITE(IWT;1031) NR1, NR1, (K, K=LL, LL1)
437 DO 1024 I=1, NR1
438 WRITE(IWT;1021) I, (TRM(I,J), J=LL, LL1)
439 CONTINUE
440 CONTINUE
441 CALL IRNOPSNR1, NLIMIT, TRM)
442 FORMAT(/'40X; TOTAL TRAFFIC='; F9.2)
443 FORMAT(/'40X; TOTAL SYSTEM TRAFFIC='; F9.2)
444 220 FORMAT('A-SJUSSUME NUMBER OF REGIONS;'
445 * / ENTER NR AND STRIKE RETURN KEY;
446 735 FORMAT(I3)
447 88A FORMAT('I1; TYPE IN PRELOADED SYSTEM TRANN, AND PSC WITH,;
448 800 FORMAT(I10X, I1; TYPE IN A SYSTEM CENTROID;
449 804 FORMAT('I10X, I1; DISTANCE ITEMS ARE OVR,
450 3 FORMAT('I10X, I1; ENTER CONF FOR PSC AND STRIKE RETURN KEY;
451 210 FORMAT('I1; ENTER CONF FOR PSC AND STRIKE RETURN KEY;
452 734 FORMAT(A)
453 4013 FORMAT('A; THE GIVEN SYSTEM COMM. CENTROID IS NOT OK, RETYPE IT!)

A-8
456 2101 FORMAT(1X,'*INPUT TOTAL NO. OF TRANSACTIONS AND NO. OF ACCESS *')
457 2102 FORMAT(1X,'*AT THE SWITCHER*/1X.ENTER WITH FA*5,13 UNDER XSEC/SEC*/1X')
458 2103 FORMAT(1X,'*READ IN LIMITS ON NO. OF SYS. TERMS. ON A INF**/')
459 2104 FORMAT(1X,'*RESPONSE TIME REGN AND NO. OF PROCESSORS WITH FORMAT *')
460 2105 FORMAT(1X,'*')
461 2106 FORMAT(13,F5.2,I2)
462 2107 FORMAT(1X,'*AT THE SWITCHER*/1X.ENTER WITH FA*5,13')
463 2108 FORMAT(1X,'*IF PLOTTING IS REQUIRED: TYPE 1 WITH FORMAT 1X**')
464 2109 FORMAT(1X,'*')
465 2110 FORMAT(1X,'*')
466 2111 FORMAT(1X,'*')
467 2112 FORMAT(1X,'*')
468 2113 FORMAT(1X,'*')
469 2114 FORMAT(1X,'*')
470 2115 FORMAT(1X,'*')
471 2116 FORMAT(1X,'*')
472 2117 FORMAT(1X,'*')
473 2118 FORMAT(1X,'*')
474 2119 FORMAT(1X,'*')
475 2120 FORMAT(1X,'*')
476 2121 FORMAT(1X,'*')
477 2122 FORMAT(1X,'*')
478 2123 FORMAT(1X,'*')
479 2124 FORMAT(1X,'*')
480 2125 FORMAT(1X,'*')
481 2126 FORMAT(1X,'*')
482 2127 FORMAT(1X,'*')
483 2128 FORMAT(1X,'*')
484 2129 FORMAT(1X,'*')
485 2130 FORMAT(1X,'*')
486 551 CONTINUE
487 STOP
488 C
489 SUBROUTINE FIMOSD(N,M)
490 C
491 C FIND THE NEXT SYSTEM TERMINATION M WHICH IS CLOSEST TO N
492 C WHERE M HAS NOT BEEN ASSIGNED TO ANY REGION YET
493 C
494 C
495 AMIN=290000.
496 M=0
497 DO 70A K=1,N1
498 IF(SWRT(NK).NE.0) GOTO 70A
499 ADIST=DIST(NK)
500 IF(ADIST+GF. AMIN) GOTO 70A
501 AMIN= ADIST
502 M=K
503 70A CONTINUE
504 RETURN
505 SUBROUTINE CREADD
506 C
507 C READ IN ACTIONS REGARDING INSPECTIONS OF SYSTEM TERMINATIONS
508 C TO PRELOADED REGIONS AND REGIONAL NETWORK OPTIMIZATIONS
509 C
510 C
511 64 FORMAT(* TYPE IN ACTION INDICES FOR EACH REGION *)
512 65 FORMAT(* 1ST ELEMENT: 1= INSERTION TO THIS PRELOADED REGION IF OK */
77-53, Vol. IV

513 • /* AND ELEMENT 1 = OPTIMIZATION IS NEEDED */
514 DO 200 NN1=1,NN1
515 DO 200 NN2=1,2
516 IACT(NN1,NN2)=0
517 200 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 */
521 250 CONTINUE
522 READ(5,201) NREG, NCODE
523 FORMAT(2I2),
524 IF(NREG.EQ.0) GOTO 265
525 IF(NREG.GT.NR1 .OR. NCODE .GT. 2) GOTO 265
526 IACT(NREG,NCODE)=1
527 GOTO 250
528 260 CONTINUE
529 WRITE(6,203)
530 202 FORMAT(* PLEASE RTYPE THE INPUT *)
531 GOTO 250
532 265 CONTINUE AND MORE INPUT
533 RETURN
534 540 CONTINUE
535 SUBROUTINE CRFADA(N1)
536 C ***************
537 C
538 C FUNCTIONS OF THIS SUBROUTINE ARE TO
539 C 1. RECEIVE TOTAL NO. OF SYSTEM TERMINATIONS, DATA BASES AND CITIES
540 C 2. RECEIVE CITY LOCATIONS (V & H)
541 C 3. RECEIVE PID NO., SYS. TFRM, NAMES, AND MAPPING AND TRAFFICS
542 C
543 C ***************
544 81 FORMAT(* TYPE IN NO. OF SYS. TFRMS, DATA BASES AND CITIES *)
545 * "WITH FORMAT 315"
546 READ (5,10) NI,N7CITY, N NUMBER OF SYSTEM TERMINATIONS
547 WRITE(6,10) NI,N7CITY
548 78 FORMAT(* THERE ARE 'I', 'SYS. TFRM', 'T', 'DATA BASES', 'T',)'
549 * CITIFS', '/', 'TYPF IN DATA BASE LOCATIONS WITH FORMAT 6(IV,A4)')
550 READ(5,15) (BASE(I),I=1,N7)
551 15 FORMAT(6(I,A4))
552 WRITE(6,16) N7, (BASE(I),I=1,N7)
553 16 FORMAT(15,I1) DATA BASES ARE AT (V,A4)
554 WRITE(6,16)
555 161 FORMAT(* TYPE IN CITY V-H WITH FORMAT (VH15,2X,15))
556 READ(5,17) (VERT(I),HORIZONTAL(I),I=1,N7)
557 17 FORMAT(33X,15,2X,15))
558 WRITE(6,17)
559 76 FORMAT(* TYPE IN PID NO., NAME, MAPPING AND TRAFFIC *)
560 * "WITH FORMAT 14,1X,5A1,46,5A1,2")
561 READ(5,2011) (NAMEST(I,J),J=1,N1),IAND(I),MAPAND(I), (TRAF(I,K,L))
562 • K=1+2+L=1+7)
563 INXPT(I)=I
564 79 CONTINUE
565 70 FORMAT(A4,1X,5A1,46,5A1,2)
567 10 FORMAT(315)
568 RETURN
569 500 FORMAT(A4,1X,5A1,46,5A1,2)
566 40 RETURN
565 CONTINUE
564 90 FORMAT(A4,1X,5A1,46,5A1,2)
567 315 RETURN
566 SUBROUTINE CRFADA(N2)
C
C CREATE A RATE APPLICATION MATRIX IRATEJIN? N2)
C
C
WRITE(6,83)
83 FORMAT(' TYPE IN NO. OF RATE STRUCTURES UNDER*,
* / * CONSIDERATION WITH FORMAT I3*
READ (5,50) N2
WRITE(6,84)
84 FORMAT(' TYPE IN RATE APPLICATION TO EACH COMM. * *
* / WRT EACH SYS. TFRM. WITH FORMAT 1012*)
DO 11 ITRATE=1,N2
READ (5,100) (IRATEJ,J=1,N2)
11 CONTINUE
WRITE(6,71)
71 FORMAT(' READ IN TRAFFIC DENSITY TYPE AND RATE STRUCTURE*,
* / FOR EACH CITY WITH FORMAT 1011)
500 READ(5,72) ((TRANP(I,J),J=1,2),I=1,NCITY)
501 FORMAT((10I2))
50 FORMAT (13)
100 FORMAT (1012)
RETURN
SUBROUTINE CREADD(N3)
C
C READ IN NAMES, UTILIZATION FACTORS AND CAPACITY FIGURFS
C FOR LINES TO BE USED IN THE SYSTEM
C
C
WRITE(6,85)
85 FORMAT(' TYPE IN NO. OF LINE TYPES APPLICABLE WITH FORMAT I3*)
READ (5,50) N3
WRITE(6,86)
86 FORMAT(' TYPE IN NAME, CAPACITY, UTIL. FACTOR AVAILABLE FOR *
* / FOR EACH LINE TYPE WITH FORMAT A6*1X6*1X*F3,2*P1X1*, I*)
DO 12 I=1,N3
10 READ(5,100)LINAME(I),LINCAP(I),UTILF(I),LIMIX(I),TDUPLY(I)
12 CONTINUE
WRITE(6,71)
71 FORMAT(' TYPE IN NAME, UTILIZATION FACTORS AND CAPACITY FIGURFS
* / FOR EACH LINE TYPE WITH FORMAT A6*1X6*1X*F3,2*P1X1*, I*)
READ(5,50) N4
RETURN
SUBROUTINE CRFADD(N4)
C
C CREATE A MATRIX OF BASIC INSTALLATION AND RECURRING COSTS
C FOR CHARGEABLE ITEMS; ASSUMING COST IS A LINEAR FUNCTION
C
C
WRITE(6,87)
87 FORMAT(' TYPE IN NO. OF DEVICES AND NAMES FOR EACH LINE TYPE*,
* / WITH FORMAT I3/10(A6*1X*)
READ (5,50) N4, (NAMEH(I),I=1, N4)
WRITE(6,88)
88 FORMAT(' TYPE IN INST. AND RECUR. COSTS WRT *
* / RATE STRUCTURE, LINE TYPE, DEVICE, TRAFFIC DENSITY *
* / AND MIPPLYING MODE WITH FORMAT 2F9,2/2F9,2*)

A-11
DO 13 IRATE=1,N2
DO 13 ILINE=1,N3
DO 13 TDNSTY=1,3
READ(5,100)((ANSTC(IRATE,ILINE,TDNSTY,J),J=1,2),I=1,9)
CONTINUE
READ (5,101)((RECRL(IRATE,ILINE,TDNSTY,J),J=1,2),I=1,9)
CONTINUE
13 CONTINUE
100 FORMAT (2F9.2)
50 FORMAT (13,10(A6,1X))
RETURN
SUBROUTINE CREATE
***********************************************************************
C CREATE A MATRIX OF BASIC INSTALLATION AND RECURRING COSTS FOR
C LINES. COST MAY OR MAY NOT BE A LINEAR FUNCTION OF DISTANCE
C
C
C***********************************************************************
WRITE(6,89)
89 FORMAT(' TYPE IN INST. COSTS FOR LINES W/T:
1 /* RATE LINE, DENSITY, AND NUREXING MOD: 
2 /* WITH FORMAT 4F9,2')
WRITE(6,90)
90 FORMAT(' TYPE IN INDEX FOR LINEARITY OF LINE RECUR. COST: 
1 /* FUNCTION WITH 1=LINEAR AND NONLINEAR OTHERWISE:
2 /* WITH FORMAT 11 FOR EACH LINE TYPE:
WRITE(6,91)
91 FORMAT(' TYPE IN RECUR. COSTS WITH FORMAT 4F9.2 IF LINEAR:
1 /* WITH FORMAT 10F9.3/10F9,3 IF NONLINEAR:
2 /* IF NONLINEAR, USF 10F9.2')
DO 14 IRATE=1,N2
DO 14 ILINE=1,N3
READ(5,200) INDEX
IFLAG(IRATE,ILINE)=INDEX GLINE COST LINEARITY INDICATOR
READ (5,100) ((ANSTC(IRATE,ILINE,TDNSTY,J),J=1,2),I=1,9)
IF (INDEX,ANF.1) GO TO 3
C LINEAR COST FUNCTION
READ (5,101) ((RECRL(IRATE,ILINE,TDNSTY,J),J=1,2),I=1,9)
GO TO 14
C CONTINUE A NONLINEAR COST FUNCTION
READ (5,400) ((RECRL(IRATE,ILINE,TDNSTY,J),J=1,16),I=1,9)
CONTINUE
100 FORMAT (4F9.2)
200 FORMAT (11)
401 FORMAT ((10F9.3/10F9,3))
RETURN
FUNCTION LOCAL(NL)
C***********************************************************************
C FIND LOCAL INDEX FOR SYSTEM TERMINATION WITH YD NL
C***********************************************************************
LOCAL=0
IF(NL,FG.0) RETURN
DO 408 NF=1,N1
IF(INOU(TV),FG,NL) GOTO 408
408 CONTINUE
RETURN  
4002 LOCAL=NN  
4003 RETURN
4004 SUBROUTINE OVERFL(J,K)  
4005 C  
4006 C  
4007 C STORE OVERFLOW ELEMENT (J,K) AT LOCATION TGER OF TABLE  
4008 C LA AND PUT A MARK 511 AT LOCATION J OF TABLE TA (DISTANCE)  
4009 C  
4010 C  
4011 IF(IOVER(I-GE..NMAX)) GO TO 4000  
4012 CALL PAK(J,511+1,DISTANCE)  
4013 IVRO(IOVER(I)+2)=J  
4014 IVER(I)=IOVER(I)+1  
4015 RETURN  
4016 CONTINUE  
4017 FORMAT(2X,'THE OVERFLOW TABLE HAS REFIN FULLY LOADED**(  
4018 *'/,2X,'PLEASE INCREASE ITS SIZE**)  
4019 STOP  
4020 SUBROUTINE CREADR  
4021 C  
4022 C RECEIVE DATA FOR RESPONSE TIME CALCULATION  
4023 C  
4024 C &LIN(NPO+2)=INPUT MSG LENGTH AS A FUNCTION OF TYPE AND PRIORITY  
4025 C MLOUT(NPO+2)=OUTPUT MSG LENGTH AS A FUNCTION OF TYPE AND PRIORITY  
4026 C AMSL(7)=AVERAGE MSG LENGTH FOR  
4027 C 1=POLLING 2=NAK RESPONSE 3=INPUT MSG WITH PRIORITY 1  
4028 C 4=INPUT MSG 5=OUTPUT MSG WITH PRIORITY 1  
4029 C 6=OUTPUT MSGS WITH PRIORITY 2 7=ALL MSGS  
4030 C TINMT(7,NP3)=AVERAGE TRANSMISSION TIME FOR ABOVE ITEMS  
4031 C RATPRT(NP3+2)=OUTPUT MSG DISTRIBUTION AND OUT-GOING MSG RATIO BY PRIORITY  
4032 C N+1.1 = PERCENT OF OUTPUT MSG SENT IN WITH PRIORITY 1 IF ITS TYPE IS N  
4033 C N+1.2 = PERCENT OF OUTPUT MSG WHOSE DESTINATION IS OUTSIDE OF  
4034 C RATIO(NP3+2)=INPUT TRAFFIC DISTRIBUTION AS A FUNCTION OF TYPE AND PRIORITY  
4035 C  
4036 C DIMENSION MSLIN(NP3+2),RATIO(NP3+2),RATIO(NP3+2),  
4037 C 1=MLSLIN(NP3+2)+MSGLIN(NP3),  
4038 C 2=NLPL(NP3)+NLPH(NP3),  
4039 C 3=MOH(NP3),TAMDMP(NP3),TAMO(NP3)  
4040 C  
4041 C WRITE(6,771)  
4042 771 FORMAT(10,I5)  
4043 772 FORMAT(2X,'TYPE IN NHL, NAK, NLPLH**, NAKOH**, MOH**,  
4044 C */'',TAMDM,TAMO IN FORMAT (514,2F7.5)')  
4045 C  
4046 C READ(5,771) (NLPL(I),NLPH(I),NMOH(I))  
4047 C  
4048 C * MOH(I),TAMD(M(I),TAMO(I),I=1,N3)  
4049 C  
4050 C WRITE(6,772)  
4051 772 FORMAT(10,1X,'NAK IN NHL,NLPH**, NAKOH**, MOH**,  
4052 C */'',TAMDM,TAMO IN FORMAT (514,2F7.5)')  
4053 C  
4054 C FORMAT(2X,'TYPE IN NO. OF MSG TYPES, AND TRAFFIC STATISTICS**  
4055 C  
4056 A-13
1: /* SUCH AS MSGNAME, MSLIN, MSLOUT, RATIO WITH */
2: /* FORMAT */
3: 4, (A6,2(2I4,' ',F6.3))
4: 5: READ(5,77) NTYP
5: 6: READ(5,179) (MSGNAME(I),MSLIN(I),MSLOUT(I),RATIO(I))
6: 7: * RATIO(I,J)=RATIO(I,J)+1+RATIO(I,J)
7: 8: FORMATT((A6,2(2I4,' ',F6.3))
8: 9: READ(5,81) CHPAVG
10: 11: 81: FORMATT(F7.4)
12: 13: C:
14: 15: C CALCULATE AVERAGE MSG LENGTH
16: 17: C:
18: 19: DO 61 I=1,7
20: 21: AMSL(I)=0.
22: 23: 61: CONTINUE
24: 25: DO 58 J=1,4
26: 27: ASUM(J)=0.
28: 29: 58: CONTINUE
30: 31: DO 62 J=1,NTYP
32: 33: J1=J
34: 35: J2=J+4
36: 37: AMSL(J1)=AMSL(J1)+MSLIN(J)+RATIO(J,J)
38: 39: AMSL(J2)=AMSL(J2)+MSLOUT(J)+RATIO(J,J)
40: 41: ASUM(J1)=ASUM(J1)+RATIO(J,J)
42: 43: 62: CONTINUE
44: 45: 66: CONTINUE
48: 49: 68: BSUM=0.
50: 51: DO 69 J=1,4
52: 53: J1=J
54: 55: J2=J+4
56: 57: AMSL(J1)=AMS(J1)+AMS(J1)
58: 59: BSUM=BSUM+ASUM(J)
60: 61: 67: CONTINUE
62: 63: 68: CONTINUE
64: 65: AMSL(7)=AMS(7)/BSUM OVERALL AVG. MSG LENGTH
66: 67: IF(ASUM(6).EQ.0.) GOTO 69
68: 69: AMSL(6)=AMS(6)/ASUM(6)
70: 71: 68: CONTINUE
72: 73: AMSL(5)=AMS(5)/ASUM(5) AVG. MSG LENGTH FOR PRIO=1
74: 75: AMSL(1)=AMS(1)/ASUM(1) AVG. MSG LENGTH FOR PRIO=2
76: 77: AMSL(3)=AMS(3)/ASUM(3) AVG. MSG LENGTH FOR PRIO=3
78: 79: WRITE(10,105) (AMSL(I),I=1,7)
80: 81: 105: FORMAT(2/5X*AVG. INPUT MSG WITH PRIO 1=,F6.1,* CHARS)
82: 83: 1 /5X*AVG. INPUT MSG
84: 85: 2 /5X*AVG. OUTPUT MSG W/ PRIO 1=,F6.1,* CHARS
86: 87: 3 /5X*AVG. OUTPUT MSG W/ PRIO=2,F6.1,* CHARS
88: 89: 4 /5X*OVERALL AVG. MSG =,F6.1,* CHARS)
90: 91: DO 65 K=1,N3
92: 93: AMSL(1)=AMS(K)
94: 95: AMS(2)=AMS(K)
96: 97: 64: CONTINUE
98: 99: BSUM=ASUM(3)+ASUM(4)
100: 101: RETURN
A-14
798 SUBROUTINE CWAITC
799 C
800 C
801 C PPF=CALCULATE CPU WAIT TIME
802 C
803 C
804 RHOCPU=XSAC*CPUAVG/MPROC
805 WRITE(1000,R50) RHOCPU
806 A50 FORMAT(* CPU UTILIZATION FOR PROCESSOR IS *,F5.3)
807 IF(RHOCPU .LE. .8) GOTO 851
808 WRITE(6,855)
809 A55 FORMAT(* THE CPU IS OVERLOADED, THEREFORE IT IS NO USE TO *,
810 * GO FURTHER.*)
811 STOP
812 851 CONTINUE
813 BETA=RHOCPU
814 IF( MPROC .EQ. 1) GOTO 700
815 RH02=RHOCPU**2
816 BETA=2.*RH02/(1+RHOCPU)
817 IF( MPROC .EQ. 2) GOTO 700
818 RH04=RHOCPU**4
819 BETA=256.*RH04/(24+72.*RHOCPU+96.*RH02+64.*RH04)
820 CONTINUE
821 700 CONTINUE
822 WAIT(4)=CPUAVG*(BETA/(MPROC*(1.-RHOCPU)))+1
823 WAIT(4)=WAIT(4)*NREQSW
824 RETURN
825 END

GPRINT STACOM.RGNNET/0777
51928 STACOM(1), RGMNET/6777
1    SUBROUTINE RGMNET(JREGN, NREGEN, NUMP, 160, NUMRN)
2    ***************
3  C
4  C DEVELOP A REGIONAL MULTIDROP NETWORK, STARTING WITH A STAR
5  C NETWORK AND THEN OPTIMIZE IT BY ESAI-WILLIAMS METHOD. GIVEN
6  C THE FOLLOWING ARGUMENTS:
7  C
8  C JREGN= THE INDEX FOR THE REGION UNDER CONSIDERATION
9  C NREGEN= THE NUMBER OF SYSTEM TERMINATIONS IN REGION JREGN
10  C NUMP= AN ARRAY THAT CONTAINS INDICES FOR ALL SYSTEM
11  C TERMINATIONS IN REGION JREGN
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 NPI=130, NP2=1, NP3=4, NP4=3
18  C  PARAMETER NP5=4
19  C  PARAMETER NWT1=100, MW4=4, NPC=360
20  C  PARAMETER NC6(NPC+NPC/2)=NP6+1/4
21  C  COMMON/CONST/N1,H2,HUX,NU,N7,NCTY
22  C  /REF/IREF(NPC), TRAF(NPI,2), NUMP(NP1), MAPA(NP1)
23  C  /LINCHR/LINMIX(NP1), LINCPA(NP3), LinTim(NP3)
24  C  /INF/IRAF(NP1,2), IRANG(NPC+2), IFLAG(NP2,2)
25  C  /ET/ SRF(NP1), HSRC(NP1), TRAFMN(NP1), TRAFT(NP1)
26  C  /NAME/NAME(NP1,4), LNAME(NP3), NAMEH(NP4)
27  C  /SUM/ ASUM(NP1), ASUM
28  C  /MSLA/ AMSL(7)
29  C  /ROUND/ NTERMS,TIMPE1,MPROC,MPLOT
30  C  /ADD/ IADD (NP1), KCHO3,KAND
31  C  /RESP/ RHOLIN(4), RPITM
32  C  DIMENSION COSTEW(NP1,4), IAPRAY(NP1,5), IAPAY(NP1,2)
33  C  DIMENSION TIMRSP(NP1), TRFSUM(NP1,2), TIMMUT(NP1)
34  C  DIMENSION HLINE(NP1,MP3), LDUMY(NP3), NUMH(RP1)
35  C  DIMENSION IOST(NP1), LINHIN(NPC+2), LINPL(NP3,2)
36  C  DIMENSION ICSTHW(NP1,NP4,2), ICSTLN(NP1,2), ICSTLN(NP1,2)
37  C  DIMENSION LSUB(NP1), LSUB(NP1), LSUB(RP1)
38  C  DIMENSION IBLANK(NP1), JCHAR(2)
39  C  DIMENSION NUMRN(1)
40  C  DIMENSION RHOF(NP1)
41  C  EQUIVALENCE (JCHAR, ICHAR)
42  C  DATA JSLANK/= 1/
43  C  RSPTIM=0.
44  C  IPOINTER, TCFST1, TCOST2, COST, COSTFW
45  C
46  C INITIALIZE COST ARRAY, INDEPENDENT OF IINF TYPE
47  C
48  C DO 399 K1=1+N1
49  C DO 399 K2=1+2
50  C ICSTLN(K1,K3)=0
51  C DO 399 K4=1+N4
52  C ICSTHW(K1,K4,K3)=0
53  C CONTINUE
54  C NN1=HRSN(JREGN) GLOBAL INDEX FOR RSC
55  C
56  C
C FIND THE LOCAL RSC INDEX IN THE REGION ARRAY

DO 98 IND=1,NOREGN
   IF(IND.EQ.NUMR(IND)) GOTO 99
98 CONTINUE
99 CONTINUE

C BUILD A STAR NETWORK

CALL STAREW

C PRINT OUT STAR NETWORK

CALL SUMPTN(NOREGN,1)

C DEVELOP A MULTIDROP NETWORK UTILIZING THE
C ECAU-WILLIAMS ALGORITHM

MAX5AV=0
MAXM=0
MAXL=0
MAXK=0
MAXKI=0
MAXLIN=0
MAXNOL=0
LINNEW=0
RSPMAX=0
RHomax=0
ICHAR=1
ITALLY=0
JTALLY=0
KCHB=1
KAD=1
I0K=0
INTRY=0
CALL E'S5Wil
WRITE(*,933) ITALLY,JTALLY
933 FORMAT(2X,'T1RLNK HAS BEEN ACCESS FD FOR',I9,' TIMES',:)
97 CONTINUE
99 CONTINUE
RETURN

C SUMMARY TIME STAREW

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

C NOREGN=NUMBER OF SYSTEM TERMINATIONS IN THE REGION

C

INTEGR COST

DO 110 K3=1,NOREGN
   DO 111 K4=2,4
      IARRAY(K3,K4)=0
   111 CONTINUE
   CONTINUE
   KK=NMR(K3)
IARRAY(K3,1)=IAND(KK)
IARRAY(K3,5)=IRSC  & LOCAL INDEX FOR RSC
ARRAY(K3,1)=TRAPIN(KK)
ARRAY(K3,2)=TRAPF1(KK)
TIMESP(K3)=0.
100 CONTINUE
IARRAY(IRSC+1)=NOREGN = 1 NO. OF NODES UNDER RSC
NM=1 ASSUMING THE 1ST SUCCESSOR WITH INDEX 1
IF(IRSC.EQ. 1) NM=2 1ST SUCCESSOR IS WITH INDEX 2
IARRAY(IRSC+2)=NM
IARRAY(IRSC+5)=0
C
C RELATE ALL OF RSC'S SUCCESSORS
DO 200 K5=1, NOREGN
IF(K5.EQ. IRSC) GOTO 200
NM=NM + 1
IF(NM.EQ. IRSC) NM=NM + 1
C
C DO 250 NODE=1, NOAF
IF(NODE.EQ. IRSC) GOTO 255
TRFIN=ARRAY(NODE+1)+0.5
TRfout=ARRAY(NODE+2)+0.5
NN2=NUMR(NODE)
DIST=DIST(N1,N2)
OST(NODE)=DIST
C
C TAKE A FIRST GUES FOR LINE CONFIGURATION
C
C COST=0
RH0=0.
MDROP=IARRAY(NOAF+1)+1
CALL LINNUM(TRFIN, TRfout, LDIMMY, LINOLD, .R, RH0)
C
C COMPUTE INITIAL RESPONSE TIME
C
C IKONT=0
DO 783 I=1,N3
IF(LINUM(1).NE. 0) IKONT=IKONT+1
C
C DO 783 CONTINUE
AINTRF=TRFIN
OUTTRF=TRFOUT
IF(IKONT.EQ. 1 .AND. LDIMMY(LINOLD).EQ. 1) GOTO 772
C
C RESPONSE TIME CALCULATION NEEDS MODIFICATION
C
ACAP=0.
DO 771 NL=1,N3
ACAP=ACAP+LINCAP(NL)*LDIMMY(NL)
C
A-18
LUMMY(NL) = 0
CONTINUE
LUMMY(LINOLD) = 1
AINTRF = TRFIN + LINCAP(LINOLD) / ACAP OPFRCE+T TRAFFIC
OUTRF = TRFOUT + LINCAP(LINOLD) / ACAP
CONTINUE
CALL RSPNSE(AINTRF, OUTRF, LINOLD, NODE, IOK)
IF (I OK) EQ. 1) GOTO 773
IF (LINOLD :EQ. 3) GOTO 774
LUMMY(LINOLD) = 0
LINOLD = LINOLD + 1
GOTO 777
CONTINUE
NLL = 0
N33 = 3
DO 776 I = 1, N33
IF (LUMMY(I) .EQ. 0) GOTO 776
NLL = I
GOTO 780
CONTINUE
LUMMY(I) = 1
GOTO 775
CONTINUE
LUMMY(NLL) = 0
LUMMY(NLL + 1) = LUMMY(NLL + 1) + 1
CONTINUE
CALL RHOFUN(TRFIN, TRFOUT, LUMMY, LINOLD, PHALT, RHO)
GOTO 781
CONTINUE
TIMRESP(NODE) = RSPTIM
KCHFE = 2
CALL ISIMPTR(SRC, NODE, COST)
RHOF(NODE) = RHO
COSTFW(NODE) = COST
COSTFW(NODE + 1) = LINOLD
DO 499 NL = 1, N3
NLINES(NODE, NL) = LUMMY(NL)
DO 499 NM = 1, 2
ICSTLN(NODE, NM) = ICSTLN(NODE, NM) + LNKCHW(NL, NK + NM)
DO 499 NK = 1, N4
ICSTHW(NODE, NK + NM) = LNKCHW(NL, NK + NM) + ICSTHW(NODE, NK + NM)
CONTINUE
JTRAF = TRFIN + TRFOUT
JTRAF = JTRAF / UTILIZ(LINOLD)
COSTFW(NODE + 3) = JTRAF / LINCAP(LINOLD) + 1
GOTO 555
CONTINUE
C
C ASSUMING TRAFFIC AT IRSC IS TAKEN CARE OF AUTOMATICALLY
C
DO 498 NL = 1, N3
NLINES(NODE, NL) = 0
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
C
C
CONTINUE
COSTFW(INODE+4)=0
RHOF(INODE)=0.

550  CONTINUE
RETURN
SUBROUTINE ISUMUP(L1,L2,L3,LT,IC)
***************
C  CALCULATE COST BETWEEN NODES L1 AND L2 AND ADD IT TO
C  TOTAL COST IC WHERE LT=LINE TYPE
C
***************
LL1=NUM(L1)
LL2=NUM(L2)
CALL ICOSTJ(LL1,LL2,L3,LNCHW,LNKL)
KK=N3
IF(LT.NE.0) KK=1
DO 211 LINTYP=1, KK
LTP=LINTYP
IF(KK.EQ.1) LTP=LT
DO 221 I=1,2
IC=IC+LNKCLN(LTP+1, I)
DO 222 I=1, N4
IC=IC+LNKCHW(LTP+2, I)
222 CONTINUE
211 CONTINUE
C
RETURN
SUBROUTINE ESWIL
***************
C  TRY AGAIN TO OPTIMIZE THE NETWORK
C
***************
5000  CONTINUE
K=IARRAY(RSC+2)  QFIRST SUBNETWORK UNDER RSC
KNEXT=IARRAY(K+3)  QNEXT SUBNETWORK UNDER RSC
IF(KNEXT .EQ. 0)  GOTO 599  QONLY ONE SUBNETWORK
560  CONTINUE
IK=0
L=IARRAY(RSC+2)  QSUBNET IS TO PE LINKED TO L=SUBNET
570  CONTINUE
IF(L .NE. K)  GOTO 575
L=IARRAY(L+3)
IF(L.EQ.0)  GOTO 660
575  CONTINUE
K1=NUM(K)
DIFF=DIST(M1,K1)
C
C  QFIRST TOTAL NO. OF TERMINALS IF K AND L ARE COMBINED
C  INTTRY= QINDICATION OF ENTRY TO TRYLNK
LINE=COSTFW(K+2)
IF(LINCAP(LINF) .GT. 9600)  GOTO 585  QNO MULTIDROPPING ON 9600
NODE=IARRAY(K+1)+IARRAY(L+1)+2
IF(NODE .GT. NTERMS)  GOTO 585  QTOO MANY TERMINALS
283  M=L
284  K=K
580 CONTINUE
581 M=NUMR(M)
582 DTRY=DISN(K1+M)/2.
583 IF(DTRY.GT.1.0) GOTO 140
584 CALL TPYLNK(K+KI+M) Q M IS THE INSERTION NODE.
585 IF(I0K.EQ.0) GOTO 585
586 140 CONTINUE
587 M=NUMR(KI)
588 IF(M.GT.1) GOTO 569
589 CALL TRYLNK(K1) Q M IS THE INSERTION NODE.
590 IF(I0K.EQ.0) GOTO 585
591 K=NUMR(KI)
592 M=K+1
593 IF(M.NE.0) GOTO 591
594 CONTINUE
595 L=ARRAY(L+3) Q NEXT SUCCESSOR
596 IF(L.NE.0) GOTO 570
597 660 CONTINUE
598 K=ARRAY(K+3)
599 IF(K.NE.0) GOTO 560 Q NOT AN END YET. REPEAT THE SEARCH.
600 CONTINUE
601 C ALL POSSIBLE COMBINATIONS HAVE BEEN TRIED
602 C IF(MAXSAV.LE.0) GOTO 599 Q NO NEED TO GO FURTHER
603 C CONTINUE
604 C UPDATING NETWORK BASED ON UP-TO-DATE MAXIMUM COST SAVING
605 C PARAMETERS
606 C JTALLY=JTALLY+1
607 C CALL UPNETW
608 C C REINITIALIZATION
609 C RSNUM=0.
610 C MAXSAV=0
611 C MAXK=0
612 C MAXL=0
613 C MAXM=0
614 C MAXI=0
615 C MAXLIN=0
616 C LINNEW=0
617 C MAXN=0
618 C RHOMAX=0.
619 C GOTO 5000
620 CONTINUE
621 C PRINT OUT COSTS FOR THE OPTIMIZED MULTIDROP NETWORK
622 C CALL MULTDRP
623 C PRINT OUT THE OPTIMIZED MULTIDROP NETWORK
624 C CALL NFPRTR
625 IF(MPLOT.NE.1) GOTO 50
626 CALL CALPLT
627 CONTINUE
628 RETURN
SUBROUTINE TRYLINK(KL,KIL,LL,ML)

**********
C TRY TO ELIMINATE CENTRAL LINK KL AND LINK IT TO THE SUBNETWORK
C LL THROUGH SYSTEM TERMINATIONS KIL AND ML.
C
**********
C
INTEGER COSTKL,COST
ITALLE=ITALLE+1
IF(INTRY .EQ. 1) GOTO 719
TFIN=ARRAY(KL)+ARRAY(LL)+ML
TOUT=ARRAY(KL)+ARRAY(LL)+ML
C
C FIND THE LINE WITH THE ENOUGH CAPACITY TO HANDLE
C THE TOTAL TRAFFIC ON THE PROPOSED SUBNETWORK LL
C
CALL LINK(KL,TFIN,TOUT,DUMMY,LINNEW,1,RHO)
IF(LINCAP(LINNEW) .GE. 9600) GOTO 132
LINUP=LINNEW+1
IF(LINUQ .EQ. 0) GOTO 712
DO 711 NL=1,LINUP
711 CONTINUE
712 CONTINUE
NLNEW=LNEW(DUMMY)(LINNEW)
IF(NLNEW .GE. 1) GOTO 132 (MORE THAN 1 LINE NOT ALLOWED)
C
C IF LIN TYPE IS THE HIGHEST, NO NEED TO GO FURTHER
C
IF(LINNEW .GE. 3) GOTO 132
LINNEW(LINNEW)=D
LINNEW(LINNEW)=E
IF(LINCAP(LINNEW) .GE. 9600) GOTO 132
LDUMMY(LINNEW)=E
NLNEW=1
CALL RHOFUN(TFIN,TOUT,DUMMY,LINNEW,RHO,LINNEW,RHO)
GOTO 3000
3000 CONTINUE

C TEST RESPONSE TIME; IF NOT SATISFIED, INCREASE LINE CAPACITY
C
CALL RSPST(KL,LL,LINNEW,INK)
IF(INK .EQ. 0) GOTO 3001
C
C IF LINE TYPE IS THE HIGHEST, NO NEED TO GO FURTHER
C
IF(LINNEW .GE. 3) GOTO 132
LINNEW(LINNEW)=D
LINNEW(LINNEW)=E
IF(LINCAP(LINNEW) .GE. 9600) GOTO 132
LDUMMY(LINNEW)=E
NLNEW=1
CALL RHOFUN(TFIN,TOUT,DUMMY,LINNEW,RHO,LINNEW,RHO)
GOTO 3000
3001 CONTINUE

CALL LCOSTK(IRSC,LL,1,COSTK) (NEW COST FOR SUBNET LL UNLINKED)

131 CONTINUE

C LINOLD=COSTEW(KL,2)
MCOSTK=COSTEW(KL,1)
NLOLD=COSTEW(KL,3)
IF(LINNEW .EQ. LINOLD .AND. NLOLD .GE. 1) GOTO 133
CALL LCOSTK(IRSC,KL,0,COSTK) (NEW COST FOR SUBNET UNDER KL)

A-22
GO TO 134
CONTINUE
ITEMP=0
KADD=0
KCHG=2
CALL ISUMP(IRSC•KL•LINNEW•ITEMP)
MCOSTK=MCOSTK+ITEMP
CONTINUE
INTRY=1 GFLAG THAT INDICATES AN ENTRY TO TRY LINK
JSAV=MCOSTEW(IL•1)+MCOSTEW(KL•1)-(MCOSTL+MCOSTK)
CONTINUE
COSTKM=0
KADD=0
CALL ISUMP(KL•KIL•LINNEW•COSTKM)
ISAVE=JSAY+COSTKM
IF (ISAVE •LE. MAXSAV) GO TO 132
RSPMAX=RSPMAX+IL
MAXSAV=JSAY
MAXX=KIL
MAXL=LL
MAXM=ML
MAXK=KL
MAXLINE=LINNEW
MAXNOL=LINNEW
RHO=MAX=MR
132 CONTINUE
RETURN
Si/UROUTINE LCOSTK(INA•NB•TCOST)
****************************************
C FIND COST FOR A SUBNETWORK, NA=BEGINNING NODE FOR THE SUBNET
C TO BE EVALUATED.
C NA=1 WHEN COST FOR CENTRAL LINK NA IS TO BE INCLUDED
C NA=0 WHEN COST FOR CENTRAL LINK NA IS NOT TO BE INCLUDED
C
***************
INTEGER TCOST
TCOST=0
KCHG=2
CALL ISUMP(INA•LINNEW•TCOST)
C START COMPUTING SUBNET COST
C
JSON=ARRAY(INA•2)
IF (JSON•EQ.0) GO TO 400
CONTINUE
JPA=ARRAY(JSON•5)
CALL ISUMP(JPA•JSON•LINNEW•TCOST)
JSON=JSON•NOD(NA•JSON)
IF (JSON•EQ.0) GO TO 400
GO TO 300
CONTINUE
IF (INB •EQ. 1) RETURN
ITEMP=3
KADD=0
CALL ISUMP(INA•LINNEW•ITEMP)
TCOST=TCOST+ITEMP
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.

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

NXTNOD=0

MM=M1

KSON=ARRAY(MM+2).

IF(KSON.EQ.0.AND. MM.EQ. L1) RETURN DA 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 0 NO MORE SUCCESSIONS WITH SAME PREDECESSOR

NXTNOD=KPRO

RETURN

CONTINUE

GO TO HIS FATHER

MM=ARRAY(MM+5)

IF (MM .NE. L1) GO TO 1 0 BACK TO THE BEGINNING

RETURN

CALL ROUTINE UPNETW

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

C UPDATE IARRAY AND COSTEW BASED ON MAXIMUM SAVING

C PARAMETERS OBTAINED

C UPDATE TRAFFIC AND NO. OF TERMINALS FOR L-SUBNET

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

NOK=ARRAY(MAXK+1)+1 0 NO. OF NODES BELOW MAXK

IARRAY(MAXL+1)=IARRAY(MAXL+1)+NOK

ARRAY(MAXL+1)=ARRAY(MAXL+1)+ARRAY(MAXK+1)

ARRAY(MAXL+2)=ARRAY(MAXL+2)+ARRAY(MAXK+2)

C UPDATE THE COSTEW

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

COSTEW(MAXL+2)=MAXLIN

COSTEW(MAXL+3)=MAXNOL

COSTEW(MAXL+4)=0

COSTEW(MAXK+2)=0

COSTEW(MAXK+3)=0

MAXKD=NUMR(MAXK)

MAXMD=NUMR(MAXM)

MAXKD=NUMR(MAXK)

MAXMD=NUMR(MAXM)

COSTEW(MAXL+4)=COSTEW(MAXL+4)+COSTEW(MAXK+4)+DIST(MAXKIN+MAXKMD)
513 * -DIST(MAXN1,N1)
514 RHOMAX=MAXH
515 MAXX(X+4)=0
516 C UPDATE MULTIDROPPED-LINE RESPONSE TIMF
517 C
518 C 91 CONTINUE
519 RHOMAX=MAXH
520 KIPA=ARRAY(MAXK1+5) "REMINDER KIPS PREDECESSOR
521 MSON=ARRAY(MAXM+2) "MIPS 1ST SUCCESSOR
522 CALL LNKOFF(MAXXI) "DELETE KI AS A SUCCESSOR OF KIPA
523 IARRAY(MAXM+2)=MAXH
524 IARRAY(MAXK1+5)=MAXH
525 IARRAY(MAXK1+3)=MSON
526 IF(MSON NE 0) IARRAY(MSON+4)=MAXH
527 IARRAY(MAXK1+4)=0
528 MAXH=MAXH
530 MAXH=KIPA
531 IF(MAXH NE MAXH) GOTO 91
532 RETURN
533 FUNCTION JCOSTA(N+KREF)
534 C ***************
535 C
536 C FIND PARTIAL SUM FOR ICST1N
537 C
538 C ***************
539 JCOSTA=0
540 DO 777 K1=1,KREF
541 JCOSTA=JCOSTA+ICST1N(K1+1)
542 777 CONTINUE
543 RETURN
544 FUNCTION JCOSTR(N+M+KREF)
545 C ***************
546 C
547 C FIND PARTIAL SUM FOR ICSTHW
548 C
549 C ***************
550 JCOST=0
551 DO 77A KK=1,KREF
552 JCOST=JCOST+ICSTHW(KK+1)
553 77A CONTINUE
554 RETURN
555 SUBROUTINE NETPRT
556 C ***************
557 C
558 C PRINT OUT CONFIGURATION OF THE MULTIDROP NETWORK
559 C
560 C ***************
561 DO 196 KK=1,N1
562 IBLANK(KK)=JBLANK(KK)
563 196 CONTINUE
564 NIP=NUMBER(IRSC)
565 WRITE(INT197,197) N1
566 197 FORMAT(1X,*REGIONAL CENTFR=*A4/*6X,*SUBNETWORK=*6X*)
567 * *BEGIN AT /*/1
568 KP=1
569 ISON=ARRAY(IRSC+2)
570   IP0INT=ISON
571   IS0N=UMARR(ISON)
572   WRITE(IWT+192) (IRLANK(I),I=1,KP),ISON
573   C
574   C Look for its first successor
575   C
576  190 CONTINUE
577   ISON=IARRAY(IP0INT+2)  CURRENT NONNAL INDEX
578   IF(IISON .EQ. 0) GOTO 191  AND MORE SOON
579   KP=KP+1  GA LEVEL DEEPER
580   ISON=UMARR(ISON)
581   WRITE(IWT+192) (IRLANK(I),I=1,KP),ISON
582  192 FORMAT(1X,P4(A6))
583   IP0INT=ISON
584   GO TO 190
585   191 CONTINUE
586   C
587   C Look for next successor with the same predecessor
588   C
589   IRAR=IARRAY(IP0INT+3)
590   IF(IRAR .EQ. 0) GOTO 193  AND MORE TO GO FURTHER
591   IRAR=UMARR(IRAR)
592   WRITE(IWT+192) (IBLANK(I),I=1,KP),IRAR
593   IP0INT=IRAR
594   GO TO 190
595   193 CONTINUE
596   C
597   C Next level up
598   C
599   KP=KP-1
600   IP0INT=IARRAY(IP0INT+5)
601   IF(KP .EQ. 0) GOTO 194  AND NPER TO GO FURTHER
602   GO TO 191
603  194 CONTINUE
604   RETURN
605   SUBROUTINE CONVRT(ICOST)
606   ***************
607   C
608   C Convert a number into its field equivalent
609   C
610   C ***************
611   JCHAR(I)=JRLANK
612   JCHAR(2)=JRLANK
613   IF(ICOST .EQ. 0) GOTO 916
614   ENCON(198,JCHAR) ICOST
615  198 FORMAT(18)
616  916 CONTINUE
617   RETURN
618   SUBROUTINE SUMPRT(NREF,NN)
619   ***************
620   C
621   C SUM up costs and prints
622   C
623   C ***************
624   TCOST1=0
625   TCOST2=0
626   DO 779 K=1,NREF
ITCOST(K1)=ICSTLN(K1) + ITCSTM(K1)
DO 7791 KC=1,N4
7791 CONTINUE

ICOST2=ICOST1+ITCOST2

779 CONTINUE

CONTINUE

WRITE(OUT,COST1+ITCOST2)

C PRINT OUT COST

C

NTRMN=nREF/10+1
ITEM=MOD(nREF,10)
IF (ITEM =EQ. 0) NTRMN=NTRMN-1

DO 919 KW=1,NTURN

DO 919 KW=1,NTURN

KWL=10*(KW-1)+1

KMU=10*KW

IF (KMU. GT. nREF) KMU=nREF

IF (N, EQ. 0) GOTO 979

IF (LPAGE, NE. 1) GOTO 9033

WRITE(INT, 9031) KW

9031 FORMAT('1',3,60X,'REGIONAL STAR NETWORK AND ITS COSTS=',1P)

GOTO 9035

9035 CONTINUE

WRITE(INT, 9034) KW

9034 FORMAT('1',3,60X,'REGIONAL STAR NETWORK AND ITS COSTS=',1P)

9035 CONTINUE

WRITE (INT, 9032) (NUMN(I), I=KWL, KMU)

9032 FORMAT('1',3,60X,'SYSTEM TRMN=',1S,10(4X, A, I=1,1))

WRITE (INT, 9033) KW

9033 FORMAT('1',3,60X,'NO. OF LINES REG=',1S)

DO 1803 N=KWL, KMU

1803 FORMAT('1',3,60X,'LINE UTILIZATION=',1S,10(4X, A, I=1,1))

WRITE (INT, 9036) (RHOF(N), N=KWL, KMU)

9036 FORMAT('1',3,60X,'LINE UTILIZATION=',1S,10(4X, A, I=1,1))

WRITE (INT, 9037) (IDST(N), N=KWL, KMU)

9037 FORMAT('1',3,60X,'DSTNCF FROM RSC=',1S,10(4X, A, I=1,1))

GO TO 806

806 CONTINUE

WRITE (INT, 9038) (KWL, KMU)

8038 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

8033 CONTINUE

WRITE (INT, 8034) KW

8034 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

8035 CONTINUE

WRITE (INT, 8035) (KWL, KMU)

8035 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

WRITE (INT, 8036) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8037) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8038) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8039) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8040) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8041) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8042) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8043) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8044) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8045) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8046) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8047) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8048) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8049) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8050) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8051) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8052) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8053) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8054) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8055) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8056) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8057) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8058) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8059) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)

GOTO 8033

DO 1803 N=KWL, KMU

 WRITE (INT, 8060) (KWL, KMU)

1803 FORMAT('1',3,60X,'FINAL MULTIDROP NETWORK AND ITS COSTS=',1P)
77-53, Vol. IV

58488 \text{MSUB(N)=NUMARR(ID)}
5855 \text{LSUB(N)=IARRAY(ID)+1)}
5866 \text{CONTINUE}
5877 \text{WRITE(IWT,1006)(MSUB(N),N=KWL,KWU)}
5888 \text{FORMAT(5X+BEGINNING NODE*,11X10(1X,A6,1X))}
5899 \text{WRITE(IWT,1007)(LSUB(N),N=KWL,KWU)}
5900 \text{FORMAT(3X+NO. OF TERM*,12X10(1A1,1X))}
5911 \text{WRITE(IWT,811)}
5922 \text{FORMAT(3X+NO. OF LINES*1)
5933 \text{DO 1008 NJ=1,N3}}
5944 \text{IF(LINMX(NJ).EQ.0) GOTO 1008}
5955 \text{WRITE(IWT,904) LINAME(NJ),(NLINES(K,NJ),K=KWL,KWU)}
5966 \text{CONTINUE}
5977 \text{WRITE(IWT,808)(RHOF(NJ),NJ=KWL,KWU)}
5988 \text{FORMAT(3X+LINE UTILIZATION*,9X10(F8.3,1X))}
5999 \text{WRITE(IWT,808)(IDST(N),NJ=KWL,KWU)}
6000 \text{FORMAT(3X+TOTAL MILEAGE*,12X10(1A1,1X))}
6011 \text{WRITE(IWT,808)}
6022 \text{CONTINUE}
6033 \text{DO 1101 N=KWL,KWU}
6044 \text{ID=N}
6055 \text{IF(NN+F0.0) INNSUB(N)}
6066 \text{TRFSUM(N,1)=ARRAY(ID,1)}
6077 \text{TRFSUM(N,2)=ARRAY(ID,2)}
6088 \text{TIMOUT(N)=TIMSP(ID)}
6099 \text{CONTINUE}
6100 \text{WRITE(IWT,1102)(TRFSUM(N+1,N=KWL,KWU)}
6111 \text{FORMAT(3X+TRAFFIC*,3X+LINE TO CPU*,11X10(F8.3,1X))}
6122 \text{WRITE(IWT,1103)(TRFSUM(N+2),N=KWL,KWU)}
6133 \text{FORMAT(3X+CPU TO LINE*,10X10(F8.3,1X))}
6144 \text{WRITE(IWT,1104)(TIMOUT(N),N=KWL,KWU)}
6155 \text{FORMAT(3X+LINE RESPONSE TIME*,7X10(F8.3,1X))}
6166 \text{WRITE(IWT,907)}
6177 \text{FORMAT(3X+SUBTOTAL*/1X+INST. COSTS*)}
6188 \text{COST=JCOSTA(1,NREF)}
6199 \text{CALL CONVRT(COST)}
6200 \text{WRITE(IWT,908)(JCHAR(L),L=1,2), (ICSTLN(NODE*,1),NODE=KWL,KWU)}
6211 \text{FORMAT(5X+LINES*,8X,A6,A2,1X10(1F1.1X))}
6222 \text{DO 1909 K=1,N4}
6233 \text{COST=JCOSTB(K1,NREF)}
6244 \text{IF(KW.NE.1) COST=0}
6255 \text{CALL CONVRT(COST)}
6266 \text{WRITE(IWT,909) NAMEHW(K),(JCHAR(L),L=1,2), (ICSTHW(NODE*,K),1}
6277 \text{+ NODE=KWL,KWU)}
6288 \text{FORMAT(5X+A6,7X+A6,A2,1X10(1A1,1X))}
6299 \text{CONTINUE}
6300 \text{WRITE(IWT,910)}
6311 \text{FORMAT(5X+ANNUAL RECUR. COST*)}
6322 \text{COST=JCOSTA(2,NREF)}
6333 \text{IF(KW.NE.1) COST=0}
6344 \text{CALL CONVRT(COST)}
6355 \text{WRITE(IWT,908)(JCHAR(L),L=1,2), (ICSTLN(NODE*,1),NODE=KWL,KWU)}
6366 \text{DO 1911 K=1,N4}
6377 \text{COST=JCOSTB(K1,NREF)}
6388 \text{IF(KW.NE.1) COST=0}
6399 \text{CALL CONVRT(COST)}
6400 \text{WRITE(IWT,909) NAMEHW(K),(JCHAR(L),L=1,2), (ICSTHW(NODE*,K),1}
6411 \text{A-28}
* NODE=WLU+KWL!
1911 CONTINUE
WRITE(IWT,+912)
912 FORMAT(1X,TOTAL COST=)
913 IF(KW.NE.1) TCOST=1
914 TCOST=TCOST+KW1
915 IF(KW.NE.1) TCOST=TCOST+KW1
916 CALL CONVERT(TCOST)
917 WRITE(IWT,+913) (JCHAR(L),L=1,2),(ITCOST(K,1),K=WLU,KWL)
918 IF(KW.NE.1) TCOST=TCOST+KW1
919 CONTINUE
WRITE(IWT,+914) KWRITE(TCOST)
920 FORMAT(*25X,TOTAL COST=*,I8)
921 RETURN
*** MULTIDROP ROUTINE ***
C PRINT OUT FINAL MULTIDROP NETWORK WITH ITS COSTS
C
C
DO 590 NL=1,N3
590 LDUMMY(NL)=0
IF(IRO=IAAY(IRO+2),1,N1)
599 CONTINUE
IF(IRO.EQ.0) GOTO 698
NK2=NUMR(IRRO)
NK1=IRRO
NSIR(IK1)=IRO
LINE=COSTFW(IRO+2)
LDUMMY(LINE)=COSTFW(IRO+3)
JS=IAAY(IRO+2)
IF(JSON.EQ.0) GOTO 694
DO 592 NK1=1,2
592 ICSTLW(K1,NM)=0
DO 593 NK=1,N4
593 ICSTHW(K1,NK,NM)=0
DO 594 NL=1,N3
594 NLINES(K1,NL)=LDUMMY(NL)
596 CONTINUE
KCHG=2
312 CONTINUE
595 CONTINUE
312 CONTINUE
IF(JSON.EQ.0) GOTO 311
NK2=NUMR(JSON) GGLOBAL INDEX FOR NEXT NODE
77-53, Vol. IV

798  \text{NK1=ARRAY(JSON+5), PRFDECFSOR}
799  \text{NK1=NNUMA(NK1), GLOBAL INDEX FOR PRFDECFSOR}
800  \text{JSON=NXTOD(IRAO+JSON)}
801  \text{GOTO 312}
802  \text{311 CONTINUE}
803  \text{LNUMY(LINE)=0}
804  \text{GOTO 591}
805  \text{694 CONTINUE}
806  \text{C}
807  \text{C USE PREVIOUS DATA}
808  \text{C}
809  \text{DO 597 NL=1, N3}
810  \text{NLINES(K1+NL)=NLINES(IRAO+NL)}
811  \text{597 CONTINUE}
812  \text{DO 598 NM=1, 2}
813  \text{ICSTLN(K1+NM)=ICSTLN(IRAO+NM)}
814  \text{DO 598 NK=1, N4}
815  \text{ICSTLN(K1+NK+NM)=ICSTLN(IRAO+NK+NM)}
816  \text{598 CONTINUE}
817  \text{LNUMY(LINE)=0}
818  \text{599 CONTINUE}
819  \text{K1=COSNEG(IRAO+4)}
820  \text{RHOF(K1)=RHOF(IRAO) SHRUFLING RHO’S DUE TO RF-NEXTING}
821  \text{IRAO=ARRAY(IRAO+3)}
822  \text{K1=K1+1}
823  \text{GOTO 699}
824  \text{698 CONTINUE}
825  \text{NOSUB=K1-1}
826  \text{CALL SIMPROT(NOSUB+0)}
827  \text{RETURN}
828  \text{SUBROUTINE CALPLT}
829  \text{**************}
830  \text{C}
831  \text{C PLOT A MULTIDROP NETWORK}
832  \text{C}
833  \text{C**************}
834  \text{KP=1}
835  \text{IPOINT=IRSC}
836  \text{CALL TRSFRM(2)}
837  \text{ISON=ARRAY(IIRSC+2) QFIRST SUCCESSOR}
838  \text{IPOINT=ISON}
839  \text{CALL TRSFRM(1)}
840  \text{C}
841  \text{C LOOK FOR ITS FIRST SUCCESSOR}
842  \text{C}
843  \text{190 CONTINUE}
844  \text{ISON=ARRAY(IPOINT+2) QFIRST SUCCESSOR}
845  \text{IF(ISON .EQ. 0) GOTO 191}
846  \text{KP=KP+1}
847  \text{IPOINT=ISON}
848  \text{CALL TRSFRM(1)}
849  \text{GOTO 190}
850  \text{191 CONTINUE}
851  \text{C}
852  \text{C LOOK FOR ITS NEXT SUCCESSOR}
853  \text{C}
854  \text{IRAO=ARRAY(IPOINT+3)}
855  IPO=1 ARRAY(IPO+5) GOMOW ITS PREDECESSOR
856  CALL TRSFPM(2)
857  IF(IPO .EQ. 0) GOTO 193
858  IPO=1
859  CALL TRSFRM(1)
860  GOTO 190
861  CONTINUE
862  C
863  C GO BACK TO ITS PREDECESSOR
864  C
865  KP=KP+1
866  IF(KP .EQ. 0) GOTO 194
867  GOTO 191
868  CONTINUE
869  CALL TRSFRL(3)
870  RETURN
871  SUBROUTINE TRSFRL(LK)
872  ***************
873  C
874  C FIND GLOBAL MADAR INDEX FOR X-Y COORDINATES AND PID NO1
875  C
876  ***************
877  DATA IPO/0/
878  IF(LK .EQ. 3) GOTO 666
879  LKI=NUMR(IPO) GGOLOBAL INDEX
880  IGD=MAPADR(LKI) GMAPADR INDEX FOR LK1
881  IF(IID .EQ. IP) RETURN
882  IP=IID
883  666 CONTINUE
884  CALL PLOTPT(IID,LK)
885  RETURN
886  SUBROUTINE LNKFPO(MP)
887  ***************
888  C
889  C DELETE MP AS A SUCCESSOR OF NONE PA
890  C
891  ***************
892  IFRONT=ARRAY(MP+4) THE SUCCESSOR REFERENCE MP
893  IBACK =ARRAY(MP+3) THE SUCCESSOR AFTER MP
894  IF(IFRONT .NE. 0) GOTO 92
895  MPA=ARRAY(MP+5)
896  IARRAY(MPA+2) IBACK 1ST SUCCESSOR UNDER NEW MPA
897  GOTO 99
898  92 CONTINUE
899  IARRAY(IFRONT+3)=IBACK
900  99 CONTINUE
901  IF(IBACK .EQ. 0) RETURN
902  IARRAY(IBACK+4)=IFRONT
903  RETURN
904  SUBROUTINE RSPTST(KK,L,LMAX,10K)
905  ***************
906  C
907  C TEST RESPONSE TIME, SATISFIED WHEN 10K=1
908  C
909  ***************
910  MDROP=ARRAY(LLL+1)+ARRAY(KKK+1)+2
911  TRFIN=ARRAY(LLL+1)+ARRAY(KKK+1)
912  TROUT=ARRAY(LLL+2)+ARRAY(KKK+P)
913  CALL RSPNRC(TRFIN,TROUT,LMAX,MDRP+10K)
914  RETURN
915  END

GPR3 STACOM IRNOP/0777
SUBROUTINE IRNOP(NR,LIMIT,TRM)

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

SUPPRAM FOR THE INTER-REGION NETWORK OPTIMIZATION

LIMIT=MINIMAL NUMBER OF PATHS NEEDED PER REGIONAL SWITCHING CENTER

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

PARAMETER NP1=130,NP2=1,NP3=4,NP4=3
PARAMETER NPC=369,NP6=(NPC*NPC/2-NPC+1)/4+1
PARAMETER NP7=11, IT=100, M=4
COMMON/CONST/N1,N2,N3,N4,N7+NCITY
COMMON/LINCHR/LINMX(NP3)+LINCAP(NP3)+UTILIZ(NP3)
* /BCOS*/!*NSTC(NP2+NP3,NP4,3,2,2)+RFCRC(NP2*NP3,NP4,7,2,2)
* ANSTLN(NP2*NP3,3,2,3)+RFCLN(NP2*NP3,NP3,3,2,16)+SUPLY(NP3)
* /NAME/INXPT(NP1)+NAMEET(NP1)+LINEF(NP3)+NAMEHW(NP4)
* /E/N/S/RA(NP1)+NSC(NW)+NNPR(N)+TRAFA(NP1)+TAFFI(NP1)
* /RFF/REFP(NP1)+TRAFF(NP1+2)+NP7,NSTNC(NP6)+MARPDP(NP1)
DIMENSION NETSUM(NP3,2)+ORINET(MW,MW,NP3)
DIMENSION NLINK(NP3)+LNKCHW(NP3,NP4,2)+LINX1N(NP3,2)
INTEGER SUMCST
INTEGER ORINET
DIMENSION TRM(MW,MW),TRJ(MW)
INTEGER ORICST,ORICS1,ORICS2
INTEGER DIVTR(3,MW),DIVTRJ(MW)
DIMENSION TRM(MW+MW)+NETCF(NW,MW,NP3)+LINEQ(NP3)
* LINAD(NP3)+LINAD(NP3)+LINAD(NP3)
* LIGF(NP3)+LIGA(NP3)+LIGD(NP3)
DIMENSION RHOF2(MW,MW)
DIMENSION TRM(MW,MW)
EQUIVALENE (LINEQ,LINEOA)+(LINEQ+LINFOR)

C RESET UTILIZATION FACTOR TO .5
C
DO 70 NN1=1,N3
70 CONTINUE
C
C COMPUTE ORINET(MW+MW+N3) FOR INITIAL TOPOLOGY WHERE N3 IS THE NUMBER OF CHARGEABLE ITEMS

C ORICST=0
ORICS1=0
ORICS2=0
NRI=NRI-1
C DO 203 NN1=1,N3
DO 203 NN2=1,N3
C       NETSUM(NN1,NN2)=0 GCOST SUM
203 CONTINUE
C
C MODIFY DUPLAXING MODE FROM HALF TO FULL, DUPLFY
C
DO 667 K1=1,N3
C
667 CONTINUE
DO 101 I=1,NRI
57 NLINK(I)=NR1  QNR1 LINKS AT THE BEGINNING
58 I1=I+1
59 DO 102 J=I1,NR
60 I2=NRS(I)
61 JMR=E(RS(I),J)
62 ATRMAX=ATAIAXI(T1(I,J), TR1(J,I))  QASSUMING FULL DUPLEX
63 CALL LINNUM (ATRMAX.O, LINFO.LINKUP.O, RHO)
64 RHO2(I,J)=RHO
65 RHO2(J,I)=RHO
66 CALL ICOSTJ (LONGL, I,J, LNKCHW, LIN)
67 DO 104 NN=I+1, NR
68 ORINJ(I,J,NN)= LINEQ(NN)
69 ORINJ(J,I,NN)= LINEQ(NN)
70 DO 105 NN=I+2  Q LINE COST
71 NETSUM(NN,NM)= NETSUM(NN,NM)+ LNKCHW(NN,NM)
72 DO 106 NK=1,NM  Q 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 DO 107 K1=1,NR
80 DO 107 NN=1,N3
81 ORINJ(K1,N1,NN)=0
82 107 CONTINUE
83 CALL OUTPUT(I)
84 ITALY=0
85 999 CONTINUE
86 MAXSAVE=0
87 DO 777 I=1,NR
88 IF (NLINK(I) .LE. LIMIT) GO TO 777
89 I1=I+1
90 DO 788 J=I1,NR
91 IF (NLINK(J) .LE. LIMIT) GO TO 788
92 IN=NTEST(ORINJ+J)
93 IF (IN .EQ. 0) GO TO 788  QNO 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=1,NR
101 L1=L+1
102 DO 138 M=1,NR
103 IFTEST(I,J,L,M)
104 IF (IN.EQ. 1) GO TO 110  QNEXT STEP NOT TO BE TESTED
105 IFTEST(I,J,L,M)
106 IF (IN.EQ. 1) GO TO 138
107 CONTINUE
108 DO 137 N=1,NR
109 IF (L.EQ. N) GO TO 137
110 IFTEST(I,J,L,N)
111 IF (IN.EQ. 1) GO TO 137
112 IFTEST(ORINJ+L,N)
113 IF (IN.EQ. 0) GO TO 137
77-53, Vol. IV

114     IY=IY+1
115     IF (IY.EQ.1) GO TO 137
116     IN=1
117     IF (IN.EQ.1) GO TO 138
118     137 CONTINUE
119     GO TO 788
120     138 CONTINUE
121     CONTINUE
122     SNIVT=0
123     SOIVT=0
124     CALL TRFDIV(IFLOP)
125     IF (IFLOP.EQ.1) GO TO 201
126     CALL MINAD(IID,MINCST)
127     GO TO 202
128     201 CONTINUE
129     CALL NETWK(MINCST)
130     CONTINUE
131     ISAV=ORICST-MINCST
132     IF (MAXSAV .GT. ISAV) GO TO 78A
133     MAXSAV=ISAV
134     IFLIP=IFLOP & GFLIP=GLOBAL INDICATOR
135     IF (IFLOP.EQ.1) GO TO 204
136     JMAX=J
137     IMAX=I
138     JMAX=J
139     DO 666 NN=1,N3
140     ILNAD(NN)=LADJ(NN) CHANGE OF LINE RFG.
141     JLNAD(NN)=JLNADJ(NN)
142     666 CONTINUE
143     204 CONTINUE
144     DO 331 K1=1,NR
145     DO 331 K2=1,NR
146     TRR(K1,K2)=TRR(K1,K2)
147     331 CONTINUE
148     78B CONTINUE
149     777 CONTINUE
150     IF (MAXSAV .LE. 0) GO TO 9999
151     CALL NETUP(IFLIP,JMAX,IMAX,JMAX)
152     ITALL=ITALITY+1
153     GO TO 999
154     9999 CONTINUE
155     109 FORMAT(1X,'THIS NETWORK HAS BEEN UPDATED FOR ',F16.10) ITALY
156     WRITE(6,109) ITALY
157     DO 81 I=1,N3
158     DO 81 J=1,2
159     NETSUM(I,J)=0
160     81 CONTINUE
161     DO 91 I=1,NR1
162     K=I+1
163     DO 92 J=K,NR
164     DO 93 K1=1,N3
165     LINEO(K1)=ORINET(I,J,K1)
166     93 CONTINUE
167     IJ=HRSC(I)
168     JJ=HRSC(J)
169     CALL ICOSTJ(LINEO,I,J,JJ,LMKCHW,LMKCLN)
170     DO 94 KK=1,N3
NETSUM(KK+1)=NETSUM(KK)+LNKCLN(KK+1)
NETSUM(KK+2)=NETSUM(KK)+LNKCLN(KK+2)
DO 95 KL=1,N3
NETSUM(KK+1)=NETSUM(KK)+LNKCHW(KK+KL+1)
NETSUM(KK+2)=NETSUM(KK)+LNKCHW(KK+KL+2)
CONTINUE
177 CONTINUE
178 CONTINUE
179 CONTINUE
180 ORIC51=0
181 ORIC52=0
182 CALL OUTPRT(2)
183 RETURN
184 C
185 C SUBROUTINE OUTPRT(N)
186 C
187 C PRINT OUT INTERREGIONAL NETWORK CONFIGURATION AND ITS COSTS
188 C
189 C
190 DO 110 I=1,N3
191 ORICSI=ORIC51+NETSUM(I+1)
192 ORICS2=ORIC52+NETSUM(I+2)
193 CONTINUE
194 NTURN=NR/10+1
195 DO 200 I=1,NTURN
196 LL=(NTURN-1)*10 + 1
197 LU=NTURN*10
198 IF(LU .GT. NR) LU=NR
199 IF(N .LT. 2) GOTO 2100
200 WRITE(IWT,2002) (IJ,J=LL,LU)
201 2002 FORMAT(11//,10X,"INITIAL INTERREGIONAL NETWORK CONFIGURATION",
+//,4//10X,13.2X)
202 GOTO 2101
203 2101 CONTINUE
204 WRITE(IWT,2012) (IJ,J=LL,LU)
205 2012 FORMAT(11//,10X,"FINAL OPTIMAL INTERREGIONAL NETWORK CONFIGURATION",
+//,4//20X,10(5X,13.2X))
206 2102 CONTINUE
207 DO 2003 I=1,N3
208 WRITE(IWT,2034) (I,J=1,LU)
209 WRITE(IWT,2031) I
210 2034 FORMAT(11//,10X,"REGION",//14X)
211 DO 2104 M=1,N3
212 WRITE(IWT,2035) (I,J=1,LU)
213 2035 FORMAT(11//,10X,"LINE UTILIZATION",//14X)
214 2104 CONTINUE
215 WRITE(IWT,2005) (4X,I,J=1,LU)
216 2005 FORMAT(11//,10X,"INSTRUCTION COST",//14X)
217 WRITE(IWT,2006) (I,J=1,LU)
218 2006 FORMAT(11//,10X,"RECEIVED COST",//14X)
219 2004 CONTINUE
220 WRITE(IWT,2001) I
221 2001 FORMAT(11//,10X,"TOTAL")
222 DO 2005 K=1,N3
223 ISUM=NETSUM(K)+NETSUM(K+P)
224 WRITE(IWT,2007) K,LINAME(K),ISUM
225 2007 FORMAT(11//5X,K,LINAME(K),NETSUM(K),15X)
226 WRITE(IWT,2008) K,LINAME(K),ISUM
227 2008 FORMAT(11//5X,LINAME(K),ISUM)
228 CONTINUE
229 WRITE(IWT,2009) I
230 2009 FORMAT(11//5X)
231 END
WRITE(IWT,2009) ORICS1,ORICS2,OPTCS
200 FORMAT(/'9X,TOTAL,4X,IT,10X,IT,10X,IT)
RETURN

FUNCTION ITEST(I,J,K,L)

TEST=0
IF (I .EQ. K .AND. J .EQ. L) ITEST=1
IF (I .EQ. L .AND. J .EQ. K) ITEST=1
RETURN

FUNCTION ITEST(NET,I,J)

TEST=0
RETURN

C TFST DIRECT LINE CONNECTIVITY BETWEEN I AND J.

C DIMENSION NET(MW/MW,N3)
DO 103 II=1,N3
IF (NET(I,J,II) .GT. 0) GO TO 10A
103 CONTINUE
IF TEST=0 
   RETURN
10A CONTINUE
IF TEST=1 
   RETURN
   Q YES, THERE IS A CONNECTION
   SUBROUTINE TRFDIV(IFLOP)

C DIVERT TRAFFIC BETWEEN I AND J THROUGH OTHER RCS.

C IT RETURNS WITH IFLOP=1 WHEN SUCCESSFUL OR IFLOP=0.
C IT ALSO CREATES TEMPORARY MATRICES TTA AND NETCDF.

SNDIVT=0, Q TOTAL TRAFFIC DIVERTED (I TO J)
SNDIVJ=0, Q TOTAL TRAFFIC DIVERTED (J TO I)
DO 205 K=1,NR
IF (I.EQ.I .OR. I.EQ.J) GO TO 220
DIVRI(K)=0. Q TRAFFIC DIVERTED THRU REGION K (I TO J)
DIVRJ(K)=0. Q TRAFFIC DIVERTED THRU REGION K (J TO I)
205 CONTINUE
DO 220 II=1,NR
IF (II.EQ.I .OR. II.EQ.J) GO TO 220
IC1=TEST(ORIGNET,I,II)
IC2=TEST(ORIGNET,J,II)
IF (IC1 .EQ. 0 .OR. IC2 .EQ. 0) GOTO 220
C DIVERT I TO J TRAFFIC THRU II
C
DIVRI(II)=0.
DIVRJ(II)=0.
CALL LINTRA(I,J,A)
DELTP= A-TP(I,J)
IF (DELTP.EQ.0.0) GO TO 160

C
DIVTR=DELTR
CALL LINTRF(I,J+A)
DLTR= A-TR(I,J)
IF (DLTR.LE.0.0) GO TO 169
DIVTR(I,II)= AMIN(DLTR+DIVTR)
IF ((DIVTR(I,II) + SDIVTI, GT, TR(I,J)) GO TO 140
SDIVTI= SDIVTI+ DIVTR(I,II)
GO TO 160
C
DIVTR(I,II)= TR(I,J)- SDIVTI
SDIVTI= TR(I,J)
C
DIVERT J TO I TRAFFIC THRU II
C
CONTINUE
CALL LINTRF(J,II+A)
DELTR= A-TR(J,II)
IF (DFLTR.LE.0.0) GO TO 220
C
DIVTR=DELTR
CALL LINTRF(I,J+B)
DLTR= A-TR(I,J)
IF (DLTR.LE.0.0) GO TO 220
DIVTR(I,II)= AMIN(DLTR+DIVTR)
IF ((DIVTR(I,II) + SDIVTI, LT, TR(J,II)) GO TO 140
SDIVTI= SDIVTI+ DIVTR(I,II)
GO TO 200
C
DIVTR(I,II)= TR(J,II)- SDIVTI
SDIVTI= TR(J,II)
C
CONTINUE
CALL LINTRF(J,II+A)
C
CONTINUE
IF ((SDIVTI, EQ, TR(J,II)) AND (SDIVTI, EQ, TR(J,II))) GO TO 340
CONTINUE
GO TO 360
IFLOP=0
GO TO 360
IFLOP=1
CONTINUE
C
C CREATE A NEW TRAFFIC MATRIX WHICH ELIMINATES THE TRAFFIC BETWEEN N AND J AND A TEMPORARY NETWORK NETCFN FOR THE PURPOSE
C OF COST EVALUATION
C
DO 191 K1=1, NR
DO 191 K2=1, NR
TRR(K1,K2)=TR(K1,K2)
DO 191 K3=1, NR
NETCFN(K1,K2,K3)=ORINT(K1,K2,K3)
191 CONTINUE
DO 190 K1=1, NR
NETCFN(I,J,K1)=0
NETCFN(I,J,K1)=0
NETCFN(J,J,K1)=0
190 CONTINUE
TRR(I,J)=0
TRR(J,I)=0

A-37
DO 380 IK=1, NR
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(IK)
TRR(J+IK) = TR(J+IK) + DIVTR(JK)
ATRMAX=AAMAX1(TRR(I+IK)+TRR(J+IK))
BTRMAX=AAMAX1(TRR(J+IK)+TRR(I+IK))
IDR1=NRSC(I)
IDR2=NRSC(J)
CALL LTNUM1 (ATRMAX,0,LINFO,LINUP,O,RHO)
RHOF2(I+IK)=RHO
RHOF2(J+IK)=RHO
DO 430 II=1, NR
NETCNF(I+IK+NN)=LINEQ(NN)
NETCNF(J+IK+NN)=LINEQ(NN)
NETCNF(J+IK+NN)=LINEQ(NN)
NETCNF(I+IK+NN)=LINEQ(NN)
430 CONTINUE
380 CONTINUE
RETURN
SUPERROU TINE LINTRF (I,J,K)
C
*************
C CONVERT LINES INTO TRAFFIC CAPACITIES BETWEEN NODES I AND J.
C
C
*************
A20
DO 100 IR=1, NR
A20=+ ORNET(I+J,IR)*LINCAP(IR)+UTIL1(IR)
100 CONTINUE
RETURN
SUPERROU TINE NETKC(SUMCST)
*************
C
C FIND TOTAL INTERREGIONAL NETWORK COST, SUMCST, BASED ON SPECIFIC
C CONFIGURATION NETCNF
C
*************
C
INTEGER SUMCST
SUMCST=0
DO 420 IR=1, NR
IR=IR+1
DO 400 IK=IR+1
IC=HEST(NETCNF(IP,IK))
400 IF ( IC.EQ.0 ) GO TO 400
IT=NRSC(IR)
J=NRSC(IK)
DO 410 III=1, NR
LINEQ(III)=NETCNF(IR+IK+III)
410 CONTINUE
CALL TCOSTJ(LINEQ,II,J,J,LNKCHW,LNKCLN)
DO 501 JJ=1, N3
JJ=JJ+1
DO 510 JS=1, N4
DIMENSION LINDJ(NP3),LINDJ(NP3)
MINCT=0
RTFJ=TRJ(J,J)-SAPPJ, REMOVE TRAFFIC FROM J TO I
RTFJ=TRJ(J,J)-SAPPJ, REMOVE TRAFFIC FROM J TO I
DO 500 II=1,NR
IF(II.EQ.Ii.OR.II.EQ.JI) GO TO 500
IF(II.R.TTI+.EG.0.) OR, TRF(J,J), GO TO 508
501 RETURN
502 CALL LINTK(SUMCST)
503 IF(SUMCST.GT.MINCST) GO TO 120
504 DO 207 NN=1,N3
505 CONTINUE
506 CALL LINTK(SUMCST)
507 CONTINUE
MINCST = SUMCST

C

C RESERT TO INITIAL NETWORK CONFIGURATION FOR NEXT TRY

C

DO 250 NN= 1,N3

NETCF(I+1,NN)= NETCF(I+1,NN) + LIMDF(NN)

NETCF(1+1,NN)= NETCF(1+1,NN) + LIMDF(NN)

NETCF(J+1,NN)= NETCF(J+1,NN) + LIMDF(J,NN)

NETCF(I+J,NN)= NETCF(I+J,NN) + LIMDF(J,NN)

250 CONTINUE

500 CONTINUE

TRR(I+IAD)= TRR(I+IAD) + RTFJ

TRR(I+IAD+J)= TRR(I+IAD+J) + RTFJ

TRR(J+IAD)= TRR(J+IAD) + RTFJ

TRR(IAD+1)= TRR(IAD+1) + RTFJ

RETURN

C SUBROUTINE NETUP(IFLIP*IAD,I,J)

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

C

C UPDATE THE INTERREGIONAL NETWORK WHEN THERE IS SOME SAVINGS

C

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

C IF (IFLIP*FO,J) GO TO 700

C

C UPDATE THE NETWORK TRAFFIC MATRIX AND

C UPDATE THE OPTIMAL INTERREGIONAL NETWORK

C

DO 99 NN= 1,N3

ORINET(I+IAD+NN)-ORINET(I+IAD+NN) + LIMAD(NN)

ORINET(I+IAD+1,NN)=ORINET(I+IAD+1,NN) + LIMAD(NN)

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

ORINET(IAD+1+J,NN)=ORINET(IAD+1+J,NN) + LIMAD(J,NN)

99 CONTINUE

700 CONTINUE

DO 701 NN= 1,N3

ORINET(I+J+NN)=0

ORINET(I+J+1,NN)=0

701 CONTINUE

C RESET TRAFFIC MATRIX TR(NR*NR)

C

DO 900 IR= 1,NR

DO 910 IK= 1,NR

TRR(IR*IK)= TRR(IR*IK)

900 CONTINUE

910 CONTINUE

C UPDATE TOTAL COST FOR OVERALL NETWORK

C

C ORICST= ORICST-MAXSAV

C

C UPDATE NLINK MATRIX

C

NLINK[I]=NLINK[I]-1

NLINK[I]=NLINK[I]-1

RETURN

END

APRT STACOM,ICOSTJ/0777

A-40
77-53, Vol. IV

5192*STACOM(1), ICOST/J/0777

C SUBROUTINE ICSTJ(LINFQ,U+J,J,TKCHW,LNKCLH)

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

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

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

PARAMETER NP1=130,NP2=1,NP3=4,NP4=3,NPC=360
PARAMETER NP7=4
DIMENSION LINFQ(NP3), LKCHW(NP3,NP3), LNKCLH(NP3,NP3)
COMMON/LINHR,LINMIX(NP3)*LINCAP(NP3)+ITIL2Z(NP3)
*/COST/NI:N2*N3*N4*N7+NCTY
*/COST/AINSTCF(NP2,NP3,NP4)+NP5*NP3,NP4,2)
AMSTL(NP2,NP3)*NP3,2)+ RECLNL(NP2,NP3,16)+IMPFLX(NP3)
*/INF/INRFL(NP2,NP3)+IRAND(NP2,NP3)+1FAGL(NP2,NP3)
*/AND/ IAND(NP1)*KCHG*KAND TERMINALS WITH SAME W-H
*//RF/RFEF(NP1)+TRAFD(NP1,NP2,NP7),DISTCEF(NP6),MAPADR(NM1)

C INITIALIZATION

II=MAPADR(I)
J=MAPADR(J)
IANDT=IAND(J)
DO 1 100 N+1=3
DO 20 100 NM+1=2
LKCHW(NL+NM)=0
DO 30 100 NK=1+3
LKCHW(NL+NM)=0

CONTINUE

100 KRATE= IRAND(I)+1, N RATE STRUCTURE TYPE FOR NODE I
KRAF= IRAND(J)+1, N RATE STRUCTURE TYPE FOR NODE J
KDENS= IRAND(I)+2, N TRAFFIC DENSITY TYPE FOR NODE I
KDENS= IRAND(J)+2, N TRAFFIC DENSITY TYPE FOR NODE J
KDNSFY= KDNSF4*KDKNSJ, N ACTUAL DENSITYANT, 13=1H AND 2L=1L)
K=KDNSFTY+1, 03=H, 23=H AND 15=L
DST = DIST(I,J), N DISTANCE BETWEEN NODES I AND J

INSTOPT

ITIP=1, N PRIMARY COST FOR H/W UNIT
IF(DST+LFE. 0.5) ITIP=2, N ONSCOMT COST FOR ADDITIONAL UNIT
K = IARTF(KRATE,KRATF), N ACTUAL RATE STRUCTURE TO BE USED
DO 40 1 IL= 1+3
INPX = IDUPLX(IL), N DUPLEXING MODF 1H AND 2F
NOV = LINCFLU(IL), N NUMBER OF LINES REQUIRED
NOV=NOV*1NORDN=KAND

C CALCULATE COSTS FOR NON-LINE TYPE CHARGES

IF (NOV.EQ.0) GO TO 1 N NO LIPS ARE REQUIRED
DO 2 IV=1+N4, N HIGH DENSITY RATE

C INSTALLATION COSTS FOR NON-LINE TYPE CHARGES

LKCHW (IL+IV+1) = AINSTC(KP,IL,IV,KP,INPX,ITIP)*KCHG*NOV
ANNUAL RECURRING COSTS FOR NON-LINE TYPE CHARGES

1.

\[ \text{LNKCLN} = \text{RECRIC} \times \text{KCHG} \times \text{NDV} \]

2. CONTINUE

CALCULATE LINE INSTALLATION COST

IF IFLAG = 0 LINEAR, IF I = 1 NONLINEAR OTHERWISE

ANNUAL LINE INSTALLATION COST

\[ \text{LNKCLN} = \text{ANSTM} \times \text{KCHG} \times \text{NDV} \]

IF IFLAG < 1 GO TO 41

LINEAR LINE RECURRING COST FUNCTION

\[ \text{BN} = \text{DST} \times \text{RECRCLN} \times \text{KCHG} \times \text{NDV} \]

IF DST < 0 GO TO 32

GO TO 32

CONTINUE

NONLINEAR LINE RECURRING FUNCTION

\[ \text{COST} = \text{RECRCLN} \times \text{DST} \times \text{KCHG} \times \text{NDV} \]

IF DST < 0 GT GO TO 51

GO TO 32

CONTINUE

\[ \text{DST} = \text{DST} \times \text{KCHG} \times \text{NDV} \]

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

RETURN

END
51928*STACOM(1), RHOFUN/0777
1 SUBROUTINE RHOFUN(T1, T2, LIMEQU, LNLMT, RHOLIN, RH0)
2 C ****************************************
3 C CALCULATE LINE UTILIZATION
4 C T1 = LINT TO SWITCHER TRAFFIC
5 C T2 = SWITCHER TO LINT TRAFFIC
6 C LNLMT = HIGHEST LINE TYPE
7 C LIMEQU = LINE CONFIGURATION
8 C ****************************************
9 C
10 PARAMETER NP3=4
11 COMMON/LINCHR, LINLIM(NP3), LINCAP(NP3), UTILIZ(NP3)
12 /
13 * SUM/N1*N2*N3*N4*N5+N6+N7+N8+N9+N10
14 * SUM/ASUM(4)+RSUM
15 * /XMT/ TIMXMT(+NP3)+WAIT(6)
16 * /AMSL/ AMSL(7)
17 DIMENSION LIMEQU(1), RHOLIN(1)
18 RH0=0
19 CAP=0
20 DO 8 N=1, N3
21 CAP=CAP+LIMEQU(N)*LINCAP(N)
22 8 CONTINUE
23 CN=LINCAP(LNLMT)/CAP. ASUMIZATION FACTOR
24 X5AC1=CN*T2*ASUM(3)/(RSUM*AMSL(5)*R1)
25 X5AC2=CN
26 IF(RHOLIN(6) .EQ. 0.0) GOTO 101
27 X5AC2=CN*T2*ASUM(4)/(RSUM*AMSL(6)*R1)
28 120 CONTINUE
29 X5AC3=CN*T1/(AMSL(4)*R1) ASUMIZATION FACTOR
30 RHOLIN(1)=X5AC1+T1*TIMXMT(5+LNLMT)
31 RHOLIN(2)=X5AC2+T1*TIMXMT(6+LNLMT)
32 RHOLIN(3)=X5AC3+T1*TIMXMT(4+LNLMT)
33 RH0=RHOLIN(1)+RHOLIN(2)+RHOLIN(3)
34 RETURN
35 END

GPR7 STACOM.LINNUM/0777
SUBROUTINE LINNUM(T1,T2,LINEQU,LNLMT,JFLAG,RHO)

C FIND LINE CONFIGURATION BASED ON THE GIVEN TRAFFIC AND
C APPLICABLE LINE TYPE
C JFLAG= 1 FOR MULTIDROP LINE CASE
C T1 = LINE TO SWITCHER TRAFFIC
C T2 = SWITCHER TO LINE TRAFFIC
C
PARAMETER NPS=4
COMMON/LINCHR/ LINMIX(NPS),LINCAP(NPS),UTILIZ(NPS)
COMMON/CONST/ N1,N2,N3,N4,N7,N8,N9+CITY
COMMON/WSLA, AMPL(7)
INTEGER TRAF
DIMENSION LINEQU(1),RHLIN(3)
TRAF=T1+T2
DO 1 I=1,N3
LINEQU(I)=0
1 CONTINUE
LNLMT=0
CALL REFER
CALL LNLMTU=LNLMT
CALL SUBROUTINE LINNUM(1)
CALL SETU INITIAL LINE CONFIGURATION
C
IF(JFLAG .EQ. 1) GOTO 10
3 CONTINUE
LINEQU(LNLMT)=LINEQU(LNLMT)+1
LNLMT=LNLMT+UTILIZ(LNLMT)
IF(AMPL(LNLMT) .GT. TRAF) GOTO 7
TRAF=TRAF-AMPL(LNLMT)
CALL REFER
GOTO 3
10 CONTINUE
LINEQU(LNLMT)=TRAF/(LINCAP(LNLMT)*UTILIZ(LNLMT)) + 1
7 CONTINUE
70 CONTINUE
CALL RHOUNIT(T1,T2,LINEQU,LNLMTU,RHLIN,RHO)
IF(RHO .LT. UTILIZ(LNLMT)) GOTO 15
IF(LNLMTU .LT. N3) GOTO 72
IF(JFLAG .NE. 1) GOTO 73
LINEQU(N3)=LINEQU(N3)+10 NEED TO BE MODIFIED
GOTO 70
73 CONTINUE
DO 2 N=1,N3
IF(LINEQU(N) .NE. 0) GOTO 20
2 CONTINUE
48 CONTINUE
20 CONTINUE
NL=N+1
NL=NL+1
IF(LINMIX(NL) .EQ. 0) GOTO 22
LINEQU(NL)=1
LINEQU(NL)=LINEQU(NL)+1
GOTO 22
57    74  GOTO 70
58    74  CONTINUE
59    74  LINEGI11)=1
60    74  GOTO 70
61    74  CONTINUE
62    74  0NLMT=0NLMT+1
63    74  IF(LINMIX(LNLMTU)+FG,0) GOTO 79
64    74  LINEG(LNLMTU)=1
65    74  GOTO 79
66    74  CONTINUE
67    74  150  0NLMT=0NLMTU
68    74  RETURN
69    74  SUBROUTINE REFER
70    74  C
71    74  C FIND THE UPPER LIMIT OF LINE TYPE ALLOWED
72    74  C
73    74  DO 14 N=1,N3
74    74  LTRAF=TRAF/UTILIZ(NN)+0.5
75    74  IF(LINMIX(NN)+FG,0) GOTO 14
76    74  LNLMT=NN
77    74  IF(LINCF(NN)+GT. LTRAF) GOTO 15
78    74  14  CONTINUE
79    74  15  CONTINUE
80    74  RETURN
81    74  END

OPRT STACOM=PACK/0777
51928\#STACOM\$(1)\#PACK/0777

1 COMPILER (FLD=ABS)
2 SUBROUTINE PACK(I,K,L,IA)
3
4 ***************
5
6 C L=1 FOR STORING AND L=2 FOR RETRIEVAL
7 C K= DISTANCE DATA CONCERNED
8 C
9 ***************
10 DIMENSION IA(1)
11 IQ=(I-1)/4 THE WORD LOCATION
12 IR=I-I04 THE QUARTER CONCERNED
13 IQ=IQ+1
14 IS=(IR-1)*Q
15 IF(L=L01) GOTO 10
16 C
17 C RETRIEVE IT(9 BITS) BEGINNING AT IS-TH BIT OF THE IQ-TH WORD
18 C
19 K=FLD(IS9,IA(IQ))
20 RETURN
21 C
22 C STORE IT(9 BITS) BEGINNING AT IS-TH BIT OF THE IQ-TH WORD
23 C
24 10 CONTINUE
25 FLD(IS9,IA(IQ))=K
26 RETURN
27 END

GPRY STACOM\#DIST/0777
FUNCTION DIST(I,J)

PARAMETER NP1=130+NPC=360
PARAMETER NP6=(NPC*NPC/2-NPC+1)/4+1
PARAMETER NP7=4
COMMON /REF/REF(NPC),TRAFF(NP1+2,NP7),
* DSTNCE(NP6),MAPADR(NP1)
INTEGER DSTNCE
DST=0.
IF(I.EQ.J) RETURN
II=MAPADR(I)  ACTUAL CITY INDXX
JJ=MAPADR(J)
IF(I.EQ.JJ) RETURN
I,J,L=LINK(I,J,J)
CALL PACK(I,JL,IDIST+2,DSTNCE)
DIST=IDIST
IF(DIST.NE. 511) RETURN
DIST=RECOVR(I,JL)
RETURN
END
FUNCTION LINK(J,K)
C
***************
C
C FIND THE RELATIVE LOCATION FOR (J,K) COMBINATION
C WHICH IS THEN USED FOR FINDING DISTANCE BETWEEN SYSTEM
C TERMINATIONS J AND K
C
***************
C
PARAMETER NP1=130+NPC=360
PARAMETER NP6=(NPC+NPC/2-NPC+1)/4+1
PARAMETER NP7=4,NP3=4
COMMON /CONST/ N1,N2,N3,N4,N7,NCITY
1 /REF/REF(NP1)+TRAFF(NP1+2,NP7),DISTCF(NPA),MAP44R(NP1)
INTEGER DISTANCE
LINK=0
10 IF(J.GT.NCITY.OR.K.GT.NCITY.OR.K.FO.J) RETURN
11 JJ=J
12 KK=K
13 IF(J,JL,T,K) GOTO 1
14 JJ=K
15 KK=J
16 CONTINUE
17 LINK=J-1+NCITY + KK = IREF(JJ)
18 CONTINUE
19 RETURN
20 END
FUNCTION RECOVR(I)

C RETRIEVE OVERFLOW DISTANCE DATA FROM IVRD

C

PARAMETER NPC=360, NPO=10, NPC
COMMON /OVER/ IVRD(NPO+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', 1X, 'FOR LOCAL INDEX', 216)
15 STOP
16 20 CONTINUE
17 RECOV=IVRD(N+2)
18 RETURN
19 END

&PRY L,PLOTPT/0777
SUBROUTINE PLOTPT(L1, L3)

C SUBROUTINE FOR MOVING CALCOMP PEN WITH OR WITHOUT PEN DOWN
C L3=1 FOR MOVING WITH PEN DOWN
C =2 FOR MOVING WITH PEN UP
C =3 FOR CLOSING THE PLOTTING
C
PARAMETER NPC=360
COMMON/VH/VERT(NPC), HORIZ(NPC)
DIMENSION IBUF(1000)
DATA IP/0/ GFLAG FOR PLOTS CALL DATA X/1,2566/
DATA Y/0/ IF(IP .NE. 0) GOTO 50
CALL PLOTS

50 CONTINUE
IF(L3 .EQ. 3) GOTO 100 GROUND IS TO BE CLOSED
AV=VERT(L1)
AH=HORIZ(L1)
BV=AV*COS(X)+AH*SIN(X)
BH=AV*SIN(X)-AH*COS(X)
BF=(BH+BV)/301.
IF(L3 .EQ. 2) GOTO 80
CALL SYMBOL(BH+BV+0.025*INT(BF), INT(BF)) OPEN IS DOWN
80 CONTINUE
CALL PLOT(RH,BV+3) OPEN IS UP
IP=1 GFLAGS CALL IS NOT NEEDED ANY MORE
RETURN
100 CONTINUE
CALL PLOT(X,0) RETURN
END

GPRP L.RSPNSE/0777
SUBROUTINE RSPNSE(T1,T2,LINTYP,N,ICK)
C ***************
C CALCULATE MEAN RESPONSE TIME FOR THE PROPOSED MULTIDROP LINE
C
C WAIT(6)=TIMED DELAYS DUE TO
C 1=WAIT FOR POLLING  2=WAIT FOR I/O  3=INPUT XMT TIME
C 4=CPU TURNAROUND  5=OUTPUT QUE/WAIT  6=OUTPUT XMT TIME
C
PARAMETER NPS=4
COMMON/RESP/ RHOLIN(4),RSPTIM
1 /XMT/ TIMXMT(7,NP3), WATT(6)
2 /BOUND/ NTERMS,TIMEDP,PROC,MPLOT
3 /CONST/N1,N2,N3,N4,N7,N7TY
DIMENSION LDUMMY(NP3)
DO 10 N=1,N3
LDUMMY(N)=0
10 CONTINUE
LDUMMY(LINTYP)=1
I0K=0
C
CALL RHOFUN(T1+T2,LDUMMY,LINTYP,RHOLIN,PHN)
RHOLIN(4)=1.0-RHO
IF(RHOLIN(4) .LE. 0.) RETURN
25 WAIT(1)=TIMXMT(1+LINTYP)+TIMXMT(2+LINTYP)*(M-1/2)
26 WAIT(2)=(1.-RHOLIN(4))*TIMXMT(7+LINTYP)/(1-RHOLIN(1)-
27 RHOLIN(2))
28 WAIT(3)=TIMXMT(3+LINTYP)
29 WAIT(5)=(1.-RHOLIN(4))*TIMXMT(7+LINTYP)/(1-RHOLIN(1))
30 WAIT(6)=TIMXMT(5+LINTYP)
31 RSPTIM=0.
32 DD 11 J=1,6
33 RSPTIM=WAIT(J)+RSPTIM
34 CONTINUE
35 IF(RSPTIM .GT. TIMREO) RETURN
36 I0K=1
37 RETURN
38 END
QPRTY L*INPUT/0777
<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><strong>Term</strong></td>
<td><strong>Definition</strong></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>STAtate Criminal Justice COMmunication Project</td>
</tr>
<tr>
<td>Star Network</td>
<td>A communication network where each system termination is directly connected to the central data processing system</td>
</tr>
<tr>
<td>SUP-Time</td>
<td>A run time estimate by the EXEC-8 accounting subsystem which accounts for the amount of time spent by a run on usage of CPU, I/O processing and execution of system control statements and executive requests</td>
</tr>
<tr>
<td>System Termination</td>
<td>A logical node in the communication system under the STACOM program, which consists of one or more physical terminals</td>
</tr>
<tr>
<td>TELPAK</td>
<td>A specific tariff for a telecommunication network</td>
</tr>
<tr>
<td>Terminal</td>
<td>A device that allows users of a data processing system to gain access to that system in a more convenient manner than the input/output devices local to that system</td>
</tr>
<tr>
<td>Terminal Response Time</td>
<td>The duration from the time a user initiates a request for network service at the terminal to the time he receives a complete response</td>
</tr>
<tr>
<td>Tree</td>
<td>A graph which has a root node without any predecessors and other nodes have unique predecessors</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UNIVAC</td>
<td>UNIVersal Automatic Computer, a computer trade name by Sperry Rand Corporation</td>
</tr>
<tr>
<td>Vertical Horizontal (V-H)</td>
<td>A pair of numbers which are designated by AT&amp;T for cities and used for the purpose of calculating distance between any two cities</td>
</tr>
</tbody>
</table>