

EFFICIENT ALLOCATION OF RESOURCES IN
CENTRALIZED COMPUTER-COMMUNICATION NETWORK DESIGN

Dixon R. Doll

The University of Michigan

This document is subject to special
export controls and each transmittal
to foreign governments or foreign na-
tionals may be made only with prior
approval of RADC (EMBIH), GAFB,
N. Y. 13440.

en 87

UMR0971

FOREWORD

This report was prepared by the Systems Engineering Laboratory of the University of Michigan, Ann Arbor, Michigan, under Contracts AF30(602)-3953 and F30602-69-C-0214, Project 5581, Task 558102 for Rome Air Development Center, Griffiss Air Force Base, New York. Secondary report number is SEL-TR-36. Messrs. L. W. Odell (EMIDA) and R. Iuorno (EMBIH) served as RADC project engineers for the respective contracts noted above.

The report covers accomplishments in the development of mathematical models for use in designing efficient centralized computer-communication networks. The primary results consist of computational algorithms which can be implemented on existing digital computers to configure new remote-access, real time computer systems. These techniques provide significant new tools for the analysis of large-scale computer system design aspects.

This report was also presented by the author in partial fulfillment of requirements for the degree of Doctor of Philosophy at the University of Michigan, 1969. The author wishes to express his appreciation to Professors Keki B. Irani, Arch W. Naylor, Bruce W. Arden, Norman R. Scott and Jack M. Miller.

Distribution of this report is restricted under the provisions of the U.S. Mutual Security Acts of 1949.

This technical report has been reviewed and is approved.

Approved:



ROCCO F. IUORNO
Information Transfer Sciences Section

Approved:



CARLO P. CROCETTI
Chief, Info Processing Branch
Information Sciences Division

ABSTRACT

The objective of this research is to formulate and solve a variety of resource allocation problems which arise in the design of centralized computer-communication networks. Since a large portion of the capital outlays in the teleprocessing systems are frequently devoted to the remote communications facilities, these problems are at least as important as those concerned with efficiently utilizing the facilities at the central location. Realistic analytic models of the telecommunication network behavior are developed and used to solve these various design problems with algorithms that are as computationally efficient as is possible. In some cases, it is necessary to consider procedures which obtain solutions that are reasonable approximations to the true optima in order to assure tractability.

These efforts commence with a discussion of the important macroscopic level design variables and the design tradeoffs which must be considered in centralized networks. A relatively simple-to-work-with, yet realistic performance measure, average response time, is precisely defined. The assumptions used to develop the problem solving models are then presented, justified, and shown to generally provide a greater level of model detail than existing approaches.

Then, queueing models are formulated for the various types of data concentrators. An optimal passage capacity assignment proce-

cedure is developed for determining channel capacities in multiplexed communication links so as to obtain the minimum average message delay throughout the entire network. An iterative application of Lagrange multiplier techniques involving constraints is the basic mathematical method used for the optimization procedure. Simulation studies reveal the efficacy of making the infinite-buffer assumption for MSC concentrators.

As a prelude to solving the general network design problem, a rigorous analysis of existing design procedures is undertaken and realistic cost functions are established. It is then seen that usage of these approaches generally leads to an inefficient utilization of the link capacity resource since it is not treated as a variable. Methods are presented for optimally allocating link capacity in a network so as to minimize line costs and satisfy a performance constraint.

All of these previous efforts are finally integrated and used to develop a comprehensive, yet realistic algorithm for designing centralized computer-communication networks. The proposed procedure can be effectively used in n terminal networks as long as n^3 sets of design calculations can be performed on the computer in a reasonable length of time.

To cite the usefulness and limitations of the design procedures, and to compare them with other methods, a variety of representative case studies are also considered. In none of these representative

examples do any of the other design procedures obtain better solutions than those obtained using techniques proposed in this research.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	iii
LIST OF TABLES	x
LIST OF ILLUSTRATIONS	xi
 <u>Chapter</u>	
I INTRODUCTION	1
1.1 Background	1
1.2 Fundamental Concepts, Terminology	4
1.3 A Technique for the Generalized Description of a Telecommunication Network	12
1.3.1 Point-to-Point Segments Character- ization	15
1.3.2 Multipoint Segment Characterization	19
1.3.3 Example System Description Using Method Presented	21
1.4 Computer-Communication Networks - A Closer Look	29
1.4.1 Comparison with General Purpose Communication Nets	29
1.4.2 Distributed and Centralized Types of Teleprocessing Network Structures	31
1.5 State of the Art	36
1.5.1 The Evolution of Communication Network Analysis Techniques	36
1.5.2 General Techniques for Determining System Performance	42
1.5.3 Centralized Computer-Communication Network Design	43
1.6 Objectives of this Effort	47
1.7 Contributions	49
II A DETAILED EXAMINATION OF THE PROBLEM AREA	52
2.1 Important Design Variables and Tradeoffs, Motivation	52
2.2 Definitions of Response Time, Notational, Important Queueing Theorems	62

TABLE OF CONTENTS (Continued)

<u>Chapter</u>	<u>Page</u>
2.2.1 Forward and Reverse Networks	62
2.2.2 Notational	65
2.2.3 Important Queueing Theorems and Definitions of Response Time	68
2.3 Precise Problem Statement, Assumptions	77
2.3.1 Discussion of Assumptions, Empirical Data	81
III ANALYSIS OF VARIOUS TYPES OF DATA CONCENTRATORS	86
3.0 Introduction	86
3.1 Various Methods of Data Concentration	87
3.2 The Message Switching Concentrator (MSC)	90
3.2.1 Model for Message Switching Center, Forward Network	91
3.2.2 Model for Message Switching Center, Reverse Network	100
3.3 The Multiplexing Concentrator (MX)	101
3.3.1 Optimal Passage Capacity Assignment Procedure, Single Stage Case	107
3.3.2 Optimal Passage Capacity Assignment Procedure, Tandem Multiplexor Case	123
3.4 Procedure for the Computation of Minimum Average Response Time in Networks with MX and MSC Concentrators	135
3.5 The Way Station Concentrator	140
3.6 Telephone Type of Concentrator	144
3.7 Comparative Performance Considerations	151
3.7.1 Message Switching Centers vs. Multiplexors in the Single Concentrator Case	152
IV COST FUNCTIONS AND EXISTING NETWORK DESIGN PROCEDURES	167
4.1 Introduction	167
4.2 Cost Functions for Link Capacity and Data Concentrators	168
4.2.1 Link Capacity Cost Function	169
4.2.2 Data Concentrator Cost Functions	169
4.3 The Esau Williams Algorithm	177
4.4 The Minimum Spanning Tree Algorithm	186

TABLE OF CONTENTS (Continued)

<u>Chapter</u>		<u>Page</u>
V	EFFICIENT UTILIZATION OF THE LINK CAPACITY RESOURCE	188
5.1	Introduction	188
5.2	Excess Link Capacity Removal Procedures which Preserve a Given Average Response Time	190
5.3	Link Capacity Variation Procedures which Allow Average Response Time to Vary within Constrained Limits	193
5.3.1	An Example Solution Using Backtrack Programming	200
5.3.2	Computational Requirements and Summarization	214
5.4	Implications of Variable Capacity Values with Respect to Existing Design Procedures	218
VI	NETWORK DESIGN PROCEDURES IN THE GENERAL CASE	220
6.1	Introduction	220
6.2	Network Design When Only One Link Capacity Value is Available	223
6.2.1	How the Data Concentrator Type is Varied	225
6.2.2	The Design Technique	235
6.2.3	Number of Structures Considered by the Procedure; Combinatorial Implications	248
6.3	The Approach to be Taken in the General Network Design Problem	253
6.3.1	The Multidimensional Design Parameter Space	257
6.3.2	Dimensionality of the Global Design Parameter Space	262
6.4	The Computational Modules	263
6.5	Network Design Procedure when Topology, Concentrator Type, and Link Capacity are Variables	265
6.5.1	The Basic Algorithm	267
6.5.2	Concluding Discussion	270
6.6	A Critical Evaluation of the Procedure; Extensions to the Basic Algorithm	271

TABLE OF CONTENTS (Continued)

<u>Chapter</u>		<u>Page</u>
VII	APPLICATIONS - SEVERAL REPRESENTATIVE CASE STUDIES	276
7.1	Introduction	276
7.2	Alternative Design Procedures to be Considered	278
7.2.1	Minimum Spanning Tree Algorithm with Excess Capacity Eliminated	278
7.2.2	Esau-Williams Algorithm with Excess Capacity Eliminated	279
7.2.3	Other Strategies	279
7.3	An Eight Node Network Design Example	280
7.3.1	Discussion of Results	287
7.4	A Twenty City Case Study	294
7.4.1	Discussion of Results	301
7.5	A Highly Clustered Case Study	305
7.5.1	Discussion of Results	308
7.5.2	A Modification of this Problem	312
VIII	SUMMARY	314
8.1	Recapitulation	314
8.2	Future Research	317
	BIBLIOGRAPHY	319
	APPENDIX A	326

LIST OF TABLES

<u>Table</u>	<u>Page</u>	
3-A	Results of GPSS Simulation of the Finite Buffer Capacity MSC for a Line Speed of 450 b/s	96
3-B	Results of GPSS Simulation of the Finite Buffer Capacity MSC for a Line Speed of 900 b/s	97
5-A	The Optimal Link Capacity Assignment for the Network of Figure 5.3 when $T_{\max} = .25$	213
7-A	The Alternative Design Strategies Used in the Comparative Studies	281
7-B	The Arrival Statistics for the Example of Section 7.3	282
7-C	Mileage Matrix for the Example of Section 7.3	282
7-D	Cost-Capacity Relationship for a Unit Length Link	283
7-E	Partial Experimental Results for the Example of Section 7.3	284
7-F	The Balance of the Experimental Results for the Case Study of Section 7.3	285
7-G	The Optimal Passage Capacity Assignments for the Links of Structure in Figure 7.2(a)	293
7-H	Mileage Matrix Used for Example of Section 7.4	295
7-I	Experimental Results for the Case Study of Section 7.4	298
7-J	Experimental Results for the Case Study of Section 7.4	299
7-K	Mileage Matrix for Example of Section 7.5	306
7-L	The Partial Experimental Results for the Clustered Example of Section 7.5	307
7-M	The Improved Solution for the Design Problem of Section 7.5 when $T_{\max} = 2.0$	309

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1. 1	Schematic Diagram of Teleprocessing System	2
1. 2	A System with One Point-to-Point and One Multipoint Segment	4
1. 3	A System Having One Multipoint Segment	10
1. 4	A System with One Multipoint Segment and More Sophisticated Terminals	11
1. 5	A Descriptive Format for a System Having P Point-to-Point and M Multipoint Segments	14
1. 6	Example Description of System with Four Point-to-Point Segments	16
1. 7	Format for Description of Point-to-Point Segment	20
1. 8	Format for Description of Multipoint Segment	22
1. 9	A System Having 1 Multipoint and 2 Point-to-Point Segments	23
1. 10	Descriptive Format for System of Figure 1. 9	24
1. 11	Centralized Teleprocessing Network Structure	33
1. 12	Distributed Teleprocessing Network Structure	34
2. 1	An Eleven Segment Point-to-Point Network	54
2. 2	A Six Segment Multipoint Network	55
2. 3	Schematic Diagram of the Typical Exponential Service Node in Networks under Consideration in Jackson's Theorem	70
2. 4	A Multiserver System	72
2. 5	Poisson Traffic Merging	73
2. 6	Poisson Traffic Partitioning	74

LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>		<u>Page</u>
3. 1	Schematic Diagram for the Single Stage MSC, Forward Network	91
3. 2	Queueing Model for the Single Stage MSC Concentrator	92
3. 3	Schematic Diagram of Model Used for Simulation of the MSC Concentrator	99
3. 4	Schematic Diagram for the Single Stage MSC, Reverse Network	100
3. 5	A Single Stage Multiplexor	102
3. 6	Tandem Multiplexor Configuration	103
3. 7	Queueing Model for Network of Figure 3.6	105
3. 8	The Single Stage Multiplexor Schematic	107
3. 9	Generalized Tandem Multiplexor Schematic	124
3. 10	Example Network	131
3. 11	A Way Station Concentrator	140
3. 12	Way Station Queueing Model for n Remote Terminals	143
3. 13	Queueing Model for Telephone Concentrator	146
3. 14	Queueing Times for Accessing and Using a Trunk Line in the Telephone Concentrator	150
3. 15	A Generalized Schematic for Single Stage Concentration	155
3. 16	Comparative Study #1 of MX and MSC Concentrators	158
3. 17	Comparative Study #2 of MX and MSC Concentrators	159
3. 18	Comparative Study #3 of MX and MSC Concentrators	160

LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>		<u>Page</u>
3. 19	Comparative Study #4 of MX and MSC Concentrators	161
3. 20	Comparative Study #5 of MX and MSC Concentrators	162
3. 21	Comparative Study #6 of MX and MSC Concentrators	163
3. 22	Comparative Study #7 of MX and MSC Concentrators	164
3. 23	Comparative Study #8 of MX and MSC Concentrators	165
4. 1	Cost vs. Capacity Function in a Unit Length Link	170
4. 2	Cost of Multiplexor Concentrators	173
4. 3	Some Example Configurations	174
4. 4	MSC Cost Function	176
4. 5	An Example Illustration of an Iteration in the Esau Williams Design Algorithm	179
4. 6	An Example Configuration	181
4. 7	Variation of Topological Structure Using the Esau Williams Design Algorithm	183
5. 1	Network for the Example of Section 3. 3. 2. 2	191
5. 2	Illustration of Link Capacity Variation Procedure	197
5. 3	An Example Network	201
6. 1	Variation of Topological Structure	226
6. 2	Generation of New Sequence of Configurations by the Excess-Capacity Trimming Operation	258
6. 3	The Space of Design Parameter Variations Resulting from the Reduction of Link Capacity in the Structures of the Sequence of Figure 6.1 When These Were Obtained Using the Capacity Value, Φ_k	260

LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>		<u>Page</u>
6.4	Graphical Representation of the Subspaces, $\mathcal{S}(\Phi_k), k = 1, \dots, N$	261
6.5	Graphical Representation of the Global Design Parameter Space \mathcal{S} where $\mathcal{S} = \bigcup_{k=1}^N \mathcal{S}(\Phi_k)$	261
7.1	The Experimental Results for the Case Study of Section 7.3	286
7.2	The Solutions Obtained by the Unmodified Design Procedures of this Thesis for the Case Study of Section 7.3	289
7.3	The Generation of Topological Structures in a Typical Situation	290
7.4	The Solution Configurations for the Case Study of Section 7.4	296
7.5	The Experimental Results for the Case Study of Section 7.4	300
7.6	Solution Configurations for Clustered Terminal Case Study, Obtained Using Unmodified Procedure of Section 6.5	304
7.7	The Experimental Results for Clustered Terminal Case Study of Section 7.5	310
7.8	The Improved Solution for the Design Problem of Section 7.5 when $T_{\max} = 2.0$	311
7.9	The Solution for the Modified Design Problem of Section 7.5.1 when $T_{\max} = .5$	313

Chapter I

INTRODUCTION

1.1 Background

Complexes of digital computer installations and communication facilities which connect remotely located terminals to one or more of the computational centers constitute a teleprocessing system.

In such systems, the configuration of the communication network for connecting remote terminals to the central facilities constitutes an important design problem which will be considered in this work.

The most significant distinction between a teleprocessing system and a conventional batched input system lies in the random manner in which requests for service (jobs) arrive at various entry points to the system. In the conventional batch mode, arrival patterns are deterministic in the sense that they are monitored by machine operators and computing center scheduling personnel. A very significant fraction of capital expenditures in a teleprocessing system is devoted to communications facilities outside of the processing centers; in the batch mode almost all system capital expenditures are devoted to software and hardware facilities at the central computational complex.

In the design of such teleprocessing systems, various queueing problems arise as consequences of the random or unscheduled requests for service and the necessity of effectively utilizing the expensive communications equipment which links the remote terminals to the

computational facilities. This research is centered around the development of techniques for the analysis and synthesis of the communications network portion of the teleprocessing system.

Figure 1.1, a schematic diagram of a teleprocessing system with one processing center depicts the important functional distinctions between the various components of a teleprocessing system.

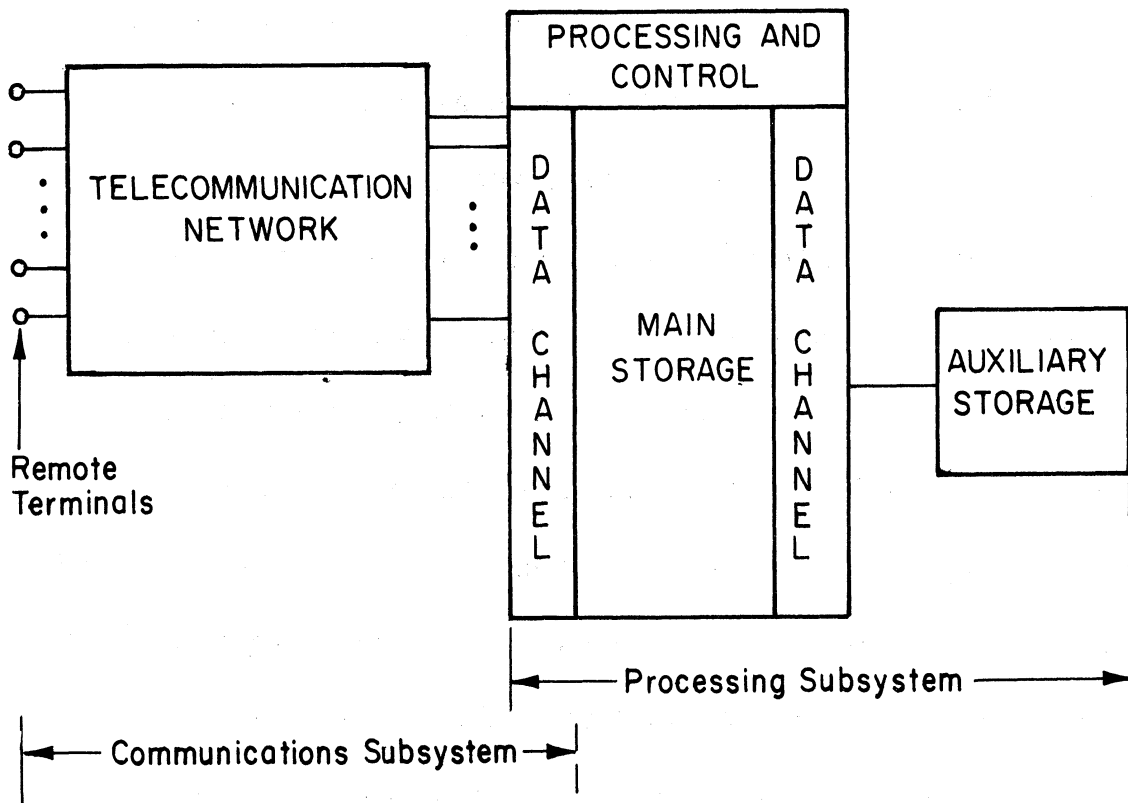


Figure 1.1 Schematic Diagram of Teleprocessing System

For the purposes of clarification, a system can be conceptualized in terms of two subsystems, one for communications and the other for processing functions. Processing subsystems may be considered to be composed of arithmetic unit(s), control circuitry, primary (or main) storage, auxiliary storage, and data channels for connecting input lines to main storage and main storage to auxiliary storage.

On the other hand, the communications subsystem will consist of the various remote terminals, communications channels, data concentrators, and interfacial facilities whose duty it is to assemble and provide the data to the processing subsystem in recognizable form. Interactions between the two subsystems are monitored by an interrupt facility which is common to both subsystems and controlled by a supervisor or executive control facility. A portion of main storage called the data channel serves as the region in which message traffic is assembled, controlled, and buffered for transmission between user and main storage. A similar region exists to expedite the flow of traffic between main and auxiliary storage. Auxiliary storage functions to preserve files, tables, program segments, and messages in various stages of processing.

One may validly conjecture that the increasing complexity and cost of the communications subsystem has necessitated that as much importance be placed on this portion of the system as on the processing centers. Efforts concerning the design and utilization of processing subsystems have been numerous and relatively fruitful over the last decade. The practical significance of these advances, however, is diminished if the resultant input-output limitations are not resolved; the motivation for equally vigorous activity concerning the telecommunications subsystem is apparent.

1.2 Fundamental Concepts, Terminology

An an embarkation point for the detailed examination of the communications portion of teleprocessing networks, certain fundamental concepts and terminology must be established. The teleprocessing network, taken in its entirety, constitutes a System. A System consists of at least one System Segment which is either a Point-to-Point Segment or a Multipoint Segment. Within a Multipoint Segment there exist more than one possible Information Paths between data sources and data sinks. For clarification of these terms, consider Figure 1.2, which illustrates a System having each type of System Segment.

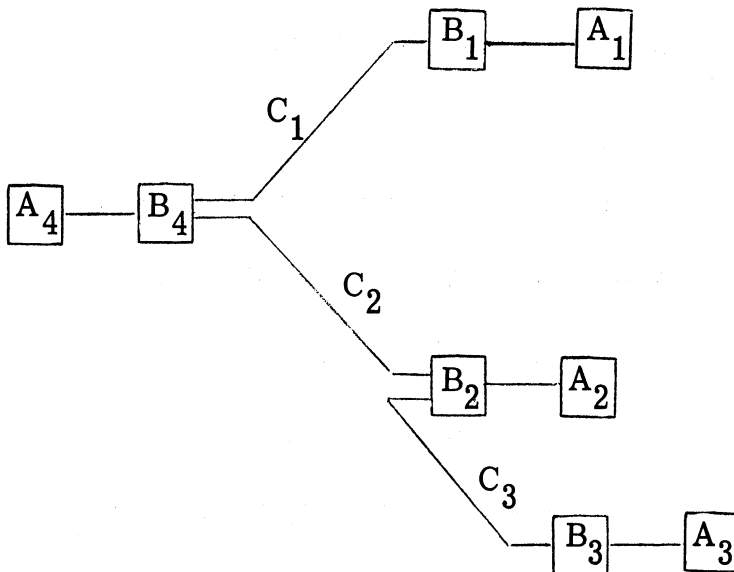


Figure 1.2 A System with One Point-to-Point and One Multipoint Segment

It consists of data sources A₁, A₂, A₃; communications interface-control units (IU's) B₁, B₂, B₃, B₄; data sink A₄; and communication links C₁, C₂, C₃. It has two System Segments:

1. A_1, B_1, C_1, B_4, A_4
2. $A_3, B_3, C_3, A_2, B_2, C_2, B_4, A_4$.

System Segment (1) is called a Point-to-Point Segment and System Segment (2) is called a Multipoint Segment. Within the Multipoint Segment, two of the possible Information Paths which may exist are:

1. $A_3, B_3, C_3, B_2, C_2, B_4, A_4$
2. A_2, B_2, C_2, B_4, A_4 .

The actual information paths within a Multipoint Segment are determined by the operating discipline of the system. The communication links C_1, C_2, C_3 are terminated at either end in signal converters which are parts of the appropriate IU's.

In Point-to-Point Segments of a System, there is an equivalence between a Channel and a communication link with allied signal converters at each end of the link. For example, in Figure 1.2, the configuration B_1, C_1, B_4 constitutes a Channel. In Multipoint System Segments, however, Channels are formed by dedicated portions of adjacent communication links and intermediate IU's such that information may be transmitted over the link components of the Channel without entering a queue. For example, in the Multipoint Segment of Figure 1.2, if information being transmitted from A_3 to A_4 can be queued at B_2 , then there would be an equivalence between terminated links and Channels. If however B_2 functions instantaneously (without delay) to multiplex data from link C_3 and source A_2 onto link C_2 , then

the two channels would consist of the following components.

For Source A_3 : B_3, C_3, B_2, C_2, B_4

For Source A_2 : B_2, C_2, B_4

Note that the two channels have system components B_2, C_2, B_4 in common.

On the shared link C_2 a fraction of the total link bandwidth (transmission capacity) is devoted to each source. This partitioning of link bandwidth results in the formation of Passages on this type of shared link.

A Passage conceptually refers to the fractions of the total link transmission capacity which are assigned to the terminals sharing the link. Thus a multiplexed link being shared by k terminals would have k Passages, each one of which is dedicated to a particular terminal. The Channels for Sources A_2 and A_3 in the example are denoted:

Channel for Source A_3 : $B_3, C_3, B_2, C_2(A_3), B_4$

Channel for Source A_2 : $B_2, C_2(A_2), B_4$

where $C_2(A_3)$ refers to the Passage on link C_2 which serves Terminal A_3 and $C_2(A_2)$ refers to the Passage on link C_2 servicing Terminal A_2 .

Thus Information Paths are composed of chains of links interlaced with appropriate IU's which provide a data transmission capability between source and sink terminals. Channels are composed of chains of Passages interlaced with appropriate IU's; messages are queued at the source and the sink for the Channel.

The hardware configuration at each end of a Channel when lumped together will be called a Terminal or Node. The boundaries of a

particular Terminal are defined in terms of the maximal collection of equipment whose components are internally interconnected solely by links on which direct transmission of signals is used.¹ Stated another way, a Terminal is formed by a closed boundary containing the largest collection of equipment possible such that all internal communication connections (those with both ends inside the boundary) use direct signal transmission. The Terminal may be simple in configuration and consist of a single device, such as teletypewriter or a line printer, or it may be a complex array of equipment consisting of a wide variety of interconnected peripherals and processors. Every Terminal always contains at least one IU and may include combinations of other modules called End Units (EU's) and Central Processing Units (CPU's); concise definitions of EU, IU, and CPU are now given.

End Unit (EU):

An EU is any device in a digital data transmission system which serves as the ultimate source or sink for information transfer. It may possess the capability of initiating or reacting to various end-to-end control procedures such as carriage return or line feed on a teletypewriter, but it does not include features for the execution

¹ Such internal links are often referred to as limited distance since the maximum allowable length of such a link is limited by signal attenuation properties of the link. In general, while the attenuation depends primarily on the quality of wire or cable used, even the best quality link is not capable of reliable transmission at distances in excess of several miles. At distances exceeding several hundred feet it is usually necessary to use coaxial cable to obtain satisfactory performance.

or control of communications procedures.

Interface-Control Unit (IU):

An IU is a device located within a Terminal which functions, as part of an Information Path, to execute and control communications procedures. It may contain modulation and/or demodulation hardware for signal conversion, and may perform the function of code conversion, error control, multiplexing, and temporary buffering or storage of data. An IU may be a discrete item of equipment or form an integral part of an EU.

Central Processor Unit (CPU):

The non-interactive portion of a Terminal which cannot be identified as either an IU or an EU. It is characterized by its ability to execute stored programs, access bulk data storage, etc., and perform functions typical of a general purpose digital computer under executive system control. Those hardware portions of a Terminal serving strictly for I/O are not included as part of the CPU, nor is the IU which serves as the interface between communication links and EU's in a Terminal.

A hierarchy of basic structural modules which comprise a teleprocessing network have now been presented. In order to complete the establishment of concepts and terminology which are necessary for mathematical modelling purposes, several measures of performance must now be introduced. These are transfer rate of information bits (TRIB), residual error rate, and response or service time.

Transfer Rate of Information Bits (TRIB):

A basic unit of throughput measurement for links or channels that expresses numerically the following ratio in bits per unit time. $\text{TRIB} \triangleq$ (Number of information bits accepted by the sink) divided by (total time required to accomplish the transport of all bits required to get those information bits accepted) and does not include uncorrected errors. TRIB is discussed at length in Section 1.3.1.

Residual Error Rate:

The ratio of the number of bits, characters, or message blocks incorrectly received but undetected or uncorrected by the error-control equipment, to the total number of bits, characters, or message blocks transmitted, respectively.

Response or Service Time:

The characteristics of system performance expressing the time delay which messages experience in undergoing transmission from source to sink. A precise, mathematical definition of response time will be given in Section 2.2.

The examples of Figure 1.3 and 1.4 further clarify the fundamental concepts and terminology presented in this section.

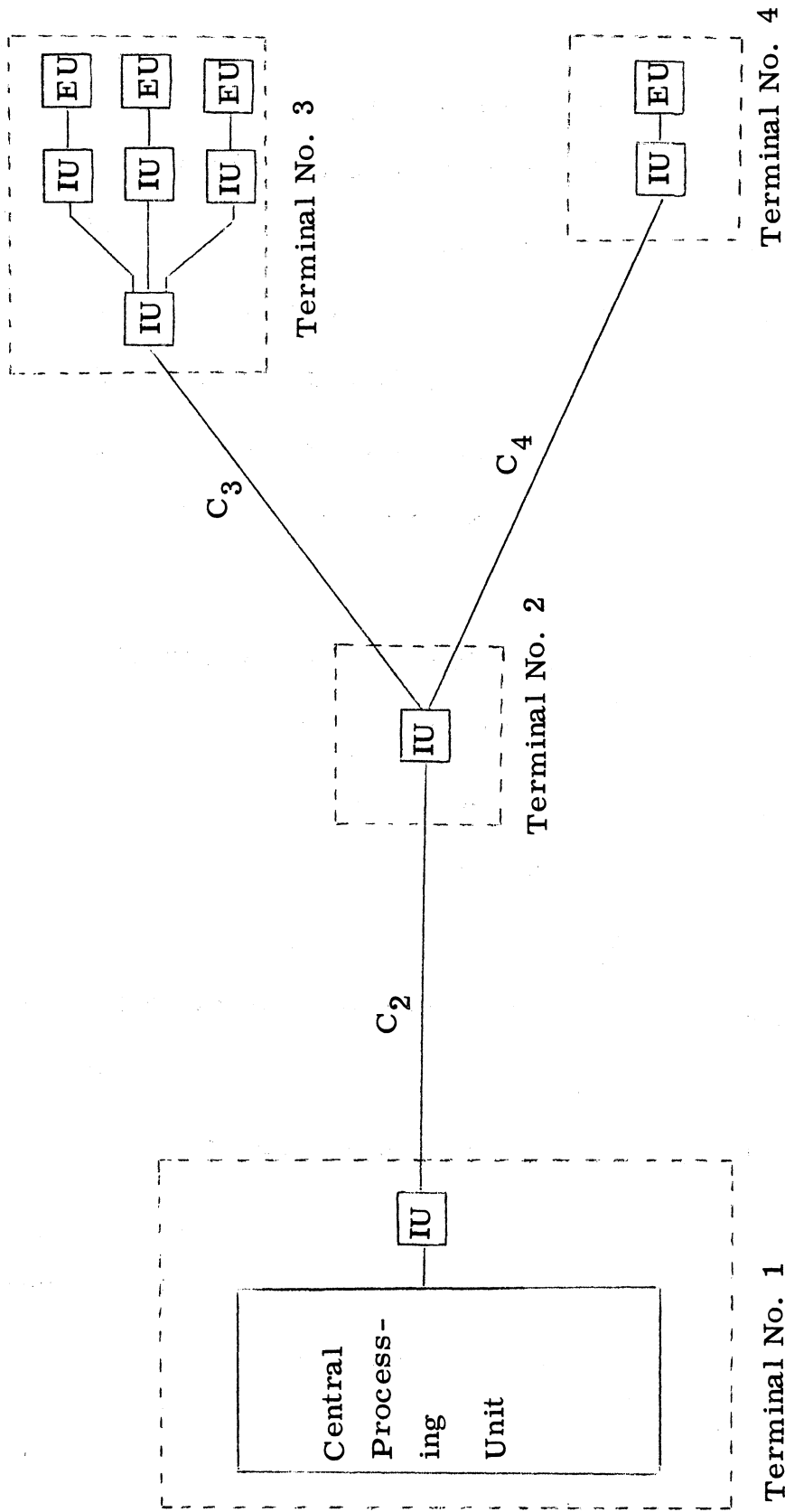


Figure 1.3: A System Having One Multipoint Segment

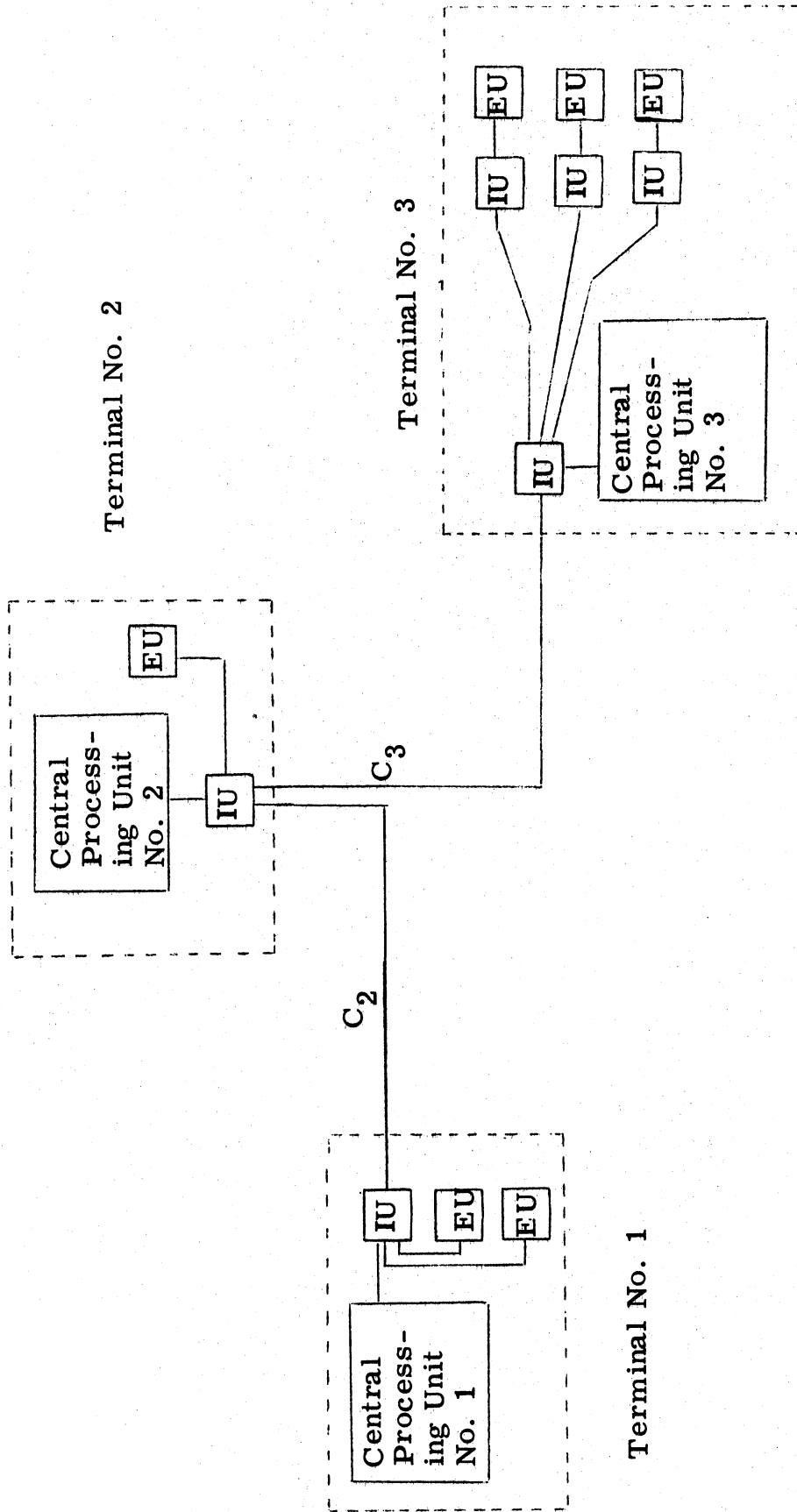


Figure 1. 4: A System with One Multipoint Segment and More Sophisticated Terminals

Terminal No. 2 of Figure 1.3 illustrates the most elementary form of a Terminal, one consisting of a single IU which concentrates message traffic from Terminals No. 3 and No. 4 onto link C_2 . No man machine interactions would take place at Terminal No. 2, since no EU exists.

On the other hand, Terminal No. 3 of Figure 1.4 illustrates a more sophisticated form. All EU's within Terminal No. 3 must by definition be linked directly (without any intermediate signal conversion provisions) to the IU which interfaces with CPU No. 3. Traffic emanating at one of these EU's in Terminal No. 3 and destined for Terminal No. 1 or Terminal No. 2 undergoes signal conversion at the IU interfacing Terminal No. 3 with link C_3 .

1.3 A Technique for the Generalized Description of a Telecommunication Network

Since the data transmission portion of a teleprocessing system is usually composed of widely varying types and makes of components, it is meaningful in establishing a basis for mathematical modelling techniques to develop a "common denominator" for identification and description of these systems. One finds a number of clearly recognizable functional characteristics and features which are common to all systems. Some such commonalities have been pointed out in the previous section, for example, point-to-point and multipoint segments; intermediate, end, and central processing units; communication links, channels, and passages, etc.. Other characteristics such as whether

a communication link is switched or dedicated, etc., have not yet been discussed.

By suitably identifying terminals and their interconnections, and eliminating unnecessary descriptive information, an orderly, hierarchical form of description can be developed which provides only that information which is pertinent to the evaluation of network performance and cost using mathematical modeling techniques.

In considering the fundamental elements which comprise the remotely accessed computing network, certain logical choices for classifying the various terminals present themselves. Either a remote terminal is directly connected to a terminal containing a central processing unit via a point-to-point arrangement, or an information path between source and sink which contains some shared links is used. In this latter situation the terminal constitutes a portion of a multipoint system segment. Since each terminal forms a portion of either a point-to-point or a multipoint segment, it is possible to completely describe the system, as shown in Figure 1.5, in terms of system segments, some of which are point-to-point and others which are multipoint.

The next step in the development of the descriptive format requires a more detailed examination of these types of system segments. Each point-to-point system segment consists of one information channel, a remote source terminal, and a terminal containing a CPU which serves as the sink. However it can happen that a terminal will desire access

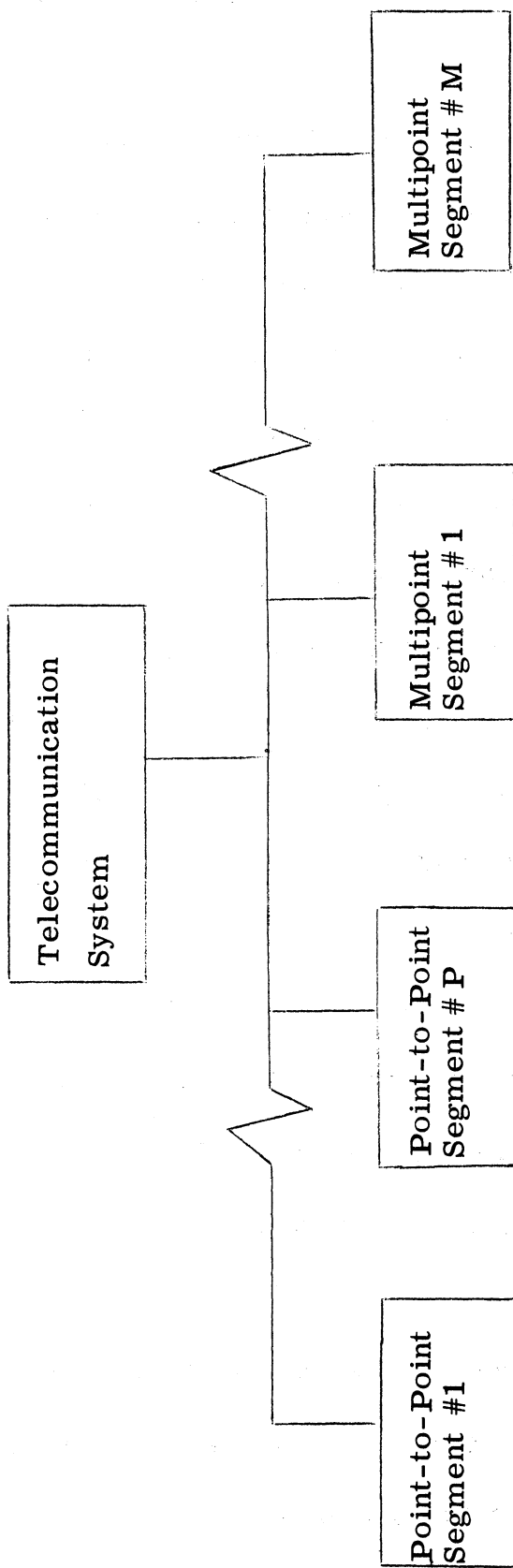


Figure 1. 5: A Descriptive Format for a System Having P Point-to-Point and M Multipoint Segments

to more than one sink terminal. Thus the same terminal will be the source in as many point-to-point system segments as there are different CPU's to which it is connected. As an illustration consider Terminal 2 (T2) in Figure 1.6(a). There are four point-to-point segments which comprise the entire system but T2 is a source terminal in three distinct point-to-point segments. Thus the description of the system in Figure 1.6(a) would be indicated as shown in Figure 1.6(b).

But the simple decomposition of the system into system segments does not provide as much detail as is necessary for mathematical characterization of the network. Hence, it is logical to proceed by further characterizing each of the two types of system segments in terms of their relevant parameters.

1.3.1 Point-to-Point Segments Characterization

Point-to-point segments are by definition, composed of two terminals connected by a single channel. In most every meaningful application at least one of these terminals will contain a central processing unit, since the purpose of the system segment is to provide a communication path between remote source terminal and computer. Thus the coarsest possible resolution of every point-to-point segment must result in a pair of terminals and a channel.

One now asks the question, how can the terminals and the information channel further be classified. Since the functional components of terminals have already been discussed in detail, only the information

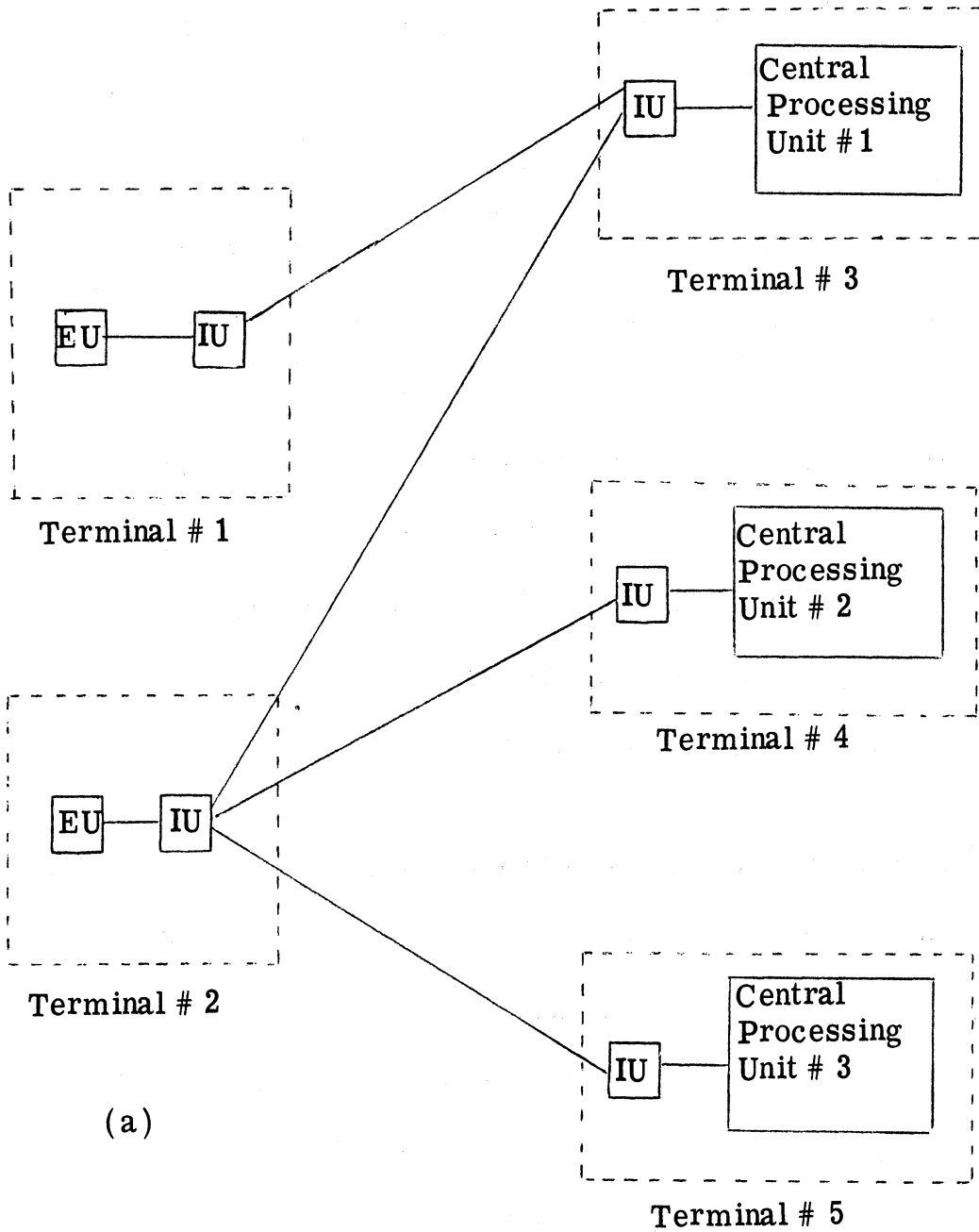
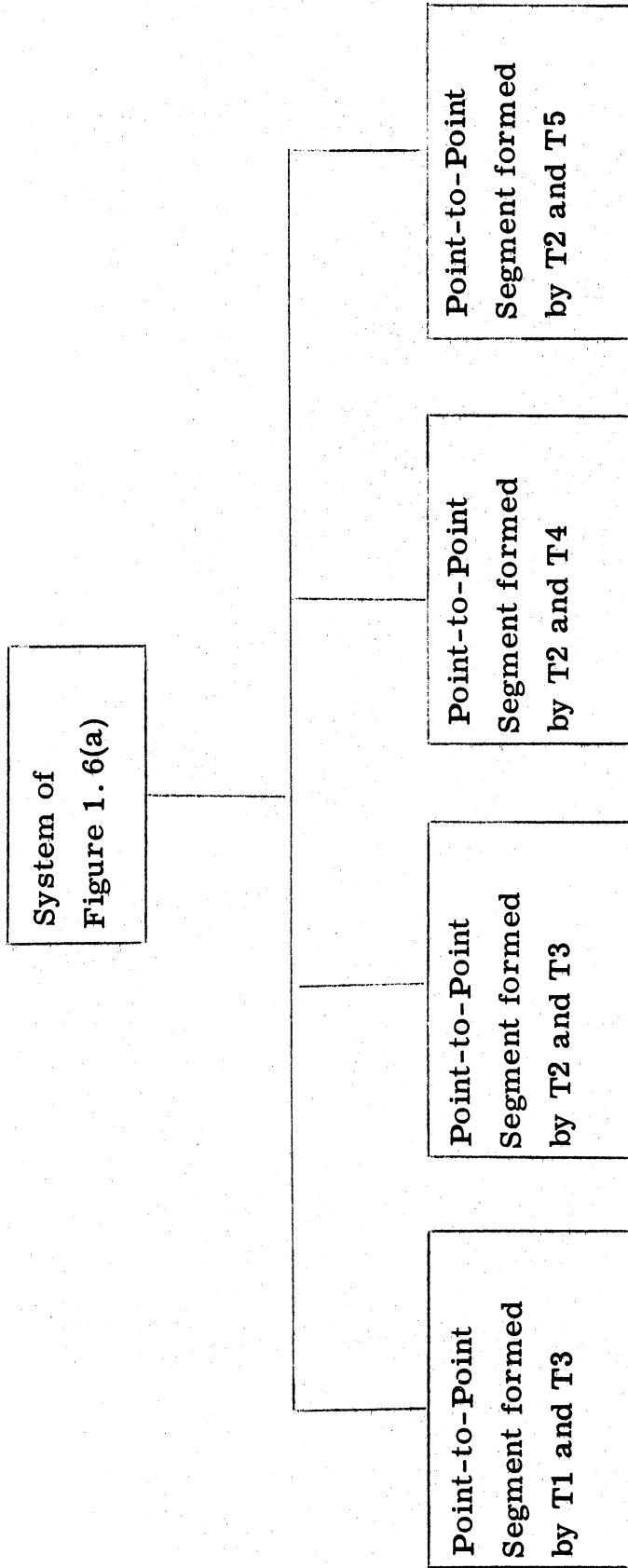


Figure 1.6 Example Description of System with Four Point-to-Point Segments



(b)

Figure 1.6 Example Description of System with Four Point-to-Point Segments

channel remains to be characterized.

In general there are two types of channels which should be considered, leased and switched channels.¹ Leased channels have the fixed cost per unit time characteristic and the leasor obtains a permanent, fixed physical information channel between the source and sink terminals. Switched channels, on the other hand, involve a non-permanent path which is created by the construction of the path through the presently existing lines, trunks, and switching terminals of networks such as those of the common carriers, Western Union and American Telephone and Telegraph. The cost of switched facilities may be determined either on a "pay as used" basis or on a fixed cost basis such as in the Wide Access Telephone Service (WATS) Facilities offered by the Bell System.

For either type of line, the mathematical parameters of interest will be the channel TRIB, channel reliability, and whether it is full or half duplex. Channel reliability procedures for offsetting the effects of noise often involve schemes for the retransmission of message blocks which contain errors. In such schemes, various error-detecting coding techniques are often used so that when an error is detected, a request for retransmission of the block is sent from the receiving terminal to the transmitting terminal. If noise in a channel causes

¹ Leased communication channels are also known as private or dedicated channels.

errors in a received message with probability q (not to be confused with the Residual Error Rate) then whenever an error is detected, an average number of $\frac{1}{1-q}$ ($= 1 + q + q^2 + \dots$) total transmissions of the message are required in order to receive a single message without error. The factor $1-q$ can be interpreted as an efficiency factor of a channel whose maximum physical capacity to transmit information is Φ bits/sec or char/sec. Thus $\text{TRIB} = \Phi(1-q)$, where the redundancy level of the channel is $\frac{q}{1-q}$. The error correction provision results in an effective reduction of channel throughput from Φ to $\Phi(1-q)$.

Similar mathematical models for relating error probabilities, maximum transmission rate, and channel TRIB are discussed by Kucera [1].

Thus a point-to-point segment connecting terminal having X_1 EU's, Y_1 IU's, and Z_1 CPU's to a terminal having X_2 EU's, Y_2 IU's, and Z_2 CPU's may be described in the manner of Figure 1.7 where the parameters for characterization of the three components are indicated below their respective blocks.

1.3.2 Multipoint Segment Characterization

Recall that a multipoint system segment is distinguished from a point-to-point segment by the existence of at least one shared link, which arises as a consequence of the multiplicity of information paths within the segment. Further discussions and illustrations of this distinction are presented in Section 2.1. In the point-to-point segment, only one information path exists and, consequently, no links are shared. The descriptive format for multipoint segments thus is obtained by the

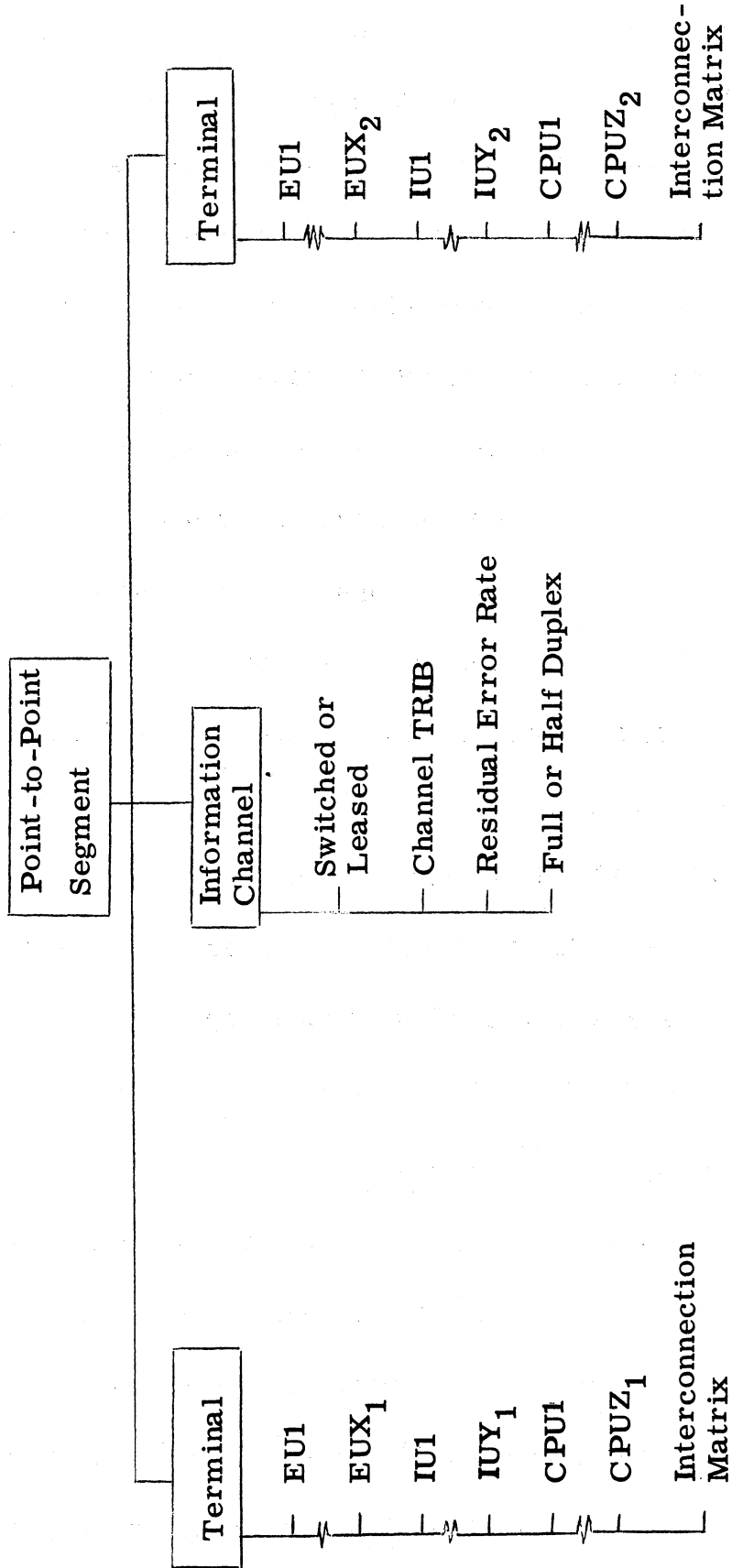


Figure 1.7 Format for Description of Point-to-Point Segment

specification of all information paths and associated source and sink terminals within an individual segment. At least one link in a multipoint segment is distinguished in the sense that it is shared; hence at least one terminal will also be common to two or more information paths within the segment. The format for the description of a multipoint segment is presented in Figure 1.8. Note that, in general, terminals having different indices in the diagram may actually be physically identical (e. g. the i -th terminal in the j -th information path may be the same terminal as the ℓ -th terminal in the m -th information path).

The descriptive format must also specify the particular characteristics of EU's, IU's, and CPU's in as much detail as is necessary to clearly identify the channels in the segment. This information is necessary for the purpose of constructing mathematical models of the multipoint system segment.

1.3.3 Example System Description Using Method Presented

Consider the system shown in Figure 1.9 which has two terminals which contain CPU's. Depending upon the application, either Terminal T6 or Terminal T1 or both might be considered the ultimate sink for the data transmission requests originating within the System. For illustrative purposes, assume that the CPU in T6 is an auxiliary CPU, dedicated to the end units within T6 but that the end units also have need for back-up processing support at T1; thus T1 serves as the sink for all source message traffic. The

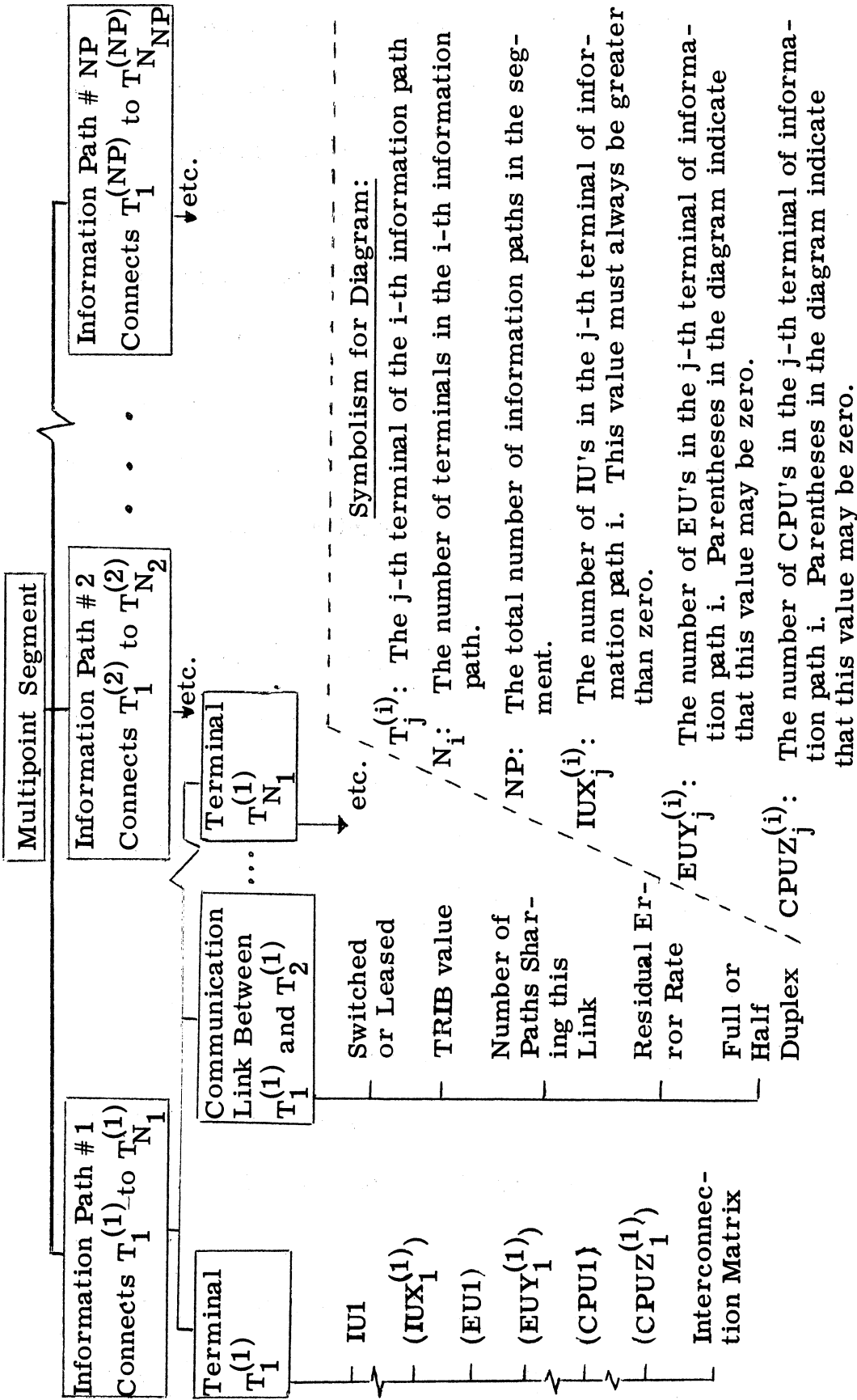
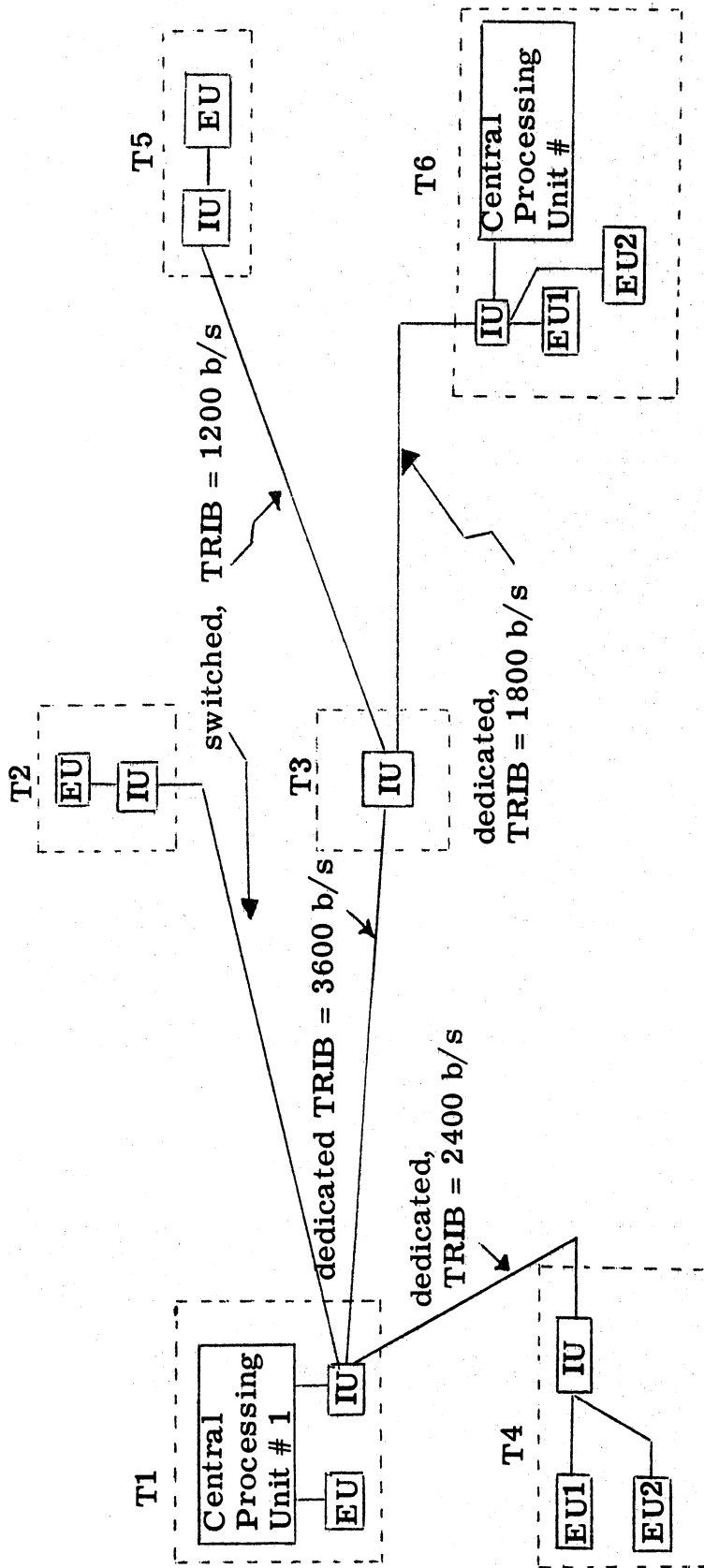


Figure 1.8 Format for Description of Multipoint Segment



Residual Error Rates: 10^{-4} /sec. on switched links

10^{-6} /sec. on dedicated links

All links are full duplex.

Figure 1.9 A System Having 1 Multipoint and 2 Point-to-Point Segments

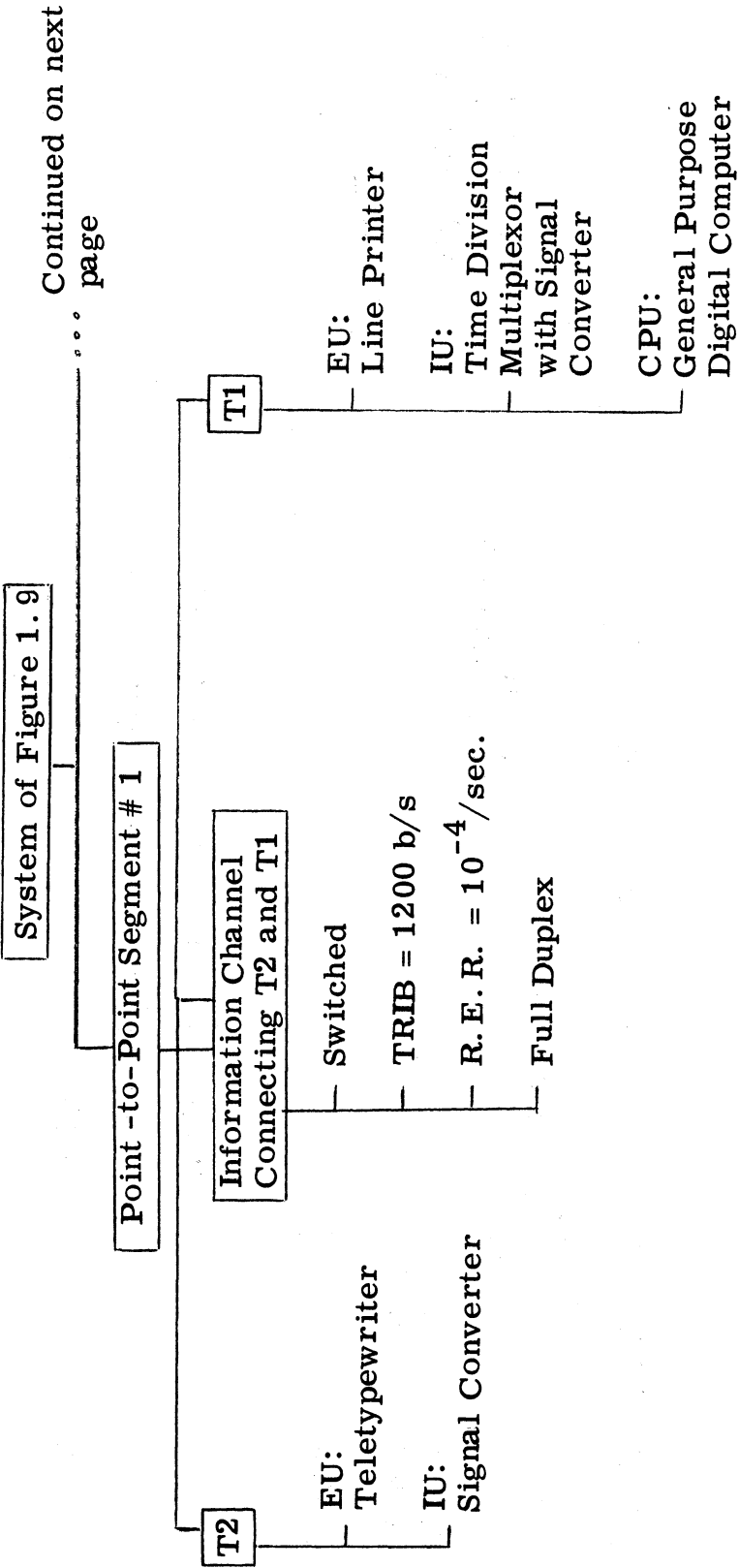


Figure 1.10 Descriptive Format for System of Figure 1.9

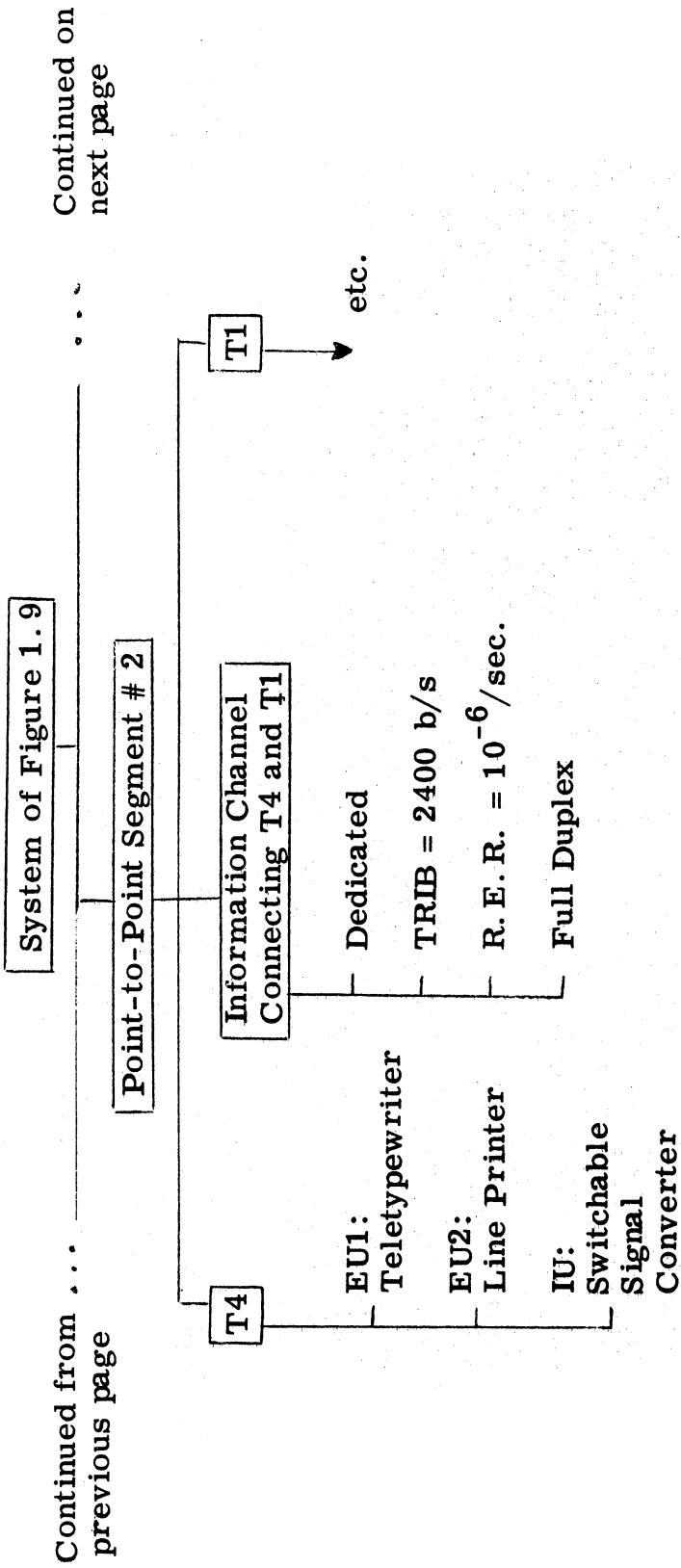


Figure 1.10 Descriptive Format for System of Figure 1.9

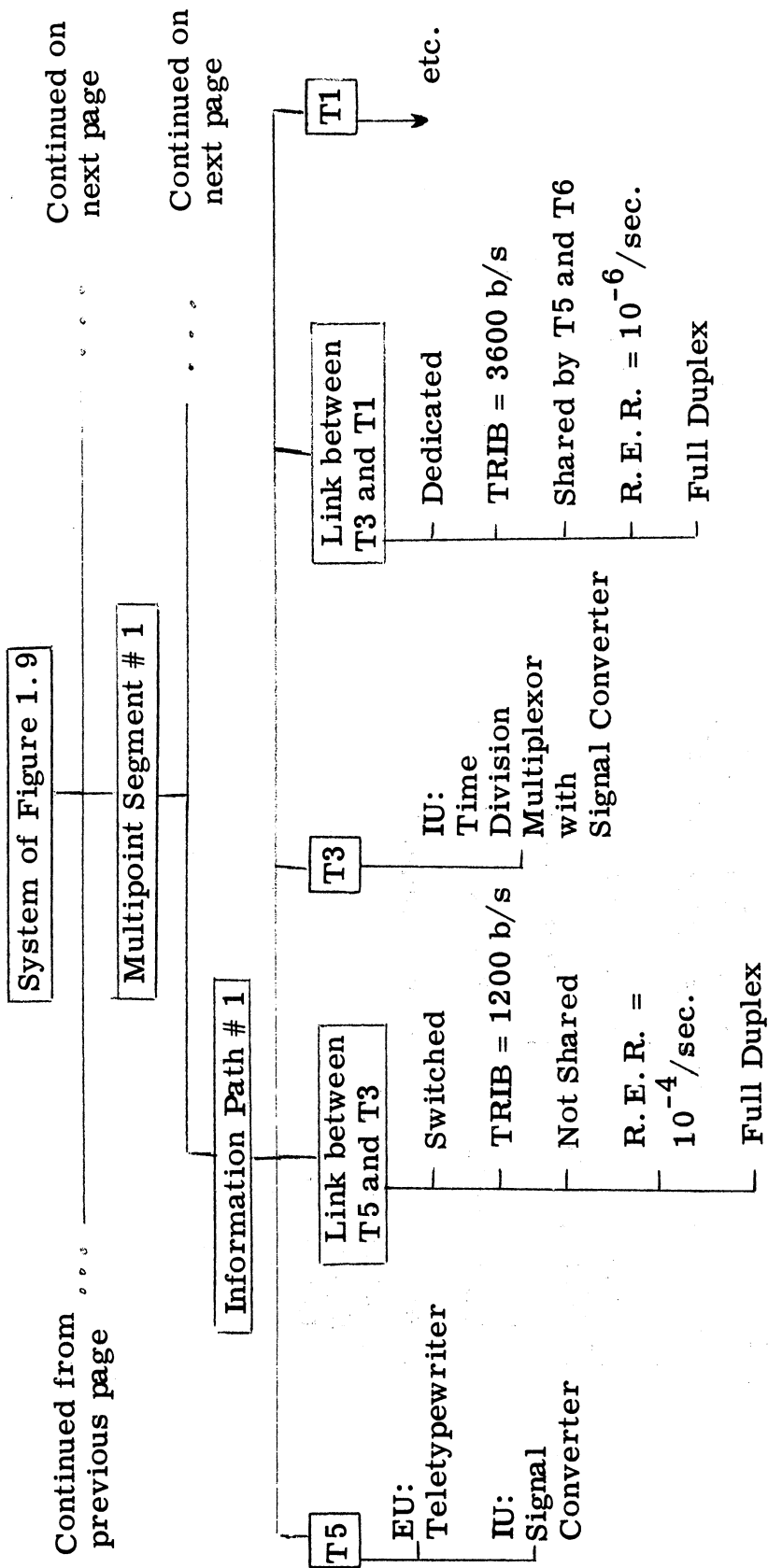


Figure 1.10 Descriptive Format for System of Figure 1.9

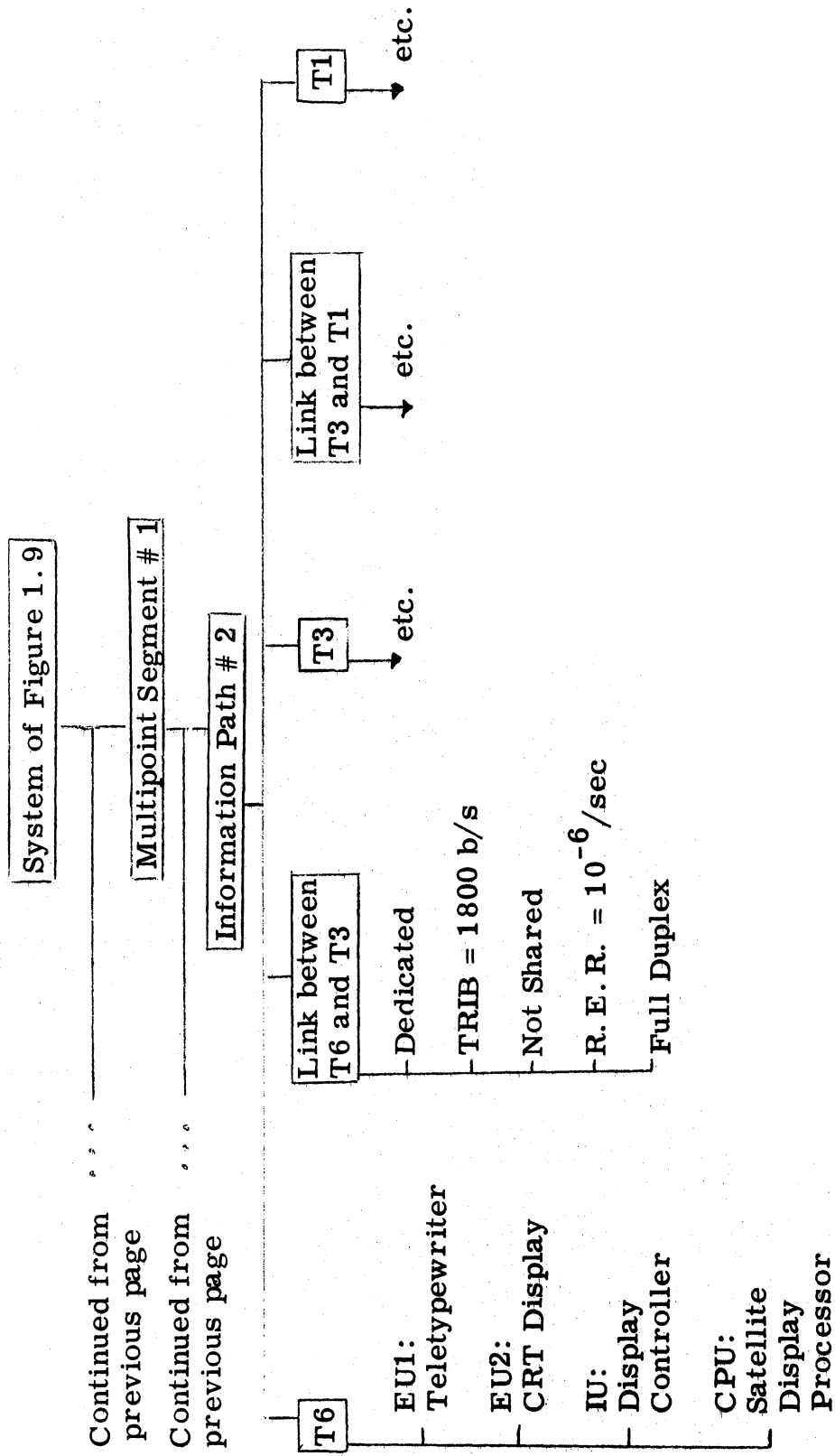


Figure 1.10 Descriptive Format for System of Figure 1.9

system contains two point-to-point segments and one multipoint segment. Within the multipoint segment there exist two information paths. T3, which serves only to concentrate requests from T5 and T6 onto the dedicated 3600 b/s link, is thus an example of a minimally structured terminal since every terminal must contain at least one IU for purposes of signal conversion. The block diagram descriptive format for the system of Figure 1.9 is given in Figure 1.10.

To illustrate further the concepts of the various components of a multipoint segment, note the manner in which the description of the system would be altered if T3 were to include an end unit. There would thus be an additional information path within the multipoint segment which would consist of the shared 3600 b/s link connecting T3 to T1.

It should be noted that implicit throughout the method of description there exists the notion of distinguished source and sink terminals. While the concept of source and sink functions really has no direct bearing on a description of a hardware configuration, such a functional classification of terminals is necessary for the development of any techniques pertaining to system design, since the functional roles played by the various terminals may alternate between source and sink.

At this point, it should be emphasized that the subject of tariff and regulatory statutes for the communications industry is one of the most complex in the entire utility field. Many authorities have suggested that an entirely new approach must be taken to the problems of communications tariffs and facilities. This thesis, while in no way attempting to minimize the nature of this dilemma, will be solely concerned with resolving the tradeoffs among important design variables from a mathematically objective standpoint.

1.4 Computer-Communication Networks—A Closer Look

1.4.1 Comparison with General Purpose Communication Nets

The primary objective of a general purpose communication network is to enable terminals or subscribers to the system to communicate with each other. In the general communication nets such as the telephone and telegraph networks of the common carriers, a pronounced degree of homogeneity exists in the properties which characterize both source and sink terminals, e. g. reaction time, total assimilation capacity etc.. Furthermore, subscribers seek to access the terminals of any of the other subscribers for the purpose of transmitting and/or receiving information. In the more special purpose environment of computer-communication networks there exist some important fundamental differences which can and will be exploited in this thesis for the purpose of extending the degree

of realism which can be incorporated into network design procedures.

The objective of teleprocessing (computer-communication) networks is to provide geographically dispersed users with access to remotely located data bases and/or computing power. As a consequence, there exist two clearly different types of nodes in the network, those which represent entry/exit points to the system and those which depict processing and data base sites. Usually, only a relatively small fraction of the terminals which originate message traffic are equipped with processing facilities, because the cost advantages of teleprocessing begin to disappear as the processing and/or data base functions become duplicated at multiple locations. In addition, there usually exists a significant speed differential between the rates at which remote stations and CPU's are designed to operate. Thus there exists a definite need for and ability to make use of concentration and deconcentration schemes in these specialized types of networks.

The efficient management of generalized communication systems also often calls for the use of concentration and deconcentration. However, these above-mentioned arguments favoring substantial concentration are more intuitively appealing in computer-communication networks as a consequence of the unique characteristics of the application - namely the speed imbalance between data sources and sinks and the existence of a relatively large number of remote terminals for each processing site. As a consequence of these characteristic differences between general purpose communication networks

and computer-communication networks, it is not surprising that the existing facilities of the common carriers are frequently considered to be inefficient for computer systems applications [2].

These functional distinctions having been noted, attention is now focused exclusively on the teleprocessing networks. As the initial effort in the detailed examination of such systems, a discussion of the different network structural forms is now presented.

1.4.2 Distributed and Centralized Types of Teleprocessing Network Structures

There exist primarily two functionally different organizations for the telecommunication networks of interest in this work, distributed and centralized configurations. In a distributed implementation such as the one shown in Figure 1.12, the computational activities are performed at more than one of the terminal sites; hence processors in the network may have occasion to transmit messages to other processor sites as well as to remote terminals. In the centralized configuration such as the one shown in Figure 1.11, all processing is performed at one central complex; hence, all remote terminals access that same central computational facility.

In the distributed implementation, as a consequence of the multiplicity of processing locations there exists two kinds of message traffic:

1. Traffic between processing sites and remote terminals, also called terminal traffic. (Traffic on the weakly shaded links of Figure 1.12.)
2. Traffic between processing sites. (Traffic on the heavily shaded links of Figure 1.12.)

However, networks having a centralized structure carry only terminal traffic since there is but a single processor.

In the distributed network structure, processors could be located at points where terminals are clustered and consequently offer the potential advantage of lower communications cost for terminal traffic. However it is generally acknowledged that relatively few distributed networks have been attempted thus far. What then are the considerations which have fostered the continuing development of centralized configurations?

J. B. Dennis [3] discusses a number of factors which have inhibited the development of distributed systems and consequently favored the centralized approaches:

1. For many applications there is clustering of terminals but little clustering of the data requests, that is, each data item is just as likely to be accessed by one cluster of terminals as by any other. This situation does not particularly favor the distributed approach because of the expense associated with maintaining duplicate copies of large data bases.

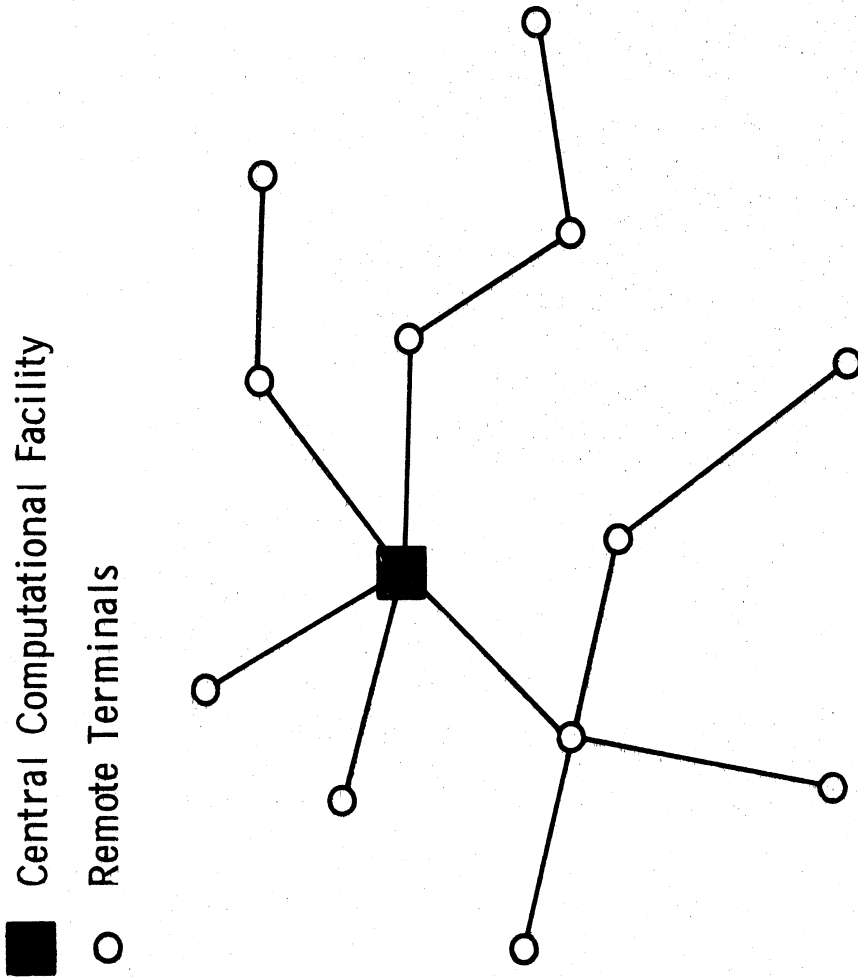


Figure 1.11 Centralized Teleprocessing Network Structure

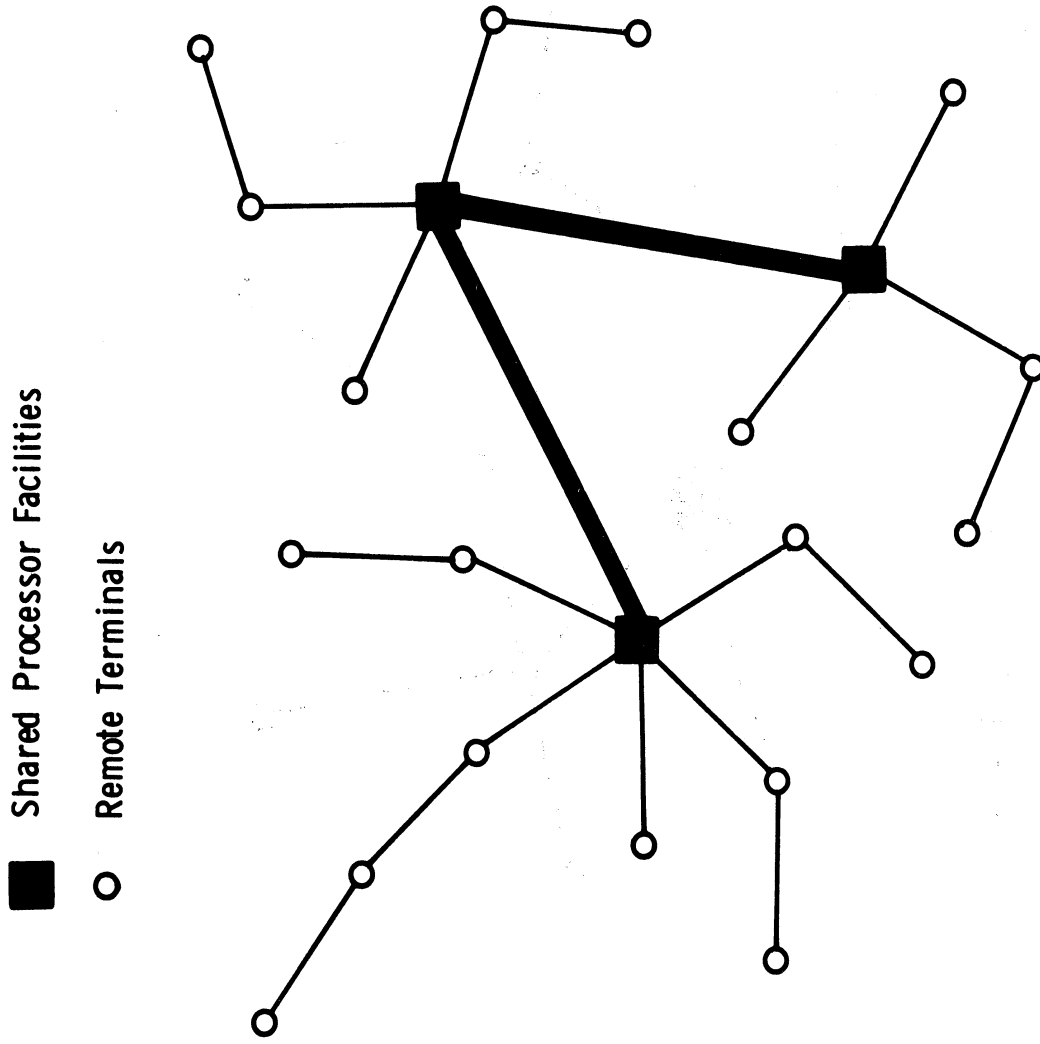


Figure 1.12 Distributed Teleprocessing Network Structure

2. Needed computational capacity has been most economically obtained using a single moderate-to-large sized installation.
3. Computer and program investment and maintenance requirements in distributed systems substantially exceed the cost of communications.
4. The management of internal (processor to processor) traffic and multiple data bases in the distributed network structure has been beyond the state-of-the-art.

Other factors have in the past and will continue in the future to impede any large scale transition to the distributed types of systems. For example, the politics of large corporations are such that the data processing manager can exert a greater influence and wield more authority when the data processing system is highly centralized. Clearly, both the design and operation of the distributed network represent significantly more complex tasks than is the case with the centralized structure. However, sophistication is not yet justifiable, as the basic economics of computer-communications continue to favor the centralized type of structure as being more cost effective. At some point in the future it may swing the other way, (if and when the aforementioned inhibiting factors tend to disappear) but this shift may, in fact, take several decades.¹

¹ See, for example, Chapter 6 in Parkhill [4].

When such a transition does finally materialize, it is likely to come about through the marriage of individual centralized systems (as has been suggested by Dennis, Parkhill, and others) which will then become constituent subsystems.

Without a thorough understanding of the design economics for centralized network structures, it is difficult to see any significant progress being made on the functionally more complex distributed processing arrangements. For these reasons, this research is devoted exclusively to the development of a design model for centralized configurations.

1.5 State of the Art

1.5.1 The Evolution of Communication Network Analysis Techniques

A proliferation of research efforts down through the years has been devoted to the analysis of delay in both circuit switched and message switched communication networks. The behavior of switching networks such as the ordinary telephone system has been the object of investigation of traffic engineers since the early twentieth century efforts of E. Johannsen and A. K. Erlang¹. Molina [6] and O'Dell [7] also made significant contributions to the early theory of queues. Since the unification of these fundamental works by T. C. Fry [8] in 1928 in his classic book on queueing problems,

¹ Reference to the works of E. Johannsen may be found in [5] which is a summary of Erlang's major contributions.

significant contributions to the analysis of congestion systems has been made by Feller [9,10], Kendall [11], Pollaczek [12], Morse [13], Khinchine [14], Syski [15], Benes [16], Saaty [17], Cox and Smith [18], Burke [19], and Riordan [20], to mention only a few. This literature concerning queueing theory has, along with the increasing use of digital computers as special purpose communications processors in a wide variety of communication networks, motivated the development of various analytic and simulation-oriented techniques for the modeling of such systems. A comprehensive bibliography of fifty-odd papers and technical reports which describe the major studies in the design and utilization of generalized communication networks since 1950 has been compiled by Pollack [21]. In addition to these, recent significant efforts have also been made by Ford and Fulkerson [22], Busacker and Saaty [23], Kim and Chein [24], and Kleinrock [25] in the form of books.

The problems discussed by Ford and Fulkerson [22] are formulated in terms of steady flow through a network of arcs having a maximal capacity for flow. In turn this type of problem has two general versions. The first, the static maximal flow problem, is concerned with determining the maximal flow between two points in a network when the capacity values of all the arcs are known. The second, minimal cost flow problems, is concerned with assigning flows in the arcs of a network so as to get a minimal cost and satisfy certain flow constraints. Most of the results described by Ford

and Fulkerson [22] are given in terms of solutions to linear programming problems. Busacker and Saaty [23] in turn present a variety of algorithms to these and other related types of problems without using the linear programming approach. Another important type of design problem which is concerned with the simultaneous satisfaction of a multiplicity of source-sink flow requirements is called a multi-commodity network flow problem. Kim and Chein [24] have made important contributions in this area, in addition to Gomory and Hu [26,27], Mayeda [28], Jewell [29], and Hakimi [30]. Except for Kleinrock [25], all of the previously cited references have dealt exclusively with the single channel queue or with steady flow through a connected network.

There exists far less literature on the subject of stochastic or unsteady flow in connected networks. Kleinrock [25] presents the results of communication network simulation studies made on the TX-2 Computer at MIT Lincoln Laboratories for the purpose of comparing the performance of a number of different topological structures and routing procedures. He has also obtained some significant analytic results, the most important of which is a solution to the design optimization problem:

Minimize average response time, subject to the
constraint that a constant level of resources is
used solely for supplying channel capacity, where

topological structure, channel capacity, routing procedure, and priority discipline are treated as design variables.

The Kleinrock design approach, however, is limited in its scope of utilization by a number of important unrealities. Most important of these is the requirement that a linear cost vs. capacity function be assumed. In addition, the type of data concentrator cannot be treated as a design variable in the Kleinrock analysis since it applies solely to networks having concentrators with an infinite buffer for storage of message traffic which is in-transit. Another obvious, limiting consequence of the problem formulation is that the system cost component due to the concentration equipment is not considered.

Hunt [31], Jackson [32], Moran [33], Onaga [34], and Prosser [35,36] have also contributed significantly to the body of literature which describes solutions to problems concerned with interconnected multiple-channel queues and service facilities.

Hunt [31] conducted some parametric studies of tandem or sequential queueing networks. Jackson [32] made an important contribution to the solution of queueing networks with feedback, that is, one in which a job or customer is allowed to enter another or the same queue after having received service. This result of Jackson is presented in detail in Section 2.2.3 of this thesis. In 1959, Moran [33] wrote one of the first probability oriented

monographs on the theory of storage. Onaga [34] formulated a model which is concerned with the efficient transmission of stochastically arriving messages in a communication network which has very large storage capacity at every node. Prosser [35,36] has performed an analysis of routing procedures in stochastically fed communication networks.

Moving closer to the application area of specific interest in this research, specific contributions have been made by Haugh and Berk [37], Harrison [38], Cohler and Rubinstein [39], Weingarten [40], Saber and Young [41], and Stimler and Brons [42]. In 1963, Haugh and Berk [37] presented a paper on the problems associated with interfacing data communication facilities to digital computers and their remote terminals. Harrison [38] developed a model for assessing the storage requirements of a computer dedicated to message switching; no consideration was given to delay in this particular paper. Cohler and Rubinstein [39] described certain results obtained using a drum storage model of a sophisticated electronic message switch. Weingarten [40] extended previous congestion models of message switching computers to obtain approximate storage requirements in terms of actual storage capacity units instead of numbers of messages. More recently, Saber and Young [41] have discussed the tradeoffs of assigned and shared storage in message switching computers. Stimler and Brons [42] present a methodology for

optimizing real-time computer system performance; the procedures are carried out in detail for a drum oriented message switching system.

Martin [43,44], Karplus [45], Weber [46], Desmonde [47], and Margopoulous and Williams [48] have made significant contributions to the literature on teleprocessing system design and operation. Bricault and Delgalvis [49], Gay [50], Chang and Wong [51], and Seaman, et al. [52] have developed analytic queueing models for analyzing storage requirements and throughput in real-time systems where the processing unit is subjected to channel interference. Such interference arises since communications activities may claim a significant fraction of memory storage cycles in the processor.

The complexity inherent in analytic models of an entire teleprocessing system has restricted efforts in this area to the analysis and design of the central complex and a limited number of subsystem components in the remote portion of the network. Esau and Williams [53] have addressed the problem of designing the entire remote portion of the network. Their design approach and certain limitations which restrict its spectrum of useful application are discussed at length in Section 1.5.3 of this work.

Until the recent widespread use of simulation there has been a lack of substantial efforts dedicated to the study of total teleprocessing system behavior. Unfortunately, however, simulation has but

limited utility in a design environment. A discussion of the relative merits of the various modeling tools available to the system designer is now presented.

1.5.2 General Techniques for Determining System Performance

Coffman [54] and then Arden [55] have noted that the tools for determination of system behavior are analysis, simulation, and observation, ordered in what is usually regarded to be an increasing degree of realism. Analysis can further be subdivided into closed form and computational analysis. Most of the queueing models referenced in the previous section are examples of the closed form type of analysis. The Recursive Queue Analyzer (RQA) [56] exemplifies an analytic computational tool which is used to solve large stochastic networks whose behavior can be modeled as a Markov chain. The results of an observation or simulation can often be employed in a formal analytic approach--thus rendering possible models which combine the various techniques for determining system behavior. For example, Kleinrock has made joint use of closed form analytic techniques and simulation in [25].

Simulation has become increasingly popular lately and can be used very effectively to model networks in almost any desired degree of detail when system complexity excludes the possibility of using analytic techniques¹. The most severe limitations of simulation,

¹ For example, see Nielsen [57] and Pinkerton [58].

however, are its limited usefulness in a synthesis environment as well as the problems associated with comparison of independently conducted studies. This latter difficulty arises as a consequence of both the ill defined nature of the systems being modeled and inconsistencies in the level of detail embedded in the models. Recent efforts to embed simulators in high-level computational optimization languages are attempting to overcome these difficulties.

A variety of special purpose company proprietary simulation systems have been and are being developed for the analysis of communication networks. However, public domain information about these simulation systems is almost nonexistent.

The need for the development of modeling tools which are useful in a design environment is obvious. This effort will make use of all three types of techniques (analysis, simulation, and observation) in developing a teleprocessing network design model which is comprehensive and realistic. The critical elements in the design optimization procedure will be analytically oriented, with observation and simulation being used to demonstrate the validity of the assumptions which form the basis of the analytic procedure.

1.5.3 Centralized Computer Communication Network Design

This author is aware of only two published treatments of non-simulation oriented teleprocessing network design procedures dealing specifically with the centralized type of configuration. The

limitations of simulation have already been discussed in Section 1.5.2, where motivation for the development of analytic procedures was originally presented. There is a logical explanation for the paucity of efforts in this area; primarily, it is a consequence of the inherent combinatorial problems associated with the development of the topological structure for a network.

It is well known¹ that there exist $(n+1)^{n-1}$ different spanning tree topological structures in a network which has n remotely located terminals. This number assumes the existence of only one link capacity value and only one type of nodal structure. If there are m values of link capacity available and still only one node type, then the number becomes $m^n \times (n+1)^{n-1}$ different structures. A network which contains any multipoint segments can therefore be extremely difficult to optimize due to this exponential growth of the number of network configurations. As an example, for a network which has as few as nine remote terminals, with six link capacity values and a single type of node for data concentration there exist $10^8 \times 6^9$ possible configurations! In networks with twenty or thirty terminals the figure becomes almost incomprehensible. Even for small values of n it is therefore unrealistic to consider any enumerative approaches. Some design technique must be

¹ See, for example, Riordan [59].

developed which enables a large fraction of the nonoptimal configurations to be directly eliminated from consideration.

M. Held and R. M. Karp [60] have reported that successive approximation techniques similar to those presented in this work have been used quite successfully in practice, although there is no guarantee that an optimal solution will eventually be reached.

In seeking an optimal network, the cost of finding such a network must be given as much consideration as the cost of the network obtained. A near-optimal configuration which can be readily obtained and whose performance characteristics can be realistically predicted is preferable to a truly optimal solution which might require days or weeks of high speed digital computer time to obtain. A discussion of the two published treatments of the problem is now presented.

1.5.3.1 The Esau - Williams Network Design Algorithm

L. R. Esau and K. C. Williams [53] of IBM have described a heuristic algorithm for the construction of multipointed tree structures which have minimum line costs. The method described in this paper is based on the following assumptions:

- Maximum tolerable link loading is known a priori.
- The capacity value of all links considered is identical.
- Costs and relative performance of various types of data concentrators are not given consideration.

- Response time of system is not given consideration except by prohibiting the formation of any structures in which link loading exceeds the maximum permissible value.

Assuming that the completely point-to-point network is the most expensive configuration in terms of line costs, multipoint branches or segments are formed in an iterative fashion by deleting a link incident to the central facility (central link) and adding some noncentral link which costs less than the central link deleted and preserves the connectedness of the network. At each iteration a new topological configuration is formed which results in maximum line savings from the configuration at the previous iteration. The optimal configuration is obtained when no new configuration, which leads to positive line savings can be formed. Also, at any step in the iteration procedure no modification is allowed (even if it has positive cost savings) which would result in some link being loaded in excess of its maximum value.

The method assumes that the transmission capacities of all links considered are identical. In general, however, an optimal configuration should contain links operating at different speeds. The authors point out this fact by noting that lightly loaded links may be more economically restructured using lower speed links, but do not give further consideration to this important problem.

1.5.3.2 J. T. Martin on Line Layout and Network Design

A considerably less substantive discussion of the problem of obtaining the optimal network configuration is contained in Martin [44]. On the subject of line layout, the following procedure is presented.

"Starting at the farthest points from the computer, the locations may be linked together until those linked have the maximum permitted load. This group will then be linked to the computer. The total line distance in doing this will be kept to a minimum. Different combinations should be tried in an attempt to minimize the total line mileage. Having established a reasonable set of interconnections the exact distances and costs may be obtained from suitable tables. However, the problem is further complicated if concentrators or exchanges can be used."

Considerable insight into the state of the art is obtained in Martin's statement:

"In general no program the author has yet examined is sufficiently comprehensive to choose between the various alternatives of concentrator, remote buffer placement, . . . and the resolution of corresponding line speed problems."

1.6 Objectives of this Effort

The primary objective of this work is the development of performance measures and computational procedures which can be used to solve various resource allocation problems arising in the design of centralized computer-communication networks. The key word in this statement of purpose is "solve", since we will not obtain theoretically optimal solutions to all of the problems which

are posed.

In particular, when the design problem is given its most general formulation, network topological structure, link capacity values, and the type of concentrator used must all be treated as independent variables. Due to the exponential growth of the space of parameter variations as cited in Section 1.5.3, the problem rapidly becomes intractable for all but the most trivial of cases.

It is not unusual in engineering practice to solve large, unwieldy problems such as this one by decomposing the large problem into a number of independent subproblems each of which is amenable to solution by optimization techniques. In so doing, a theoretically optimal solution to the overall problem is sacrificed. However, in return the designer obtains an suboptimal solution to an otherwise unsolvable problem. The quality of the approximate solution can only be assessed by considering all of the following factors

1. How realistic is the design model?
2. How do the solutions obtained compare with those produced by alternate suboptimal procedures?
3. What is the cost of obtaining the suboptimal solution (in computer storage and execution time requirements)?

In this thesis we have approached the most general formulation of the network design problem by decomposing it into smaller independent subproblems. Optimization techniques will be used to solve the subproblems, so that we will end up with a "reasonable, suboptimal"

solution to the overall design problem. By reasonable, suboptimal solutions we mean:

1. It is reasonable to attack the unwieldy overall problem by decomposition into solvable subproblems.
2. The proposed design model reflects a greater degree of realism than existing design alternatives by considering all of the important variables to some extent.
3. As a consequence of such increased reality, the solutions obtained by this procedure are shown to be better than those from existing design procedures.
4. The solution techniques proposed herein are shown to be computationally efficient.

The efficacies of this approach will be discussed as the effort progresses.

1.7 Contributions

Having established the objectives of this research and the approach to be taken, we will now summarize the contributions which have been made in the order of their treatment.

1. A realistic, yet relatively simple analytic model has been developed for considering performance in the design of centralized computer-communication networks.¹ This model enables us to consider a variety of resource allocation

¹ This performance measure is the average response time which will be concisely defined in Section 2.2.

problems which arise in the design environment.

2. An optimum passage capacity assignment procedure is developed which enables one to obtain the minimum average response time for a network whose traffic, topological structure, link capacity values, and type of concentrator (multiplexor or message switching center) are known.
3. Simulation studies have been performed which establish the practical validity of making the infinite-buffer assumption in analyzing message switching center concentrators.
4. A unified presentation of queueing models for the various types of data concentrators is made. Some comparative studies are performed.
5. An optimization procedure of restricted scope is developed for the solution of the problem: In a network whose traffic, topological structure, and types of concentrators are known, obtain the least costly assignment of link capacity values which does not violate a given average response time constraint.
6. A network design procedure is developed which obtains solutions that utilize communication network resources more efficiently than those obtained by existing design procedures. The noteworthy advantages of this procedure are:

- a) The system designer may take maximum advantage of any economies of scale which exist in the cost of data transmission, since the model is capable of using nonlinear cost vs. capacity functions. It therefore yields a solution in which communication links need not all operate at the same speed.
- b) Cost and performance of various types of data concentrators are given explicit consideration.
- c) An analytic determination of performance can be made for any configuration considered.¹ Parametric effects of variations in any or all of the key design variables (structure, type of concentrator, link capacity, and cost) may thereby be readily assessed.
- d) It is shown that use of this procedure results in no loss of computational tractability.

¹ In some cases, the analytically determined value will represent an approximation which contains a negligibly small amount of error. Section 3.2 contains a detailed discussion of this point.

Chapter II

A DETAILED EXAMINATION OF THE PROBLEM AREA

2.1 Important Design Variables and Tradeoffs, Motivation

The important tradeoffs which influence the design procedure are now discussed; motivational considerations which influence various assumptions which will be made are also presented.

In general, the important macroscopic variables with which the designer of the network must be concerned are:

- Topological structure
- Link capacity value
- Response time or performance of the network
- System reliability
- Control procedures
- Types of data concentrators
- System cost.

The development of the topological structure of the network is probably the most critical portion of the design process; it is next to impossible to get any degree of cost-effective performance in a network where structure is poorly chosen. The rooted tree structure has been selected as a most natural representation for the centralized telecommunication network structures to be considered in this work. Since the average rate of requests from remote terminals for service at the central processing site is usually significantly below the rate at which the I/O channels of the central processor are capable

of accepting them, some type of concentrating scheme is desirable in the telecommunication network. Nodes in the tree which have more than one incident link represent the terminals at which this data concentration takes place. On the other hand, those nodes having only one incident link are called leaf nodes; the link incident to such nodes carries only the traffic associated with that particular terminal.

Recalling the methods for the classification of system structure presented in Section 1.2, in a completely point-to-point network (Figure 2.1) every remote terminal is connected to the central complex with a link which is not shared with any other terminal. A multipoint network (Figure 2.2) is a configuration in which at least one communication link in the network is shared by two or more terminals, that is to say at least one multipoint segment is to be found in the structure. A segment consists of the maximal set of terminals and interconnecting links which share a common link incident to the central complex. A segment is multipoint if two or more terminals share the link incident to the central facility and point-to-point otherwise.

The completely point-to-point network can seldom be justified from an economic standpoint since there is no sharing of communication links, and the cost of lines in this structure tends to be high in comparison to other structural alternatives. Thus a strong motivation exists for the development of techniques which lead to a sharing of the communication links in a network, and thereby reduce the total cost of

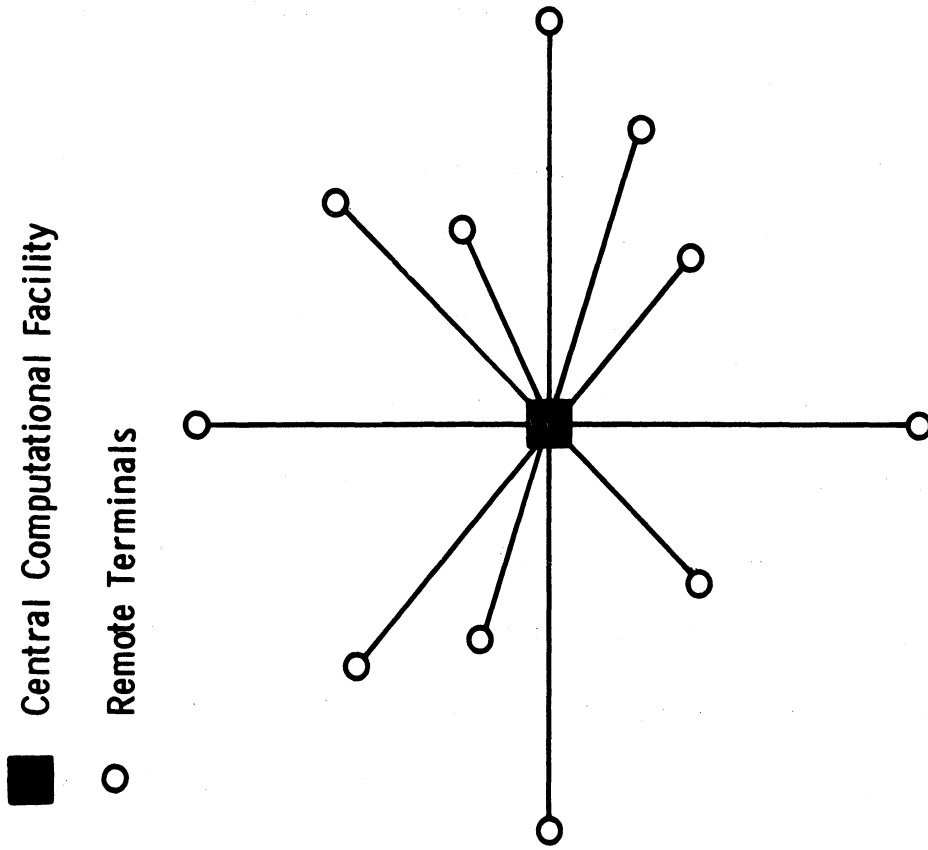


Figure 2.1 An Eleven Segment Point-to-Point Network

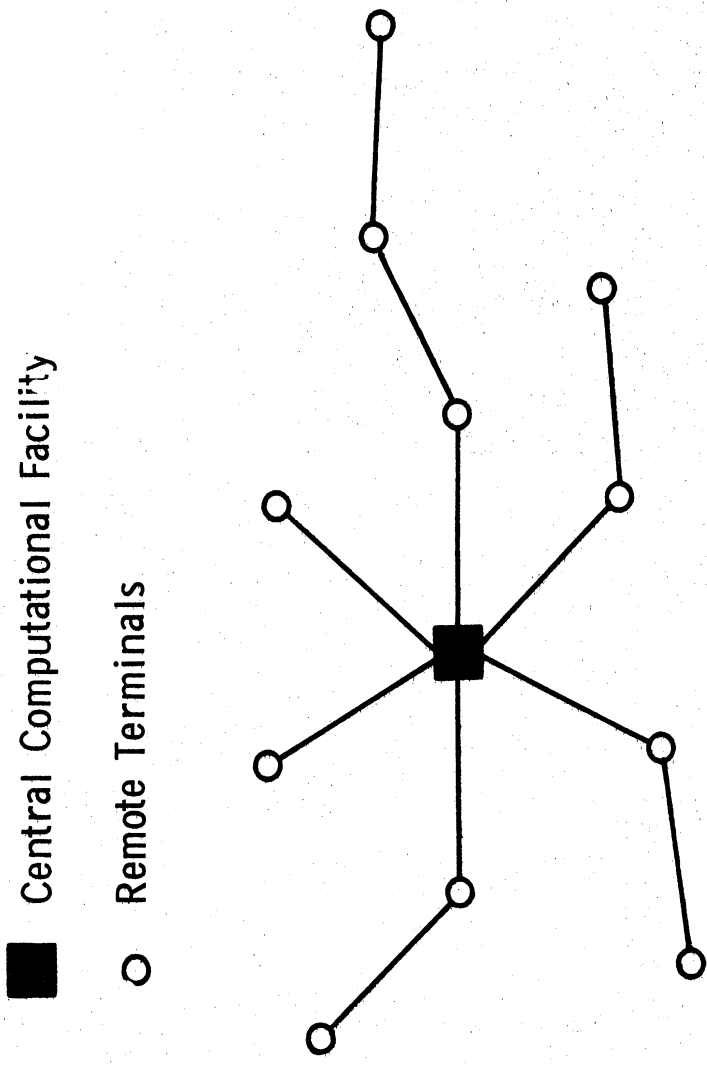


Figure 2.2 A Six Segment Multipoint Network

lines. However, as the number of shared links in a tree network increases, additional costs for concentrating equipment are incurred and must be traded off with line savings.

In general, a sequence of tree configurations with line costs successively lower than those of the point-to-point network can be generated by deleting a central link¹ and adding a less expensive non-central link which preserves the connectedness of the network. As these multipointed segments are formed, however, increased amounts of traffic are submitted to links which are shared. Unless the link capacity is increased, the congestion on such links increases and average response time increases. Usually, the increases in response time can be tolerated until the utilization² of a communication link approaches 50 or 60 percent. In general, no links are ever designed with utilizations in excess of 80 percent or 90 percent since response time fluctuates wildly on links loaded in excess of these values. As multipointed branches are so formed, even though line savings must be offset by additional costs for data concentrators, it is frequently possible to obtain substantial net savings in the process.

¹ Central links are those connecting a remote terminal directly to the central complex.

² Utilization and loading are used interchangeably throughout this work. In accordance with commonly used definitions, we refer to utilization here as a ratio of the average flow rate of bits (or characters) in a link to the maximum flow rate of bits (or characters) attainable.

Whenever a design procedure leads to configurations in which links are lightly loaded, it is usually possible to reduce the transmission capacity of lightly loaded links without altering performance severely. Line costs are thereby reduced and response time increases (although such increases are not likely to be intolerable when modified loadings do not exceed .5 or .6).

Although the per mile cost of a link increases as its transmission capacity increases, the rate of increase becomes smaller as capacity increases. (A typical cost function illustrating this point may be found in Figure 4.1 of Chapter 4). As a result, the cost per unit of transmission capacity is lower for links having larger transmission capacity values. Important line savings can be realized from these economies of scale in configurations which make use of higher speed snared links to the greatest extent possible.

Reliability problems are likely to arise for primarily two different reasons. On a given link, the signal to noise ratio may temporarily fall below a minimum acceptable value for a variety of reasons and cause spurious errors in the messages being transmitted over the channel. Error correcting procedures such as those presented in Section 1.3.1 are often used to minimize the effects of errors arising as a consequence of these types of anomalies. On the other hand, an entire link may fail, rendering any transmission impossible for a prolonged period of time. Since all the information paths which make use of that link are destroyed until the fault is repaired, it is

possible to use stand-by links such as those of the common carriers and counter the effects of such total outages. By incorporating such alternate direct links, the tree characteristic of the network is altered and the reliability characteristics are significantly improved.

Because the first type of error is likely to occur more frequently and because consideration of the second type of error renders any mathematical network analysis procedures essentially useless, this thesis will assume links that are reliable to the extent that no errors of this second type will occur under the normal operating conditions for which the design is intended. Errors of the first type will be assumed to be correctable (in a macroscopic sense) by using schemes such as those presented by Kucera [1]. Since the inability to detect and correct this type of error is almost always disastrous, certain uniformly acceptable redundancy levels will be used in all links. This redundancy will serve to reflect the price which the system designer is willing to pay for the ability to detect and correct all errors of the first type.

Regardless of the structural characteristics of a network, there are certain control requirements which are common to all telecommunications networks. The control procedures perform the following tasks:

1. Assembling the bits into characters and forming messages and message segments from these units of data.
2. Message editing (e. g. removing local control characters

from messages destined for the central computer.

3. Error detecting and correction.
4. Initiating and monitoring the message transmissions to and from the terminals.

In a macroscopic level approach to the design of teleprocessing systems, it is common to use a model building approach in which the micro-level tasks are subordinated to macro-level considerations in the early stages of design. Consequently, Tasks 1 and 2 above would be considered micro-level in relation to the effects which Tasks 3 and 4 have upon system performance. For this reason, alternative methods of implementing these first two control functions will not be treated in the models of this thesis. The third control function, that of error detection and correction has already been discussed.

Task 4, the monitoring of message arrivals from remote terminals is accomplished through a combination of software and hardware resources. Essentially, it consists of telling each remote terminal when to start transmitting or receiving its message. There are basically two different schemes which are commonly used for this control procedure :

1. Contention
2. Polling.

In a contention system, terminals in a particular multipoint segment openly compete by requesting a communication link or facility on an "as needed" basis. If the service facility in question is free,

transmission takes place; if not, a queue of contention requests builds, and queued requests are serviced according to some discipline which is a function of message priority, length and other factors. In the polling method of control, a supervisory program selectively queries the terminals on the polling list, asking each to transmit if it has anything to send. A terminal is thus queried at a frequency proportional to the number of times it appears on the polling list. Since entries on the polling list are usually specified under program control, the frequency of query can be matched to the average rate of arrivals and priority provisions can easily be incorporated. It is also common to equip the terminals with special hardware for automatic terminal address decoding of polling messages and for answering the polling query.

What are the conditions for which these two types of control should be used? Due to the inherent nature of operation, the contention scheme (as the nomenclature implies) is somewhat more vulnerable than polling to overload conditions in a highly utilized network. Polling is likely to be used in such environments because it allows more rigid control over the maximum delay which any one terminal might experience. But the implementation of the polling control scheme is more costly (especially from the standpoint of system programming) and will not always improve system performance appreciably.

In general, then, how is it possible to ascertain which of these two alternative control schemes is to be preferred? This decision

depends upon the polling rate and the number of terminals sharing the link. If line utilization is not too high (50 - 60 percent or under is a good rule-of-thumb) then there will be a close proximity between the response times of the two schemes. Also, if the time taken by the terminal control routine to do the polling is negligible, then the polled line would behave almost identically to the line operating under the contention scheme. Martin [44] has estimated that whenever the number of characters in the message transmitted is at least ten times the number in the polling query, the contention and polling control schemes are essentially equivalent from the standpoint of response time.

Whenever polling time is significant, arrivals to the system are likely to be somewhat less than completely random. Analytical techniques which are based on the assumption of random arrivals generally will predict response times higher than actual values, resulting in a conservative design with a built-in safety factor which makes the network less sensitive to minor fluctuations and results in a better operation. The techniques presented in this work will be precisely applicable whenever

- contention schemes are used for control
- or • polling control schemes are used in which negligible polling time is taken
- or • if polling time is not negligible, then there will exist a design safety factor which is desirable.

2.2 Definition of Response Time, Notational, Important Queueing Theorems

In this section notational conventions which will be used throughout the remainder of this work are introduced. A concise definition of response time is given and some important queueing theorems which will be used are also presented.

2.2.1 Forward and Reverse Networks

In the telecommunication networks of interest in this work, clearly both forward (source to processor) and reverse (processor to sink) types of message traffic exist. In the forward network, traffic will always be merging at points of concentration. In the reverse network, there will be a partitioning of traffic at the points of deconcentration. Input message traffic to the forward network occurs at the remote terminals; these messages exit from the forward network into the data channel region at the central processor site. Messages enter the reverse network at the data channel and exit from the reverse network at the respective remote terminals.

It must be pointed out that the response time to be considered in this work is not the same as the delay experienced by the user submitting message requests from a remote console. The actual response time experienced by such a user consists of three components:

1. Response time in the forward telecommunication network
2. Time spent in job queue and processing at the central processor site

3. Response time in the reverse telecommunication network.

The central purpose of this research is the development of techniques which minimize the average values of components (1) and (3). In general, the design variables which have the most substantial affect on component (2) are more properly associated with the processing subsystem than with the communications subsystem of the overall network. However, there is one such important variable that can affect component (2) which may be internal to communications subsystem—namely, the type of data concentrator. Sophisticated special purpose communications processors can relieve the CPU of routine control-oriented functions and thereby lower CPU overhead. This relationship can be accounted for in the design model by reducing the cost of such data concentrators commensurately with the value of the lower CPU overhead. For example, if the use of this type of concentrator would enable a less expensive CPU to be used in a given application, the net cost of using the sophisticated data concentrator would be given by its actual cost reduced by whatever savings are associated with using the less expensive CPU. Aside from this particular consideration, however, the focal point of this research will be the development of design procedures which minimize the average values of components (1) and (3) of the user response time.

As a consequence of the speed differential between remote terminals and CPUs, the average rate of message arrivals to the reverse

network destined for a given terminal usually exceeds the rate of message arrivals to the forward network originating from that same terminal. As evidence of this fact, Totschek [61], in observations of the SDC Time Sharing System, indicates that typical values for the ratio of console output arrival rates to console input arrival rates range between 2 and 4. The average length of messages in the reverse network is also usually larger than that for messages in the forward network. To be sure, no two systems are identical, but most generally, the product of the mean arrival rate and the mean message length for the reverse network will exceed that same product for a corresponding terminal in the forward network¹.

This observation leads to the often conjectured conclusion that symmetric channels are not ideally suited for computer-communications activities. In designing a telecommunication network, separate considerations should therefore be given to both the forward and reverse networks. Since little progress toward achieving widespread utilization of asymmetric channels has been reported, the design techniques presented herein are intended to be of use in structuring either the forward or the reverse network or both. Throughout the work it will be assumed that the forward and reverse networks operate on a noncompetitive basis and that there is complete isolation

¹ This product is equivalent to the average flow in bits/sec. or char/sec. to or from a remote terminal.

of forward and reverse message traffic in communications facilities which are shared. However, it must be noted that if symmetric channels are used, then the design is constrained to be based on the particular network (frequently the reverse) which submits traffic having the largest mean arrival rate, mean message length product. In this case, the design will be a joint solution for both forward and reverse networks only if the average flow of message traffic from sources to computer equals the average flow of message traffic from computer to the sinks at the same terminals.

An alternative design approach would be to consider the usage of unidirectional channels only and design the forward and reverse networks independently whenever there is an asymmetric relationship between message flow in the forward and reverse networks. In any case, though, the procedures to be discussed herein can be applied to either network. The notion of isolation between the forward and reverse traffic will be implicitly understood.

2.2.2 Notational

Now, as a prelude to citing concise mathematical relationships for the performance parameters of networks, the following notation is introduced.

$\gamma_i \triangleq$ Poisson arrival rate of messages entering the network per second from an external source at the i -th terminal and destined for the CPU.

$k_i \gamma_i \triangleq$ Poisson arrival rate of messages entering the network per second from the CPU and destined for the i -th terminal.

$1/\mu_f \triangleq$ Mean length of exponential message length distribution governing forward traffic, or that originating from external sources at remote terminals and destined for the CPU¹.

$1/\mu_r \triangleq$ Mean length of exponential message length distribution governing reverse traffic, or that originating at the CPU and destined for remote terminals².

$C_i^{(f)} \triangleq$ Transfer rate of information bits (TRIB) of forward channel emanating from the i -th terminal.

$C_i^{(r)} \triangleq$ Transfer rate of information bits (TRIB) of reverse channel submitting traffic to the i -th terminal.

The above symbols are idemic (symbols which implicitly represent values). The following symbols are, on the contrary, intended to be unvalued, or those for which value is dependent upon whether the forward or reverse network is being considered. Expressions which follow will be stated in terms of unvalued symbols, whenever not explicitly indicated otherwise.

¹ Implicit in this definition is the notion of forward message traffic having identical message length distributions for all terminals.

² Implicit in this definition is the notion of reverse message traffic having identical message length distributions for all channels.

Symbol	Definition	Value if the forward network is under consideration	Value if the reverse network is under consideration
$1/\mu$	Mean length of exponential message length distribution for channels of the network	$1/\mu_f$	$1/\mu_r$
λ_i	Total Poisson arrival rate of message traffic to the i-th channel of the network. I denotes the set of all terminals whose path to/from the CPU includes channel i.	$\sum_{j \in I} \gamma_j$	$\sum_{j \in I} k_j \gamma_j$
C_i	Transfer rate of information bits (TRIB) associated with the i-th channel in the network.	$C_i^{(f)}$	$C_i^{(r)}$
T_i	Average delay (response time) for a message passing through channel i, includes both the time in queue and time in transmission. (See Section 2.2.3 for derivation).	$\frac{1}{\mu_f C_i^{(f)} - \sum_{j \in I} \gamma_j}$	$\frac{1}{\mu_r C_i^{(r)} - \sum_{j \in I} k_j \gamma_j}$
T	The average per message delay for all source sink paths in the network hereafter referred to as the response time of the telecommunication network. A mathematical definition expressing T as a function of other network parameters is given in Section 2.2.3.	T_{forward} See Expression 2.1	T_{reverse} See Expression 2.2

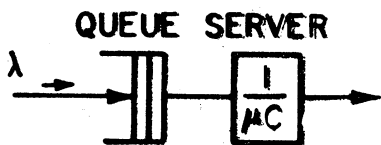
$$T_{\text{forward}} = \sum_i \left[\frac{\sum_{x \in I} \gamma_x}{\sum_j \gamma_j} \quad \frac{1}{\mu_f C_i^{(f)} - \sum_{x \in I} \gamma_x} \right] \quad (2.1)$$

$$T_{\text{reverse}} = \sum_i \left[\frac{\sum_{x \in I} k_x \gamma_x}{\sum_j k_j \gamma_j} \quad \frac{1}{\mu_r C_i^{(r)} - \sum_{x \in I} k_x \gamma_x} \right] \quad (2.2)$$

where the i and j summations are over all channels in the network.

2.2.3 Important Queueing Theorems and Definitions of Response Time

Consider the following simple queue server combination in which arrivals are Poisson with rate λ , queue capacity is infinite, servicing is first-come, first-served, and the service time is an exponentially distributed random variable with mean $\frac{1}{\mu C}$.



It is well known¹ that the expression for the average time spent in queue and service, denoted \bar{t} , is given by

$$\bar{t} = \frac{1/\mu C}{1 - \frac{\lambda}{\mu C}} = \frac{1}{\mu C - \lambda} \quad (2.3)$$

¹ See, for example, Feller [62].

Now, further, it is clear that in a communication network if messages arrive at the i -th channel with Poisson rate λ_i having exponentially distributed lengths with mean $\frac{1}{\mu}$, and if the transmission capacity of the channel is C_i , then there will be an average delay time T_i , for messages on that channel which is given by:

$$T_i = \frac{1}{\mu C_i - \lambda_i} \quad (2.4)$$

This expression precisely defines T_i which was introduced in the previous section. To go from this single channel case to the more interesting instance where the average response time reflects the contributions of all the channels in a network, we first cite important theorems due to Jackson [32] and Burke [19].

Jackson's Theorem:

Assume that in a queueing network the following conditions hold:

1. All queues have an infinite capacity.
2. There exist n exponential service nodes (Figure 2.3) and that the k -th ($k=1, 2, \dots, n$) consists of N_k parallel servers, each with a mean service time S_k .
3. Messages arrive at node k from outside the system in a Poisson stream at mean rate γ_k . Messages also arrive at node k from other nodes inside the network.
4. Whenever a message is processed at node k , it moves either to node j ($j=1, 2, \dots, n$) with probability θ_{kj} , or out

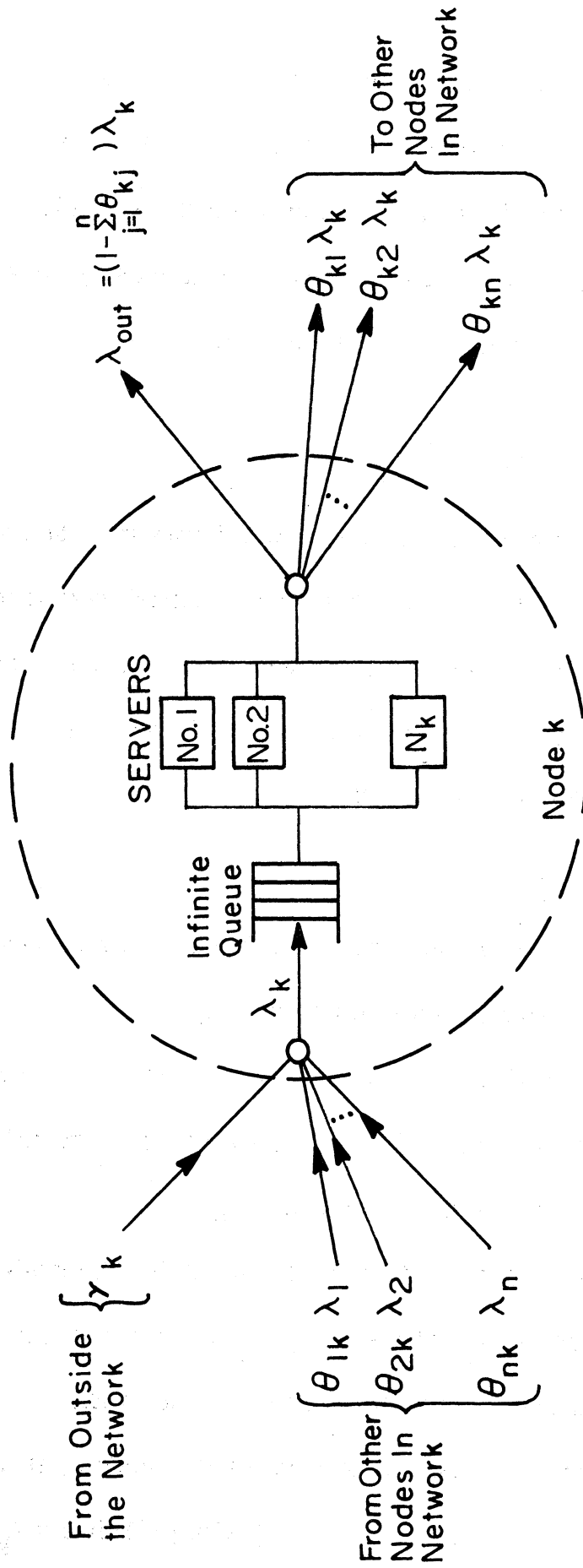


Figure 2.3 Schematic Diagram of the Typical Exponential Service Node in Networks under Consideration in Jackson's Theorem

of the system, completing its transmission through the network with probability $1 - \sum_{j=1}^n \theta_{kj}$. Thus the total message traffic arrival rate at node k from inside and outside the system is given by $\lambda_k = \gamma_k + \sum_{j=1}^n \theta_{jk} \lambda_j$.

5. The total message arrival rate λ_k to node k is less than N_k/S_k , the average service rate at node k ($k = 1, 2, \dots, n$).

Then, in the steady state, each node k is an independent multiserver subsystem with Poisson input λ_k , as far as queue size and average queueing time are concerned. Thus each node may be analyzed separately from the others using conventional techniques for a single multiserver configuration.

As will be shown at a later point in this section, Jackson's theorem enables the mean message delay in the overall network to be obtained by a summation of the mean message delays arising at each of the independent servicing subsystems. Important groundwork for Jackson's theorem was performed by Burke [19] who is responsible for characterizing the steady state interdeparture distribution from a single multiserver system to which the arrivals are Poisson and whose individual servers are exponential.

Burke's Theorem

Consider Figure 2.4, a schematic diagram of a multiserver queueing system. In the steady state, the distribution of inter-departures from an infinite queue having m identical exponential servers in parallel and Poisson arrivals with mean λ will itself be Poisson.

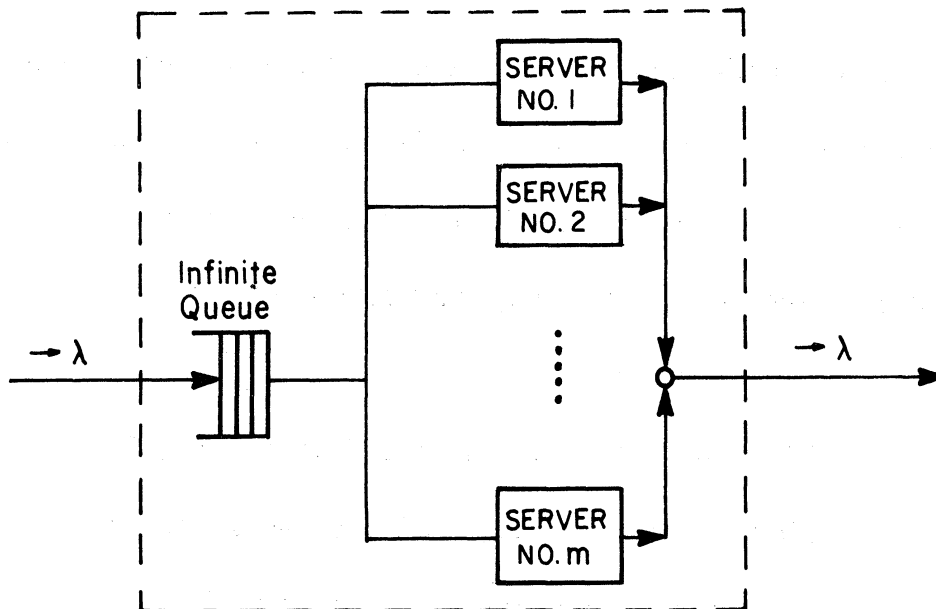


Figure 2.4 A Multiserver System

To cite the usefulness of Jackson's and Burke's theorems in analyzing rooted tree networks, consider the cases of the merging and partitioning of Poisson traffic in queueing networks where all servers are exponential with the same mean, there is an infinite storage capacity at each node, and the servicing discipline is first-come, first-served. In Figure 2.5, the merging situation, input and output Poisson parameters may be related as follows:

$$\lambda_{n+1} = \sum_{j=1}^n \lambda_j$$

and

$$\lambda_{n+m+1} = \sum_{i=1}^m \lambda_{n+i}$$

and so forth.

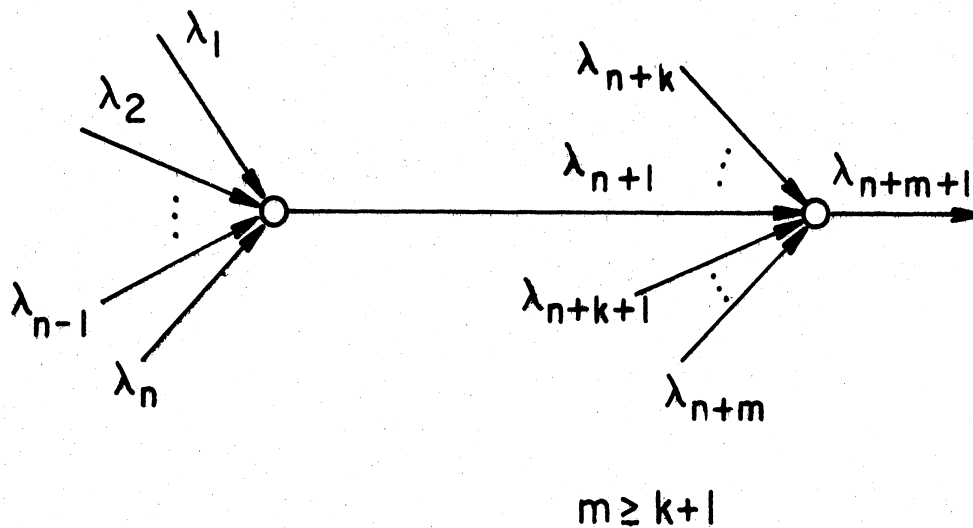


Figure 2.5 Poisson Traffic Merging

In the case of Poisson traffic partitioning, (Figure 2.6), if, when a message arrives at the partitioning node, it randomly chooses route i with probability p_i , then the theorems imply:

$$\lambda_i = p_i \lambda \quad i=1, 2, \dots, n$$

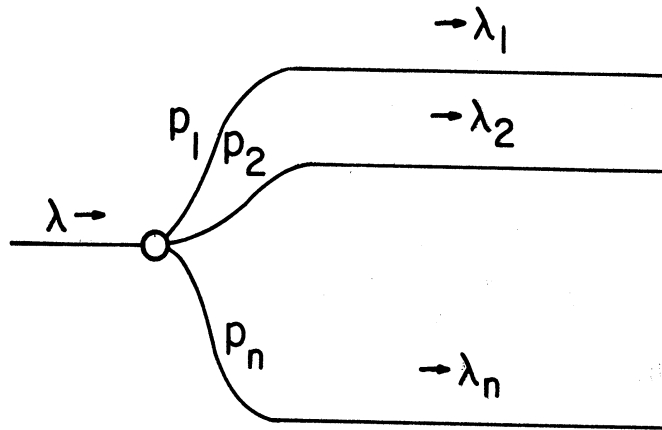


Figure 2.6 Poisson Traffic Partitioning

In queueing networks satisfying the conditions for Jackson's theorem, then, an equivalence can be shown between the average per message delay over all origin-destination pairs and the value obtained by weight-summing the average delays of all the independent channels of the network.¹ The average per message delay, T , over all paths can be defined as:

$$T = \frac{\sum_{i,j} \beta_{ij} Z_{ij}}{\sum_{i,j} \beta_{ij}} \quad (2.5)$$

where

$\beta_{ij} \triangleq$ average Poisson arrival rate of messages entering the network with origin i and destination j .

¹ See Kleinrock [25], p. 66. It is also assumed that any two messages originating at node i destined for node j will be transmitted over the same channels. Thus there is a single, fixed route for every origin-destination node pair in the network.

and

$Z_{ij} \triangleq$ average message delay for messages having origin i and destination j .

and the $\sum_{i,j}$ is over all origin-destination pairs (i, j) in the network. T is thus properly defined to be the weighted sum of the Z_{ij} where the weighting factor is chosen to be proportional to the total number of messages which must suffer the delay Z_{ij} .

As a consequence of Jackson's theorem, it is possible to decompose a particular Z_{ij} into a sum of the average delays which must be encountered in traversing the channels along the path from origin i to destination j . For a particular i, j origin-destination node pair in a network having n channels we can write

$$Z_{ij} = \sum_{k=1}^n \delta_k T_k \quad (2.6)$$

where

$$\delta_k = \begin{cases} 1 & \text{if channel } k \text{ transmits messages having origin} \\ & \text{ } i \text{ and destination } j \\ 0 & \text{otherwise} \end{cases}$$

and T_k denotes the average transmission delay for the k -th channel of the path from i to j . Insert expression (2.6) for Z_{ij} into expression 2.5 and expand fully. Then, after grouping terms involving T_i , we will have an expression of the form

$$T = \sum_{i=1}^n \frac{\lambda_i T_i}{\sum_{j,k} \beta_{jk}} \quad (2.7)$$

where the j, k sum is taken over all origin-destination node pairs in the network and

$$\lambda_i = \sum_{(x,y) \in I} \beta_{x,y}$$

where $(x, y) \in I$ denotes all origin-destination node pairs which transmit messages over the i -th channel of the n channel network. In networks which have but a single destination node, clearly the second subscript on β_{jk} and β_{xy} in the above expressions is superfluous.

Throughout the remainder of this work, γ_j will be notationally equivalent to β_{j1} where the destination node 1 will denote the central processing site for the forward network. In the reverse network, $k_j \gamma_j$ will be notationally equivalent to β_{1j} where origin node 1 will again denote the central processing site. Thus a pair of expressions for the average response times in both the forward and reverse networks can be written:

$$T_{\text{forward}} = \sum_i \frac{\lambda_i}{\sum_j \gamma_j} T_i \quad (2.8)$$

$$T_{\text{reverse}} = \sum_i \frac{\lambda_i}{\sum_j k_j \gamma_j} T_i \quad (2.9)$$

where the i and j summations are over all independent channels and terminals in the network, respectively. The expressions for λ_i and T_i in the two cases can be recalled:

$$\lambda_i \triangleq \begin{cases} \sum_{j \in I} \gamma_j & \text{for (2.8)} \\ \sum_{j \in I} k_j \gamma_j & \text{for (2.9)} \end{cases}$$

and

$$T_i \triangleq \begin{cases} \frac{1}{\mu_f C_i^{(f)} - \sum_{j \in I} \gamma_j} & \text{for (2.8)} \\ \frac{1}{\mu_r C_i^{(r)} - \sum_{j \in I} k_j \gamma_j} & \text{for (2.9)} \end{cases}$$

I is again the set of all terminals which exchange information with the CPU using the i -th channel. Direct substitution of these values into (2.8) and (2.9) respectively will lead to the expression (2.1) and (2.2) for the average response time in the forward and reverse network

2.3 Precise Problem Statement, Assumptions

This thesis will devote primary attention to the problem of developing realistic models and efficient computational procedures which can be used to solve various resource allocation problems

arising in the design of centralized computer-communication networks. Specifically, in Chapter 3 an optimization procedure is developed for solving the problem of minimizing the average response time in a network whose topological structure, link capacity values, and types of concentrators (multiplexor or message switching center) are known. The variables are the passage capacity assignments in multiplexed links. Then in Chapter 5 we address the problem of assigning the minimum amount of capacity to the links of a network whose topological structure and types of concentrators are known, without violating a given average response time constraint.

Finally, in Chapter 6 we consider the most general formulation of the design problem in which topological structure, types of data concentrators, and link capacity values are all treated as the independent variables. Any solution configuration considered is constrained to have an average response time less than a given maximum acceptable value. As was discussed in Section 1.6, the design procedure will obtain reasonable, suboptimal solutions which are shown to be better than those obtained by existing design procedures. The design model will incorporate important tradeoffs between all the critical design variables - tradeoffs not considered in previously reported efforts. Other noteworthy features and desirable characteristics of the design procedure are discussed in Sections 1.7 and 6.2.

Since the important macroscopic level design parameters which will be varied are topological structure, link speeds, types of data

concentrators, average response time, and system cost, the design model will accept the following primary inputs:

- The Poisson message arrival rate for each remote terminal.
- The mean of an exponentially distributed message length variable.
- A nonlinear, discrete-valued cost vs. link capacity function.
- Geographic coordinates for the various remote terminals and central processor.
- Cost data for the data concentrating equipment.
- A maximum line loading or utilization factor.
- The maximum average response time, T_{\max} , which is acceptable.

Output from the procedure will consist of a topological configuration, link capacity values, the type of data concentrator at each concentration node, the optimal passage capacity assignments within those links which are multiplexed, the average response time, and the total system cost.

The assumptions upon which the design procedures is based are now presented. Appropriate discussion of the assumptions follows.

1. Message arrivals to the network are Poisson.
2. Message lengths are exponentially distributed with the same mean for all terminals.
3. All messages have equal priorities.
4. Whenever a data concentrator is used that is equipped with buffer storage for in-transit messages, the buffer size is large enough so that the assumption of Poisson inter-departure statistics from such concentrators can be justified.
5. Special telephone or exchange types of concentrating equipment will not be considered.
6. Data transmission errors will be assumed correctable by using various redundancy techniques; however reliable links will be assumed to the extent that no total link failures will occur.
7. Leased lines are assumed although the model can be used for analyzing networks with switched links operating after all circuits have been established.¹
8. In multipoint segments, either contention control is used, or polling schemes are used in which negligible polling time is taken, or if polling time is not negligible the resultant safety factor is within acceptable limits.

¹ No provision is made for considering the time delay associated with establishing the circuit connection in the switched link case.

9. Forward and reverse networks operate on a noncompetitive basis, that is message traffic in the forward network is effectively isolated from that in the reverse network and vice versa.
10. It will be assumed that remote terminals are buffered and hence capable of receiving messages as fast as the network can transmit them.
11. Data concentrators will be located at terminals which include at least one source or sink.
12. Cost of link transmission capacity is a nonlinear, discrete-valued function of capacity and a linear function of the length of the communication link.

2.3.1 Discussion of Assumptions, Empirical Data

Assumptions 1, 2 are commonly made in both analytical and simulation oriented studies of real time systems. Arden [55] indicates that Assumption 1 seems to be a reasonable one. Scherr [63] has shown that analytic models which require Assumption 1 can be quite accurate in modeling actual system behavior (the MAC CTSS system specifically). He also presented empirical data which demonstrated that the central processor service time distribution for the MAC CTSS system is quite close to exponential. Application of Burke's theorem (section 2.2.3) would imply that when Poisson arrivals to the forward telecommunications network are processed

by an exponential server (the CPU) that the interdepartures from the CPU and hence the interarrivals to the reverse telecommunication network would also be Poisson. It should be noted that the assumption of exponentially distributed CPU service time has not been explicitly made in this thesis and is not necessary if the distribution of message arrivals to the reverse network can be assumed to be Poisson.

Erikson of SDC [64] concludes that certain studies of job running times and user arrival times have shown these distributions to be closely but not exactly described by the Poisson and exponential distributions, respectively. In numerous other studies (Coffman [54], Schrage [65], Smith [66], and Fife [67]) authentic models of remote access, real time systems have been developed that use these classical assumptions concerning Poisson and exponential distributions.

Empirical data gathered on the JOSS system at the Rand Corporation [68] supports both Assumptions 1 and 2. Other evidence such as the SDC data [69] supports Assumption 1, but is less convincing concerning exponential length distributions. Private correspondence with J. T. Martin, author of several publications on the design of real time systems, has revealed that most of the empirical data gathered from the airline reservation systems tends to support both Assumptions 1 and 2. He has also noted that systems in which long messages tend to occur are often designed to automatically fractionate the long messages at the point of origination, effectively forcing the distribution toward the negative exponential.

Even when intuition suggests that arrivals would seem to be non-Poisson and/or the message length distributions non-exponential, valuable use of Assumptions 1-2 can be made. It is well known that when arrivals tend to be more deterministic than Poisson, other factors aside, making the Poisson assumption will lead to a predicted delay which is larger than the actual; such safety factors are often desirable in the early stages of design. On the other hand, when the arrivals come in bunches, Assumption 1 will lead to an underestimation of the congestion problem¹. A wide dispersion in the lengths of messages will not justify using Assumption 2. However, if the distribution has a lower degree of dispersion than the negative exponential, the second assumption will again lead to a design with an inherent safety factor. Further evidence of the usefulness of these analytic distributions comes from the observation that they are often used even in simulation studies where arbitrary probability distributions would have been no more difficult to incorporate².

The additional mathematical complexity associated with models in which any of Assumptions 3, 7 and 10 are relaxed would have a significantly adverse affect on the efficiency of the design procedure. Furthermore, it would appear that relaxing any of these assumptions would extend the design model's scope of applicability to, at best, a limited extent.

¹ For an example of analysis of delay in some specialized cases of bunched arrivals, see Reference [70], p. 53.

² See, for example, Reference [71].

Assumption 4 is not as restrictive as might initially appear for several reasons. Data concentrators which provide intermediate buffer storage for messages are cost effective only in a limited number of situations and usually demand an environment involving large numbers of incident links. Usage of such concentrators will be limited as a consequence. Efficient operation of a telecommunication network demands that the probability of a message attempting to enter the concentrator and finding a saturated buffer be kept very low; 10^{-3} or 10^{-4} are common design objectives. Buffers designed for these saturation probabilities reject few enough messages that there is a negligible performance differential between this type of service node and the infinite capacity queueing system discussed earlier in this chapter. Simulation has been used in Chapter 3 to compare the performance of the finite and infinite storage capacity buffers for these types of concentrators.

Assumption 5 has been made solely because the telephone and exchange types of concentrating devices are not strictly general purpose devices since they usually require all input and output line speeds to be identical. Furthermore, it is shown in Chapter 3 that these types of concentrators do not generally utilize the lines in the network efficiently, and that inordinate congestion for accessing a facility can result as a consequence of the relatively uncontrolled environment which exists when these types of concentrators are used. The primary purpose of this thesis is the development of a model

which considers the cost effectiveness of strictly general purpose concentrating equipment. Assumptions 6, 8, and 9 have been discussed previously in Section 1.3.1, 2.1, and 2.2.1 respectively. Assumption 10 was made in the interests of generality and with a foresight reflecting the likelihood of near term advances in the terminal hardware state-of-the-art. Assumptions 11 and 12 were made in order to incorporate as much efficiency and reality into the design procedure as possible.

Having established these assumptions and specified the problems to be solved, we now begin our efforts to accomplish these objectives. As a first step in the desired direction, Chapter 3 will be devoted to a detailed examination of various types of data concentrators in the case where network structure and link capacity values are known.

Chapter III

ANALYSIS OF VARIOUS TYPES OF DATA CONCENTRATORS

3.0 Introduction

Since the average arrival rate of requests from remote terminals for service at the central processing terminal is usually significantly below the rate at which the central processor is capable of accepting such requests, efficient utilization of system resources demands the usage of some means for concentrating the requests from a number of remote terminals into one or more high speed input-output channels of the central processor. The question of the physical location at which the concentration takes place is a significant one whenever the remote terminals are not all clustered within a short distance of the CPU site. If all data concentration functions are performed local to the CPU, then each remote terminal will require its own individual communication link connection to the CPU. It is frequently possible to obtain significant line cost savings by concentrating at various remote locations a number of terminals onto high speed links whose speeds are commensurate with the I/O channel speed of the central processor.

In this chapter queueing models for the various types of data concentrators are presented. These models are then used to parametrically assess tradeoffs involved in the selection of a particular mode of concentration. Optimum passage capacity assignment procedures for multiplexed links are also presented. Some of the

results of this chapter are then incorporated into models used to solve the other design oriented resource allocation problems considered in Chapters 4 - 6.

3.1 Various Methods of Data Concentration

There exist in the communications industry today a number of different types of concentrating devices. In general these devices have certain operational principles in common; hence their discussion as a class of products is useful. A knowledge of the advantages and limitations of the various modes of concentration can be useful in determining the direction in which communications hardware design efforts should proceed, as well as in enabling the system designer to decide which type of concentrator to use in a particular application.

The four basic types of concentrators are:

1. Message Switching Center Concentrator (MSC)

This type of concentrator is a special purpose communications processor with hardware and software features for bit-character operations, error correction and detection, control, storage of messages while in transit, etc. In this type of concentrator, messages from low duty-cycle incoming lines are accepted without delay from the terminals, assembled and queued, awaiting transmission on the high speed channel. Thus the high

speed line acts as a server for the queue of message transmission requests which arrive over the input links.

In general, this type of concentrator is only used when large numbers of input lines funnel message traffic through a concentrator node.

2. Multiplexing Concentrator (MX)

This type of concentration scheme involves the interlacing of data from a number of low speed input links onto a high speed link by a speed changing scanner which operates in the multiplexor to form separate channels on the high speed line for each input link.

These channels are formed via the creation of frequency bands or time slots on the multiplexed link.

- Frequency Division Multiplexor (FDM)

Independent frequency bands are formed on the multiplexed link for each of the input low speed lines. Frequency shift keying techniques are used to position the center frequencies for each of the individual bands.

- Time Division Multiplexor (TDM)

There are numerous ways of implementing TDM schemes.

Since it is important to allow input terminals to run at their own clock speeds rather than be dependent on the MX device, asynchronous schemes which allow convenient routing of signals through tandem multiplexor

stages are the most flexible types. This type of TDM devotes a fraction of the total input line scan cycle to each of the input lines, thus forming a time-slot channel on the high speed link for each input line. This type of operation encompasses store and forward schemes in which buffering is done on a character or bit basis but not on a message basis.

3. Way Station Concentration

This type of concentrator, not as widely used as either of the first two types, selectively gives access to the links of a certain information path to users who physically share certain links in the path. The way station device schedules the usage of the path according to some control procedure. Lines which are connected using this type of concentrating scheme are sometimes called multidrop lines.

4. Telephone Type Concentrator

This type of concentrator functions similarly to a telephone exchange; hence its name. A large number of low speed links of the same speed, are selectively connected, using some control scheme, to a smaller number of trunks. Multiplexing is therefore accomplished on a message by message basis with no speed change between the input and output or trunk lines.

In general, this type of concentrator is used for terminals which are not heavily utilized. The way station concentrator would be functionally equivalent to a telephone type in which there is only one trunk.

3.2 The Message Switching Center Concentrator (MSC)

MSC's are relatively expensive devices and hence are economically justifiable only at locations where the costs can be spread over large numbers of incident links. It was, however, noted in Section 2.2.1 that the net cost of using MSC concentrators can be less than the actual cost of the devices, since these sophisticated concentrators are capable of relieving the CPU of a large fraction of routine communications-oriented processing tasks. Whatever value is associated with this resultant decreased CPU overhead can offset the cost of the MSC's which are incorporated in the system. Models for assessing the tradeoffs which arise when the IBM System/360 is subjected to varying degrees of this channel interference have been developed by Chang and Wong [51] and Gay [50]. They can be used to estimate the CPU throughput increases which result from reduced data channel interference and to determine the actual value of this throughput increase. In this work, for reasons that will be explained in Section 3.7, we have elected not to consider such effects. Our efforts will be solely concerned with models for considering message throughput in the remote network. We now commence these efforts with the queueing models for the MSC.

3.2.1 Model for Message Switching Center, Forward Network

Consider the single concentrator configuration of Figure 3.1, where $\{\alpha_1, \dots, \alpha_n\}$ denotes the TRIB values of the input links and α_H the TRIB value for the shared link. It is of interest to relate the parameters of the network in Figure 3.1 to the general expressions (2.8) and (2.9) for the average message delay (response time) in a network with an arbitrary number of channels.

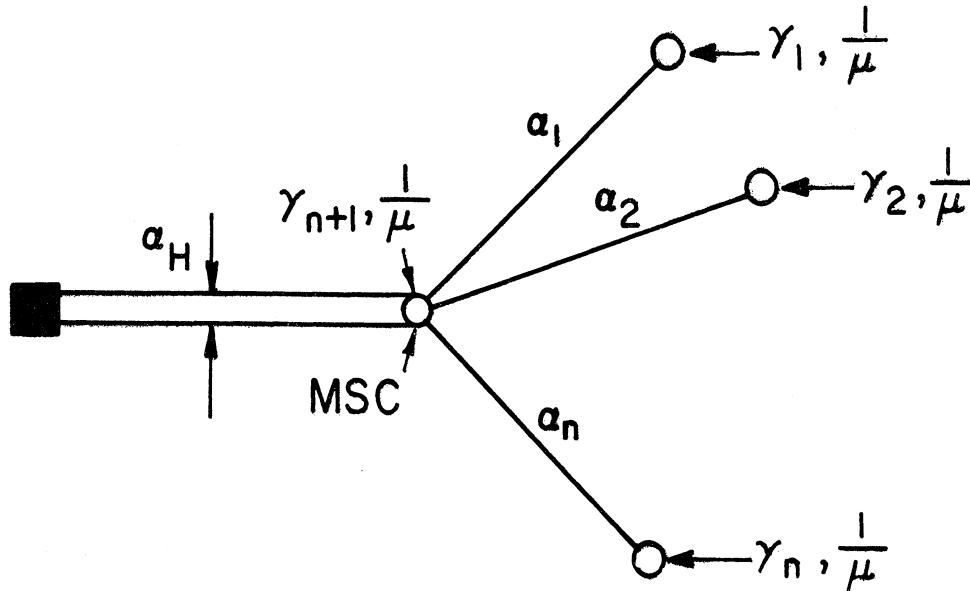


Figure 3.1 Schematic Diagram for the Single Stage MSC, Forward Network

If it is assumed that the MSC has an infinite buffer storage area so that the message traffic on the high speed link is Poisson then the analysis of the queueing model of Figure 3.2 leads to the expression (3.1) for T , the average response time of the network.

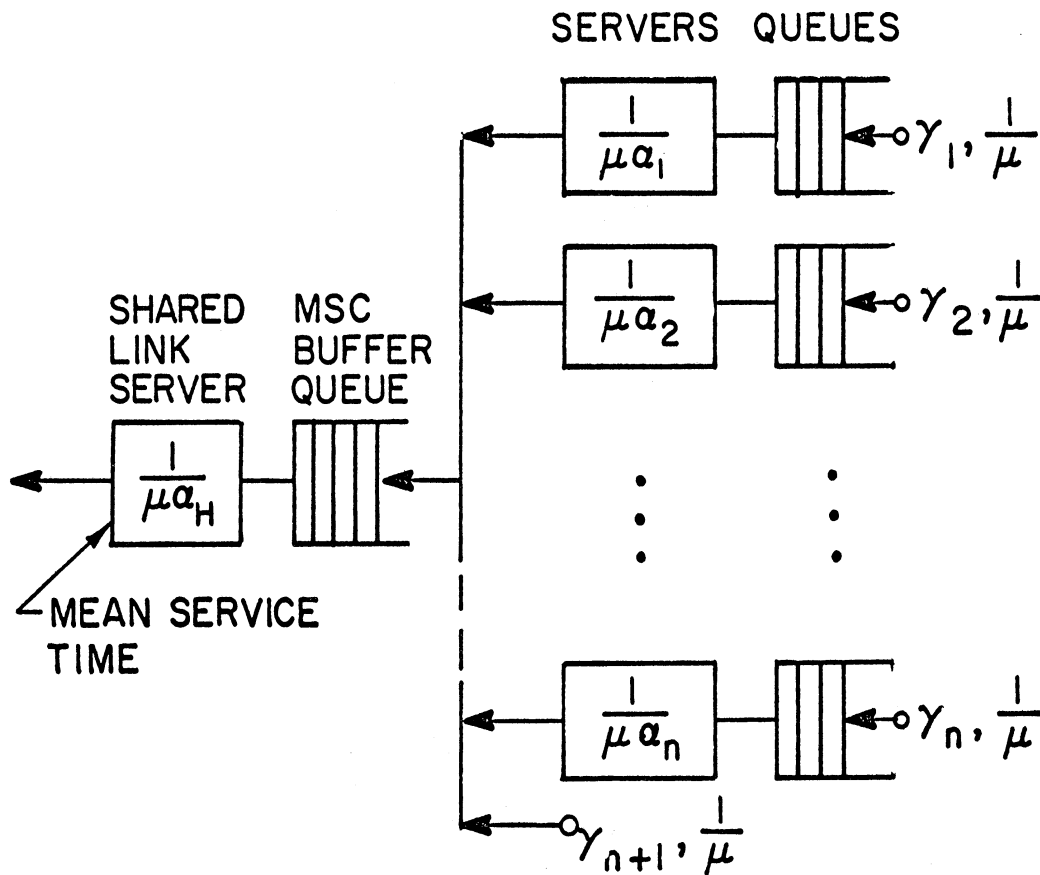


Figure 3.2 Queuing Model for the Single Stage MSC Concentrator

Since there are clearly $n+1$ independent channels in the system, and the shared link carries Poisson traffic of intensity

$$\lambda_{n+1} = \sum_{i=1}^{n+1} \gamma_i, \text{ we can write}$$

$$T = \frac{1}{\sum_{i=1}^{n+1} \gamma_i} \left[\sum_{i=1}^n \left(\frac{1}{\frac{\alpha_i}{\gamma_i} - 1} \right) + \frac{1}{\frac{\alpha_H}{\lambda_{n+1}} - 1} \right] \quad (3.1)$$

Obviously in most practical applications of interest, the buffer storage capacity in the MSC is finite, so that the expression (3.1)

becomes an approximation to the true average delay, as a consequence of error introduced in the second term. A number of papers have been published on the subject of storage requirements in a general message switching system having a multiplicity of input and output channels¹. The majority of these analytic results are directed toward the statistics of the number of messages in the system; such a measure is adequate in evaluating storage effectiveness only when the basic buffer is implemented with storage units each capable of containing a message of arbitrary length (a difficult requirement to satisfy with exponentially distributed message lengths, as is sagely pointed out by Snyder [72]). In present-day message switching centers, however, messages are usually assembled and stored in blocks or segments of storage, the basic units of which are bits, characters, or words, etc.. Messages having certain parameters in common (such as the same output link destination) usually share particular blocks of the buffer storage region to obtain the most effective utilization of storage. In such systems, occupancy of the segmented store must be determined as a function of the block size and the message length distributions, using basic storage capacity denominations such as bits, characters, words, etc., as has been noted by Snyder [72].

¹ See, for example, References [37], [38], [39], [40], and [41].

These particular modeling requirements make it extremely difficult, if not impossible, to obtain exact, closed-form expressions for the variables of interest in such storage systems using analytic techniques. However, simulation can be a valuable tool in examining such problems, since it represents the only mechanism whereby the effects of a multiplicity of complex non-idealities in the analytic models may be readily assessed. If the non-ideal conditions do not cause the behavior of the system to change appreciably from the ideal case, then the analytic expressions can be of great utility, especially in the early stages of network design¹.

Toward this end, several simulation studies have been conducted using the model of Figure 3.3 where the finite capacity buffer represents the MSC buffer queue in Figure 3.2 and λ of Figure 3.3 is equal to $\sum_{i=1}^{n+1} \gamma_i$, the γ_i 's being those in the model of Figure 3.2. Incoming messages enter the buffer which is shared by all messages seeking transmission on the shared link. Each message enters the same common storage block and reduces a free storage pool by as many units as the message length dictates. When a particular message has been fully transmitted over the shared link, the storage which it had occupied while in the queue is released into the free storage pool. A message which attempts to enter the buffer when the capacity of the free storage pool is less than the message length is assumed permanently lost (the same information will be transmitted with

¹ Martin advocates a similar design philosophy in Chapter 26 of Reference [44].

another message at a future time). The delay characteristics of a MSC with finite capacity buffer have been obtained and the results compared with those which are predicted by the analytic expressions relevant to the previously discussed infinite capacity queueing model. The results of these studies indicate that the average response time characteristic of a wide range of realistic finite-buffer capacity MSC configurations differs from that of the infinite capacity MSC by an amount which is well within acceptable design tolerances, especially in the early stages of system design.

These simulation studies were performed using GPSS/360 in order to determine the error which arises in the second term of expression (3.1) by making the infinite capacity assumption. Buffer storage capacity, the speed of the shared link, and the number of input links to the MSC were all varied, using typical values of γ_i and $\frac{1}{\mu}$. It should be noted that actual error is introduced only in the second term of expression (3.1) and hence does not depend on the α_i ; however the percentage of error in T clearly does depend on the α_i values. Tables 3-A and 3-B depict the percentage error ΔT in the second term only of expression (3.1), as a function of the total number of messages submitted to the MSC concentrator for different Poisson mean arrival rates and shared link speeds. A storage capacity of 4000 bits was used. Larger values of buffer capacity were also examined but 4000 bits of storage proved adequate to justify making the infinite capacity assumption, even in the most

Line Speed, $\alpha_H = 450$ b/sec.; Mean Message Length, $\frac{1}{\mu} = 120$ b/mess; 4000 bit buffer					
$\lambda = 1$ mess/sec.		$\lambda = 3$ mess/sec.			
No. of Arrivals Simulated	Simulated Response Time, T_{sim}	ΔT	No. of Arrivals Simulated	Simulated Response Time, T_{sim}	ΔT
4000	352.5 ms	3.1%	10000	1253.8 ms	6.0%
8000	357.7 ms	1.6%	14000	1272.8 ms	4.6%
12000	361.9 ms	.5%	18000	1252 ms	6.1%
16000	363.8 ms	.0%	22000	1311.4 ms	1.6%
			26000	1324.3 ms	.7%

Table 3-A Results of GPSS Simulation of the Finite Buffer Capacity MSC for a Line Speed of 450 b/s

Line Speed, $\alpha_H = 900$ b/sec.; Mean Message Length, $\frac{1}{\mu} = 120$ b/mess; 4000 bit buffer			
$\lambda = 1$ mess/sec;		$\lambda = 3$ mess/sec;	
No. of Arrivals Simulated	Simulated Response Time, T_{sim}	ΔT	No. of Arrivals Simulated
4000	149.2 ms	3.0%	4000
8000	151.7 ms	1.4%	8000
12000	152.8 ms	.7%	12000
16000	154.0 ms	.0%	16000
			Simulated Response Time, T_{sim}
			211.2 ms
			216.0 ms
			219.1 ms
			221.0 ms
			ΔT
			4.9%
			2.8%
			1.4%
			.5%

Table 3-B Results of GPSS Simulation of the Finite Buffer Capacity MSC for a Line Speed of 900 b/s

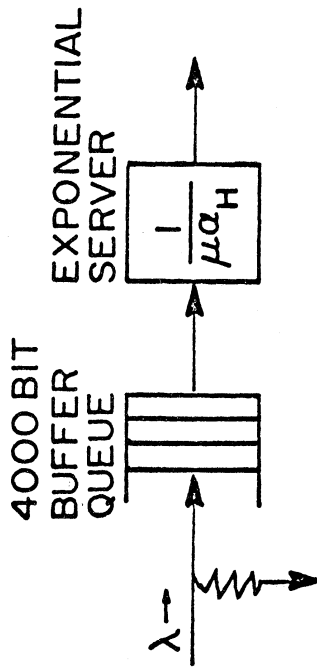
highly congested cases, assuming that errors of less than 1 percent can be considered negligible. The error ΔT is defined to be

$$\left| \frac{T_{\text{sim}} - T_{\infty}}{T_{\infty}} \right| \text{ where}$$

$$T_{\infty} = \frac{1}{\mu\alpha_H} = \frac{1}{\frac{\mu\alpha_H}{1-\lambda}} = \frac{1}{\mu\alpha_H - \lambda}, \text{ the average response time if an}$$

infinite capacity buffer is assumed. T_{sim} is the average response time obtained from the simulation in which the buffer capacity is finite. Only 450 b/s and 900 b/s line speeds were considered since as shared link speed increases, all other parameters being held constant, the error, ΔT , decreases. The $\lambda = 1$ mess/sec Poisson value corresponds to 30 TTY terminals, when each is submitting messages at the fairly typical rate of $\frac{1}{30}$ mess/sec.¹ The $\lambda = 3$ mess/sec corresponds to the case of 90 TTY terminals each of which submits traffic at an average rate of $1/30$ mess/sec. Appendix A contains a listing of the GPSS/360 simulation programs which were used to obtain the values of T_{sim} in Tables 3-A and 3-B.

¹ Values such as these are reported in reference [68].



All messages have mean length,

$$\frac{1}{\mu} = 120 \text{ bits/mess}$$

Mean Service Time

$$= \frac{1}{\mu \alpha_H} = \frac{120}{\alpha_H} \text{ sec/mess}$$

$T_{\infty} = \frac{1}{\mu \alpha_H - \lambda}$, the average response time if the buffer capacity is infinite, sec/mess

$\lambda \triangleq$ Poisson arrival rate of messages, mess/sec

$\alpha_H \triangleq$ Speed of Link, bits/sec

Figure 3.3 Schematic Diagram of Model Used for Simulation of the MSC Concentrator

3.2.2 Model for Message Switching Center, Reverse Network

In the reverse network, the single-stage deconcentration procedures may be represented by the schematic of Figure 3.4

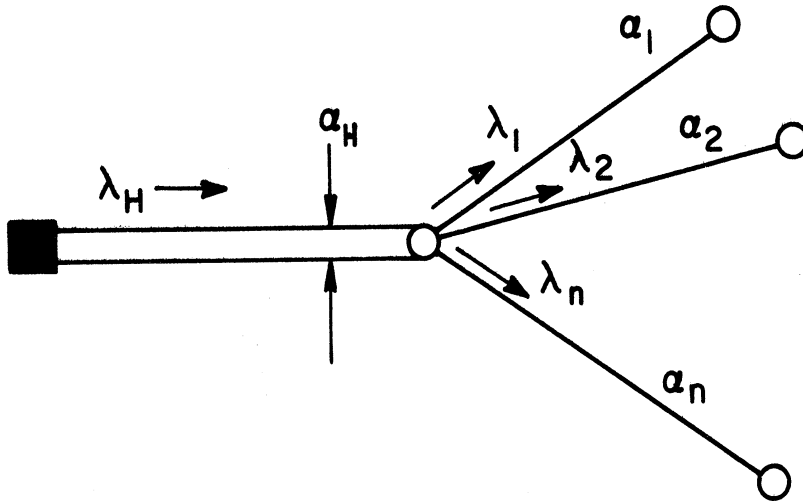


Figure 3.4 Schematic Diagram for the Single Stage MSC, Reverse Network

Since the message traffic on the high speed, shared link is Poisson, a particular message arriving at the MSC will be destined for terminal i ($i=1, \dots, n+1$) with probability p_i where

$$p_i = \frac{\lambda_i}{\lambda_H} = \frac{\lambda_i}{\sum_{j=1}^{n+1} \lambda_j}$$

In order to make use of the exact analytic relationship for the average response time, it is again necessary to assume an infinite buffer storage area. Static buffer allocation is also assumed, that is, a permanent physical buffer is assigned to each of the outgoing lines. The Poisson arrival rate to the i -th line buffer is given by

$p_i \lambda_H = \lambda_i$ and the expression for T becomes

$$T = \frac{1}{\sum_{i=1}^{n+1} \lambda_i} \left[\sum_{i=1}^n \left(\frac{1}{\frac{\alpha_i}{\lambda_i} - 1} \right) + \frac{1}{\frac{\alpha_H}{\lambda_H} - 1} \right] \quad (3.2)$$

Expression (3.2) becomes an approximation to the true average response time if the storage capacity of a line buffer is finite. This time, the difference between the true value and the approximation arises as a consequence of error introduced in the first term of expression (3.2). Since the percentage error in only the first term depends on each of the values of α_i and λ_i , it is not possible to create tables similar to Tables 3-A and 3-B which have the same general degree of utility. Simulation could, however, again be used to estimate the magnitude of these differences for certain specific parameter value ranges of interest in a particular application.

3.3 The Multiplexing Concentrator (MX)

Multiplexing concentrators allow a number of remote terminals to share communication links through the creation of a number of dedicated passages within links. Each passage comprises a part of one of the channels that transmits data over a link. Thus the multiplexor subdivides the link and a fraction of its total capacity for transmission is allocated to each of its constituent passages. The sum of the individual passage capacities within a link is constrained to be equal to

or less than the total transmission capacity of the link. In modelling the frequency and time division multiplexing schemes, a notion of duality between time and frequency similar to that proposed by Bello [73] will be exploited. Bello summarizes this concept:

A concept of duality, called the time frequency duality which is applicable to a class of nets called communication signal processing nets is developed. . . . The usefulness of time-frequency duality stems from the fact that two situations which may be quite distinct physically can have identical behavior patterns except for an interchange of the roles played by time and frequency. As a result the solution to a detection or estimation problem may be found directly from the solution of the dual problem, if known merely by replacing variables and quantities by their duals.

Using this concept of duality, time and frequency division multiplexors can be analyzed with one generalized model in which passage capacity is given whichever interpretation is appropriate.

Figure 3.5 depicts an equivalent structural model for a one stage multiplexor, while Figure 3.6 depicts the tandem multiplexor system.

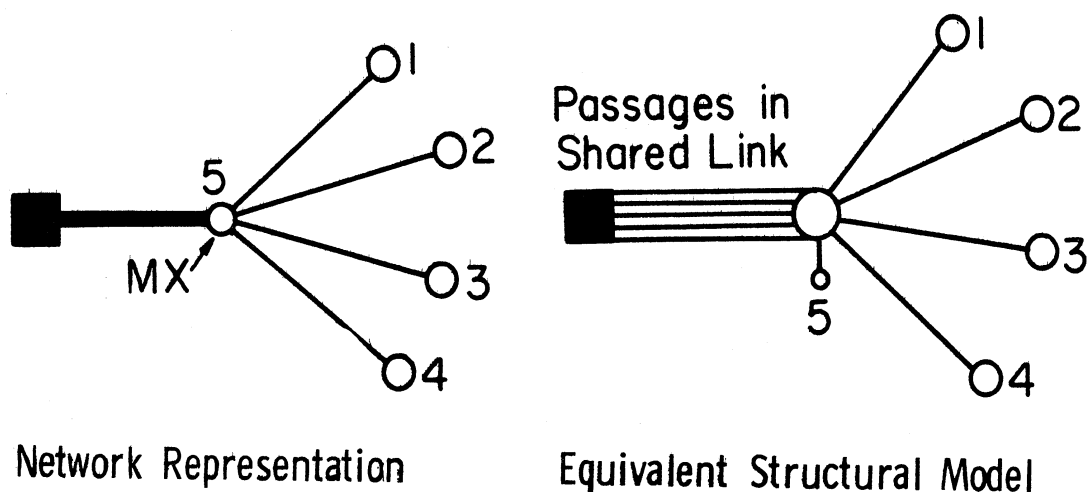
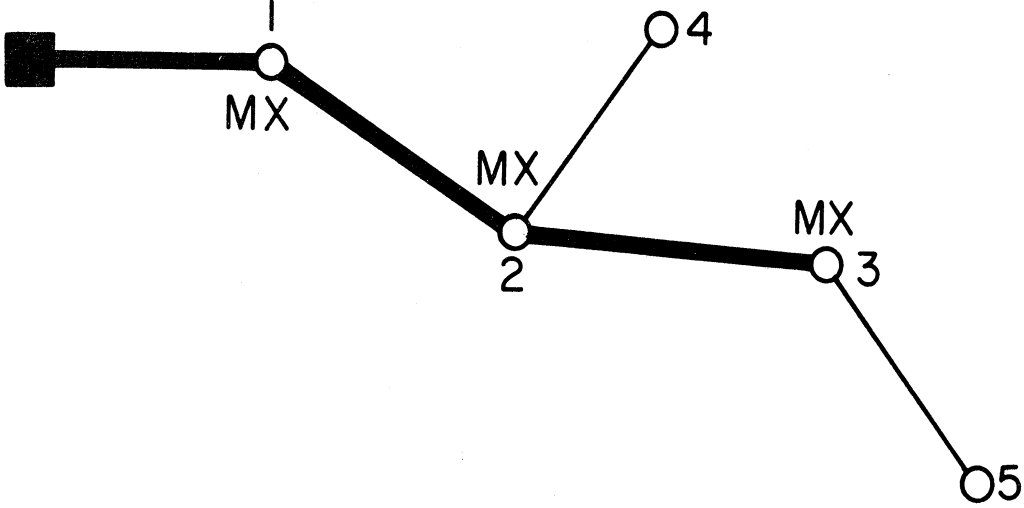
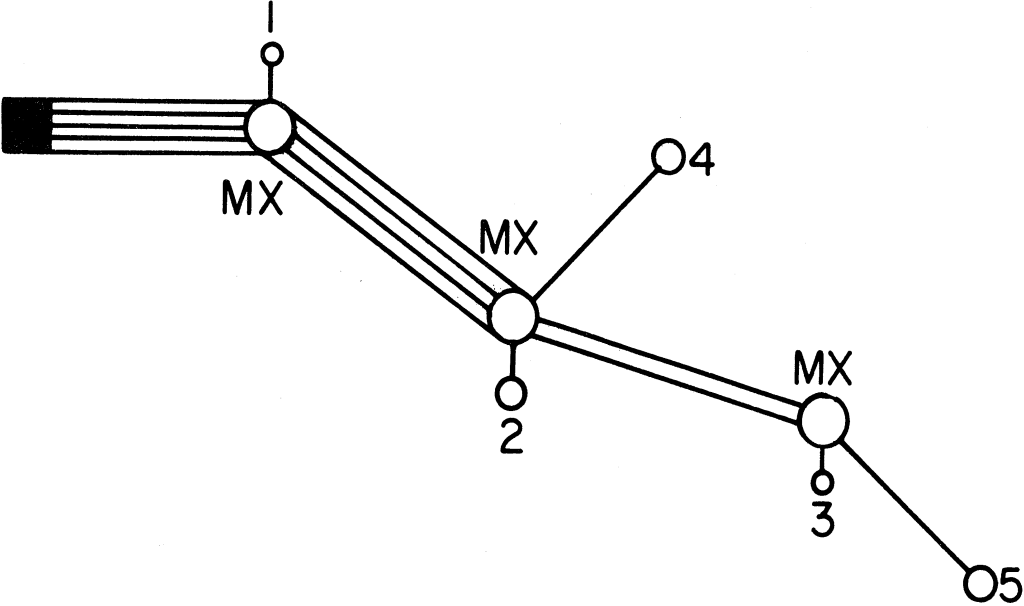


Figure 3.5 A Single Stage Multiplexor



Network Representation



Equivalent Structural Model

Figure 3.6 Tandem Multiplexor Configuration

Since a channel is composed of a chain of these dedicated passages in contiguous links which serve a common terminal, the capacity of a particular channel is determined by the passage which has the minimum transmission capacity of all those comprising the channel. To obtain the average response time of the network of Figure 3.6, the delay components of each of the five independent channels (one originates at each remote terminal) are averaged using expressions (2.8) and (2.9) for the forward and reverse portions of the network, respectively. The value of C_i to be used in these expressions is given by

$$C_i = \min_j \{ \alpha_{ij} \} \quad (3.3)$$

where α_{ij} denotes the transmission capacity of the passage which serves terminal i in link j , and the α_{ij} are only defined for values of i, j such that terminal i uses a passage in link j .¹ The queueing model for the forward network of Figure 3.6 is illustrated in Figure 3.7, where the expressions for the channel capacities are given by

¹ Link i refers to the first link which message traffic originating at terminal i passes over enroute to the central facility.

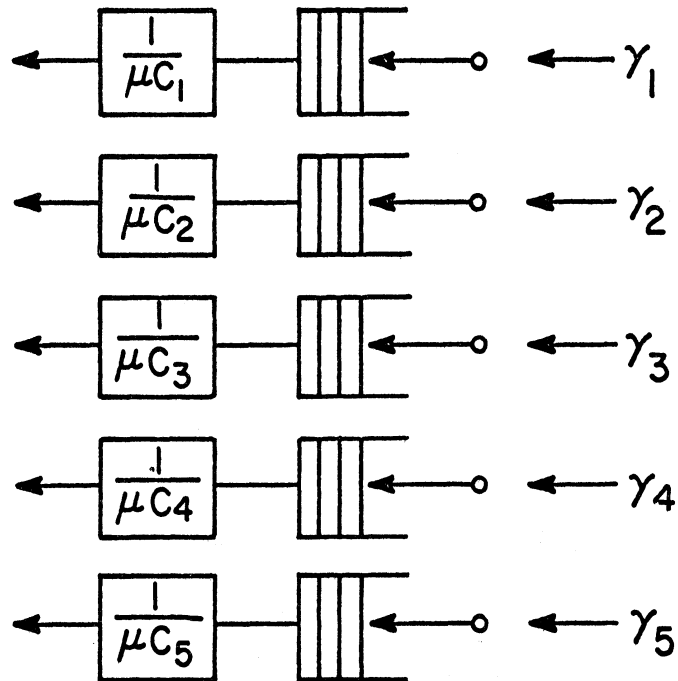


Figure 3.7 Queueing Model for Network of Figure 3.6

$$\begin{aligned}
C_1 &= \alpha_{11} \\
C_2 &= \min\{\alpha_{22}, \alpha_{21}\} \\
C_3 &= \min\{\alpha_{33}, \alpha_{32}, \alpha_{31}\} \\
C_4 &= \min\{\alpha_{44}, \alpha_{42}, \alpha_{41}\} \\
C_5 &= \min\{\alpha_{55}, \alpha_{53}, \alpha_{52}, \alpha_{51}\}
\end{aligned}$$

Thus we can write a general expression for the average response time, T , in an n channel forward network in which MX concentration is used exclusively:

$$T = \frac{1}{\sum_{k=1}^n \gamma_k} \left[\sum_{i=1}^n \left(\frac{1}{\frac{\min\{\alpha_{ij}\}}{\gamma_i} - 1} \right) \right] \quad (3.4)$$

where the α_{ij} are only defined for i, j values such that terminal i uses a passage in link j . This same expression can be used for the reverse network by reinterpretation of the γ_i and $\frac{1}{\mu}$ parameters.

Expressions (3.3) and (3.4) enable one to compute the average response time in a multiplexed tree network when the passage capacity values α_{ij} are known, but nothing has thus far been said about how the values of these important control variables should be established for optimal network performance. It is quite clear from the last two expressions that the assignment of these passage capacity values can have a critical effect on the average response time;

furthermore, the total transmission capacity of any particular link is fixed. Thus, since the cost of a given link is also fixed, it is desirable to allocate, in every link, a fraction of the total capacity to each of the constituent passages in such a fashion that the assignment will yield the minimum average response time. A method to accomplish this objective is now presented.

3.3.1 Optimal Passage Capacity Assignment Procedure, Single Stage Case

Consider the case of the single stage multiplexor shown in Figure 3.8. Treatment of the forward network is implicit, but again the identical procedure can be used for the reverse network if the γ_i and $\frac{1}{u}$ parameters are reinterpreted.

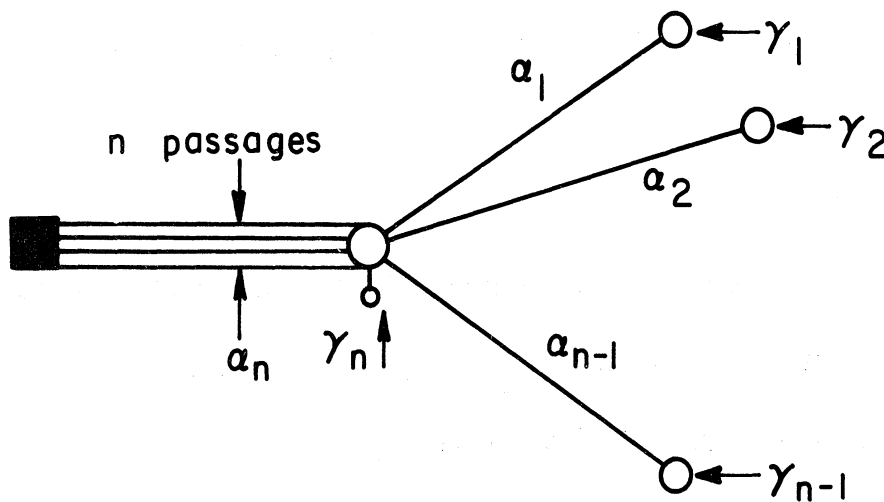


Figure 3.8 The Single Stage Multiplexor Schematic

The constants of the optimization problem are: $\frac{1}{\mu}; \gamma_1, \gamma_2, \dots$

γ_n ; and the link capacity values $\alpha_1, \alpha_2, \dots, \alpha_n$. The independent variables are $\alpha_{1n}, \alpha_{2n}, \dots, \alpha_{nn}$, constrained so that $\sum_{j=1}^n \alpha_{jn} = \alpha_n$. The dependent variable, T, is related to the other parameters by the general expression (3.4) which can be written more precisely as indicated by expression (3.5) below. The precise optimization problem can be formulated as:

Find the values (denoted $\hat{\alpha}_{1n}, \hat{\alpha}_{2n}, \dots, \hat{\alpha}_{nn}$) of the independent variables $\alpha_{1n}, \alpha_{2n}, \dots, \alpha_{nn}$ that minimize

$$T = \frac{1}{\sum_{j=1}^n \gamma_j} \left[\sum_{i=1}^{n-1} \left(\frac{1}{\frac{\min\{\alpha_i, \alpha_{in}\}}{\gamma_i} - 1} \right) + \frac{1}{\frac{\alpha_{nn}}{\gamma_n} - 1} \right] \quad (3.5)$$

subject to $\sum_{i=1}^n \alpha_{in} = \alpha_n$. The solution to this problem is presented in the following theorem.

Theorem 3.1

In a single stage multiplexor concentrator, the values of the independent variables $\alpha_{1n}, \alpha_{2n}, \dots, \alpha_{nn}$ that minimize T as given by expression (3.5) are given by

$$\hat{\alpha}_{in} = \delta_i$$

where

$$\delta_i \triangleq \frac{\gamma_i}{\mu} + \frac{\left[\alpha_n - \sum_{j=1}^n \frac{\gamma_j}{\mu} \right] \sqrt{\frac{\gamma_i}{\mu}}}{\sum_{j=1}^n \sqrt{\frac{\gamma_j}{\mu}}} \quad i=1, \dots, n \quad (3.6)$$

if $\delta_i \leq \alpha_i$ for $i=1,2,\dots,n-1$. Furthermore, the minimum value, \hat{T} , of average response time, T , which results from this optimal assignment under the same hypothesis is given by

$$\hat{T} = \frac{1}{\gamma} \left[\sum_{i=1}^n \left(\frac{\sqrt{\frac{\gamma_i}{\mu}} \sum_{j=1}^n \sqrt{\frac{\gamma_j}{\mu}}}{\alpha_n - \sum_{j=1}^n \frac{\gamma_j}{\mu}} \right) \right] \quad (3.7)$$

where $\gamma = \sum_{j=1}^n \gamma_j$.

Proof: Let us introduce a new objective function

$$T^* = \frac{1}{\gamma} \left[\sum_{i=1}^n \frac{1}{\frac{\alpha_{in}}{\gamma_i} - 1} \right]$$

and find the optimal assignments $\{\alpha_{in}^*\}$ for T^* , subject to $\sum_{i=1}^n \alpha_{in} = \alpha_n$. Since $T^* \leq T$ for any $(\alpha_{1n}, \dots, \alpha_{nn})$ of interest, if $T^*(\alpha_{1n}^*, \dots, \alpha_{nn}^*) = T(\alpha_{1n}^*, \dots, \alpha_{nn}^*)$ then $\{\alpha_{in}^*\}$ will also minimize T . We begin by determining $\{\alpha_{in}^*\}$. Let $\alpha_{in} = \alpha_n f_i$ so that the constant capacity constraint can now be written as $\sum_{i=1}^n f_i = 1$.

Using the method of Lagrange multipliers,¹ we let

$$G = T^* + \beta \left(\sum_{i=1}^n f_i - 1 \right) \quad (3.8)$$

where β is some undetermined constant multiplier. Differentiating G with respect to f_i and setting this result to zero, we have

¹ See Reference [74].

$$\frac{\partial G}{\partial f_i} = -\frac{1}{\gamma} \left[\frac{\frac{\alpha_n}{\phi_i}}{\left(\frac{\alpha_n f_i}{\phi_i} - 1 \right)^2} \right] + \beta = 0$$

where $\phi_i = \frac{\gamma_i}{\mu}$. Solving for f_i :

$$f_i = \frac{\phi_i}{\alpha_n} + \frac{1}{\sqrt{\beta\gamma}} \sqrt{\frac{\phi_i}{\alpha_n}} \quad (3.9)$$

Summing (3.9) on i , we have

$$\sum_{i=1}^n f_i = \sum_{i=1}^n \left[\frac{\phi_i}{\alpha_n} + \frac{1}{\sqrt{\beta\gamma}} \sqrt{\frac{\phi_i}{\alpha_n}} \right] = 1$$

or

$$\sum_{i=1}^n \frac{\phi_i}{\alpha_n} + \sum_{i=1}^n \frac{1}{\sqrt{\beta\gamma}} \sqrt{\frac{\phi_i}{\alpha_n}} = 1$$

which leads to

$$\frac{1}{\sqrt{\beta\gamma}} = \frac{1 - \sum_{i=1}^n \frac{\phi_i}{\alpha_n}}{\sum_{i=1}^n \sqrt{\frac{\phi_i}{\alpha_n}}} \quad (3.10)$$

Substituting (3.10) back into (3.9) we obtain

$$f_i = \frac{\phi_i}{\alpha_n} + \frac{\left(1 - \sum_{j=1}^n \frac{\phi_j}{\alpha_n} \right) \sqrt{\phi_i}}{\sum_{j=1}^n \sqrt{\phi_j}} \quad (3.11)$$

Since $\alpha_{in} = \alpha_n f_i$, we multiply (3.11) by α_n to obtain the optimal values:

$$\alpha_{in}^* = \frac{\gamma_i}{\mu} + \frac{\left[\alpha_n - \sum_{k=1}^n \frac{\gamma_k}{\mu} \right] \sqrt{\frac{\gamma_i}{\mu}}}{\sum_{j=1}^n \sqrt{\frac{\gamma_j}{\mu}}} \quad i = 1, 2, \dots, n$$

which is the desired result for $\{\alpha_{in}^*\}$.

Now since we know that $T^* \leq T$ for any point $(\alpha_{1n}, \dots, \alpha_{nn})$ of interest, the assignment $\{\alpha_{in}^*\}$ which minimizes T^* will also minimize T if $T^*(\alpha_{1n}^*, \dots, \alpha_{nn}^*) = T(\alpha_{1n}^*, \dots, \alpha_{nn}^*)$. But from the hypothesis since $\delta_i \leq \alpha_i$ for $i = 1, \dots, n-1$, it follows immediately that $T^*(\alpha_{1n}^*, \dots, \alpha_{nn}^*) = T(\alpha_{1n}^*, \dots, \alpha_{nn}^*)$. Thus $\{\alpha_{in}^*\}$ also minimizes T and since $\delta_i = \alpha_{in}^*$ for $i = 1, 2, \dots, n$, expression (3.6) is established, completing the first part of the proof. The second part of the proof is obtained directly by substituting the optimal values of $\hat{\alpha}_{in}$ from (3.6) into the general expression (3.5) for the average response time, T , which gives

$$\hat{T} = \frac{1}{\gamma} \sum_{j=1}^n \frac{1}{\left\{ \frac{\frac{\gamma_j}{\mu} + \left(\alpha_n - \sum_{i=1}^n \frac{\gamma_i}{\mu} \right) \sqrt{\frac{\gamma_j}{\mu}}}{\sum_{i=1}^n \sqrt{\frac{\gamma_i}{\mu}}} \right\}^{-1} \frac{\gamma_j}{\mu}}$$

and, upon rearrangement, establishes (3.7), completing the proof.¹

¹ This theorem is similar to Kleinrock's Theorem 4.3 (Reference [25]). The new result developed in this section is an extension of Theorem 3.1 to a more general case where we do not confine the range of the δ_i ($i = 1, \dots, n-1$) and consequently Theorem 3.1 does not apply.

3.3.1.1 Extension of Theorem 3.1

Recalling that the δ_i given by expression (3.6) denote the optimal $\hat{\alpha}_{in}$ as long as no $\delta_i > \alpha_i$ for any $1 \leq i \leq n-1$, note that if $\delta_i > \alpha_i$, the passage capacity for terminal i on link n exceeds the capacity of the passage for the same terminal on link i .¹ It is obvious from expression (3.3), that this excess capacity $\Delta_i = \delta_i - \alpha_i$ cannot improve the performance of the channel serving terminal i over the response obtained if only α_i units of capacity are used. Thus the optimal values $\hat{\alpha}_{in}$ are not given by the δ_i whenever $\delta_i > \alpha_i$ for one or more $i, 1 \leq i \leq n-1$. This is so because the excess capacity Δ_i for all passages such that $\Delta_i > 0$, could be reassigned to other passages for which $\Delta_i < 0$ with resulting improvement in overall response time. However this reassignment may in turn cause some passages on the shared link that are not initially limited by their respective input links to become limited. Thus a general optimization procedure with no constraints on the range of the δ_i consists of a repetitive application of expression (3.6) for successively smaller values of α_n and for only those terminals whose passages on link n were not limited at a previous step. We now present an algorithm to describe this optimal reallocation procedure; it is nondeterministic in the sense that the number of iterations to completion depends on the particular values of the input parameters.

¹ Link i again refers to the first link which message traffic originating at terminal i passes over enroute to the central facility.

3.3.1.1.1 Algorithm Description¹

B will denote the set of all terminals for which a shared link passage capacity was assigned in excess of its input link capacity at some prior iteration. \bar{B} denotes the complementary set, such that $\bar{B} \cup B$ denotes the set of all remote terminals in the network. It is understood that terminal n , the one located at the concentrator site, will always be in the set \bar{B} . The variable k is simply an index on the iterations of the procedure. In step 2 of the algorithm to be presented, B_k is the set of all terminals which are limited by their respective input links at the k th iteration; we define M to be largest value of k for which $B_k \neq \phi$ in Step 2. Thus $B_M \neq \phi$ and $B_{M+1} = \phi$.

Step 1: Using expression (3.6) determine values for α_{in} , $i=1, \dots, n$. Set $k=0$ and $B=\phi$, the null set.

Step 2: $k=k+1$; determine the set of terminals for which $\alpha_j < \alpha_{jn}$, $j \in \bar{B}$, where the α_{jn} are values determined at the previous step. Denote this set of such terminals by B_k for the k -th iteration. Set $\alpha_{jn} = \alpha_j$ for all terminals $j \in B_k$. Whenever $B_k = \phi$, set $M=k-1$ and terminate the procedure. Thus termination occurs when, for terminals in \bar{B} , no passage capacity values were assigned in excess of their respective input link capacity values at an immediately preceding step.

¹ A computational module called ASSIGN will later be introduced to denote all of the computations performed by this algorithm.

Step 3: Let $B = B_1 \cup B_2 \cup \dots \cup B_k$. Compute

$$\alpha_{jn} = \frac{\gamma_j}{\mu} + \frac{\left(\alpha_n - \sum_{i \in B} \alpha_i - \sum_{i \in \bar{B}} \frac{\gamma_i}{\mu} \right) \sqrt{\frac{\gamma_j}{\mu}}}{\sum_{i \in \bar{B}} \sqrt{\frac{\gamma_i}{\mu}}} \quad (3.12)$$

for $j \in \bar{B}$ and go to Step 2.

If we let $\hat{\alpha}_{jn}$, $j \in \bar{B}$, denote the assignment of passage capacity values given by expression (3.12) after the final iteration, then an exact expression for the minimum average response time, \hat{T} given by

$$\hat{T} = \frac{1}{\gamma} \left[\sum_{i \in B} \left(\frac{1}{\frac{\alpha_i}{\frac{\gamma_i}{\mu}} - 1} \right) + \sum_{j \in \bar{B}} \left(\frac{1}{\frac{\hat{\alpha}_{jn}}{\frac{\gamma_j}{\mu}} - 1} \right) \right] \quad (3.13)$$

and it is understood that the single-passage channel originating at the concentrator site is always in the set \bar{B} . A proof of this algorithm may be found in the next section. The proof consists of establishing that it is not possible to improve upon the solution obtained by the algorithm just presented.

3.3.1.1.2 Proof of Algorithm

The proof will consist of establishing that it is not possible to improve the solution generated by the algorithm. We can write expression (3.13) for \hat{T} as follows

$$\hat{T} = \frac{1}{\gamma} \left[\sum_{i \in B} \left(\frac{\phi_i}{\alpha_i - \phi_i} \right) + \sum_{i \in \bar{B}} \left(\frac{\phi_i}{C_i - \phi_i} \right) \right] \quad (3.14)$$

where $\phi_i = \frac{\gamma_i}{\mu}$ and

$$C_i = \phi_i + \frac{\left[\alpha_n - \sum_{j \in B} \alpha_j - \sum_{j \in \bar{B}} \phi_j \right] \sqrt{\phi_i}}{\sum_{j \in B} \sqrt{\phi_j}} \quad (3.15)$$

We recall that if any passage capacity assignment is made larger than α_i for $i \in B$ then the average response time must increase from the value indicated in expression (3.14). This happens because the excess capacity above α_i does not reduce T below the value it would have if only α_i units were used. Furthermore, since the total available capacity for all passages is constrained to be constant, existence of any such excess implies that some other passage capacity is under assigned as a consequence. Thus we will give further consideration only to those cases in which the passage capacity assignments for terminals $i \in B$ are reduced from the value α_i . Let

$\Delta T_i^+ \triangleq$ the increase in T if the passage capacity of the i -th terminal of the set B is reduced by an amount x_i ($x_i \geq 0$ for reason cited above)

$$\Delta T_i^+ = \frac{1}{\gamma} \left[\frac{\phi_i}{\alpha_i - x_i - \phi_i} - \frac{\phi_i}{\alpha_i - \phi_i} \right] = \frac{x_i \phi_i}{\gamma (\alpha_i - \phi_i) (\alpha_i - x_i - \phi_i)}$$

$$\Delta T^+ = \sum_{i \in B} \Delta T_i^+ = \frac{1}{\gamma} \sum_{i \in B} \frac{x_i \phi_i}{(\alpha_i - \phi_i)(\alpha_i - x_i - \phi_i)}$$

Now similarly let

$\Delta T^- \triangleq$ the maximum decrease in T possible if all $K = \sum_{i \in B} x_i$ units of capacity are optimally reallocated to the passages for terminals in the set \bar{B} .¹ This ΔT^- actually represents an upper bound on the magnitude of the decrease obtainable. It is only possible to realize the full amount of this decrease if the reallocation does not limit any terminals which were in \bar{B} before the reallocation. However, since the magnitude of any reduction in T may not in any case exceed ΔT^- , we use ΔT^- throughout the proof in showing that $\Delta T^+ \geq \Delta T^-$.

$$\Delta T^- = \frac{1}{\gamma} \left[\sum_{i \in \bar{B}} \left(\frac{\phi_i}{C_i - \phi_i} \right) - \sum_{i \in \bar{B}} \frac{\phi_i}{C'_i - \phi_i} \right] \quad (3.16)$$

where the C_i values are given by expression (3.15) and

¹ Optimal reallocation implies that the new passage capacity assignments C'_i for terminals $i \in \bar{B}$ are given by the familiar form:

$$C'_i = \phi_i + \frac{(\alpha_n - \sum_{j \in B} \alpha_j - \sum_{j \in \bar{B}} \phi_j + K) \sqrt{\phi_i}}{\sum_{j \in \bar{B}} \sqrt{\phi_j}}$$

$$C'_i = \phi_i + \frac{\left(\alpha_n - \sum_{j \in B} \alpha_j - \sum_{j \in \bar{B}} \phi_j + K \right) \sqrt{\phi_i}}{\sum_{j \in \bar{B}} \sqrt{\phi_j}} \quad (3.17)$$

and

$$K = \sum_{i \in B} x_i \quad (3.18)$$

Expression (3.16) can therefore be rewritten as follows:

$$\Delta T^- = \frac{1}{\gamma} \left[\sum_{i \in \bar{B}} \frac{\phi_i \sum_{k \in \bar{B}} \sqrt{\phi_k}}{\left(\alpha_n - \sum_{k \in B} \alpha_k - \sum_{j \in \bar{B}} \phi_j \right) \sqrt{\phi_i}} - \sum_{i \in \bar{B}} \frac{\phi_i \sum_{k \in \bar{B}} \sqrt{\phi_k}}{\left(\alpha_n - \sum_{k \in B} \alpha_k - \sum_{j \in \bar{B}} \phi_j + K \right) \sqrt{\phi_i}} \right]$$

$$\text{Let } N_1 = \sum_{k \in \bar{B}} \sqrt{\phi_k}$$

$$S_1 = \alpha_n - \sum_{k \in B} \alpha_k - \sum_{j \in \bar{B}} \phi_j$$

then we have

$$\begin{aligned} \Delta T^- &= \frac{1}{\gamma} \left[\sum_{i \in \bar{B}} \left(\frac{\sqrt{\phi_i} N_1}{S_1} - \frac{\sqrt{\phi_i} N_1}{S_1 + K} \right) \right] \\ &= \frac{1}{\gamma} \left[\sum_{i \in \bar{B}} \left(\frac{\sqrt{\phi_i} N_1 (S_1 + K) - S_1 N_1 \sqrt{\phi_i}}{S_1 (S_1 + K)} \right) \right] \end{aligned}$$

$$= \frac{1}{\gamma} \left[\sum_{i \in \bar{B}} \frac{KN_1 \sqrt{\phi_i}}{S_1(S_1 + K)} \right] = \frac{1}{\gamma} \left[\frac{N_1^2 K}{S_1(S_1 + K)} \right]$$

but from (3.18) we may replace K by $\sum_{i \in B} x_i$ so that we have

$$\Delta T^- = \frac{N_1^2}{\gamma S_1(S_1 + K)} \sum_{i \in B} x_i$$

Recall that at successive iterations of step 2 we defined the sets B_k for $k=1, 2, \dots, M$ where M is the index for B_M , the last non-empty set formed. The set B was formed by taking the union of the B_k for $k=1, \dots, M$. Hence it is possible to write ΔT^- as follows:

$$\Delta T^- = \frac{N_1^2}{\gamma S_1(S_1 + K)} \left[\sum_{i \in B_1} x_i + \sum_{i \in B_2} x_i + \dots + \sum_{i \in B_M} x_i \right] \quad (3.19)$$

It is possible to write ΔT^+ in a similar fashion

$$\Delta T^+ = \frac{1}{\gamma} \left[\sum_{i \in B_1} \frac{x_i \phi_i}{(\alpha_i - \phi_i)(\alpha_i - x_i - \phi_i)} + \sum_{i \in B_2} \frac{x_i \phi_i}{(\alpha_i - \phi_i)(\alpha_i - x_i - \phi_i)} + \dots \right. \\ \left. + \sum_{i \in B_M} \frac{x_i \phi_i}{(\alpha_i - \phi_i)(\alpha_i - x_i - \phi_i)} \right] \quad (3.20)$$

Before attempting to compare ΔT^+ and ΔT^- given by expressions (3.19) and (3.20) it is necessary to establish several inequalities.

Recall that we know from Step 2 of the algorithm for $k=1$

$$\phi_i + \frac{\left(\alpha_n - \sum_{j=1}^n \phi_j \right) \sqrt{\phi_i}}{\sum_{j=1}^n \sqrt{\phi_j}} > \alpha_i \text{ for } i \in B_1$$

which can be rewritten as

$$\frac{\left(\alpha_n - \sum_{j=1}^n \phi_j \right)}{\sum_{j=1}^n \sqrt{\phi_j}} > \frac{\alpha_i - \phi_i}{\sqrt{\phi_i}} \text{ for } i \in B_1 \quad (3.21)$$

A similar inequality can be obtained by noting that from Step 2 of the algorithm for $k = 2$, the following relation holds true

$$\phi_i + \frac{\left(\alpha_n - \sum_{j \in B_1} \alpha_j - \sum_{j \in \bar{B}_1} \phi_j \right) \sqrt{\phi_i}}{\sum_{j \in \bar{B}_1} \sqrt{\phi_j}} > \alpha_i \text{ for } i \in B_2$$

which gives

$$\frac{\left(\alpha_n - \sum_{j \in B_1} \alpha_j - \sum_{j \in \bar{B}_1} \phi_j \right)}{\sum_{j \in \bar{B}_1} \sqrt{\phi_j}} > \frac{\alpha_i - \phi_i}{\sqrt{\phi_i}} \text{ for } i \in B_2 \quad (3.22)$$

Extension to the general inequality follows similarly from a knowledge of what happens at Step 2 of the algorithm when $k = q+1$. Let $\beta_q = B_1 \cup B_2 \cup \dots \cup B_q$ denote the set B just prior to the inclusion of the subset B_{q+1} . Then we have

$$\frac{\alpha_n - \sum_{j \in \beta_q} \alpha_j - \sum_{j \in \bar{\beta}_q} \phi_j}{\sum_{j \in \bar{\beta}_q} \sqrt{\phi_j}} > \frac{\alpha_i - \phi_i}{\sqrt{\phi_i}} \quad \text{for } i \in B_{q+1} \quad (3.23)$$

and $0 \leq q \leq M-1$.

This set of inequalities is not alone sufficient for our comparison of ΔT^+ and ΔT^- , however. We also need a relationship between the magnitudes of the following quantities:

$$\frac{\alpha_n - \sum_{j \in \beta_q} \alpha_j - \sum_{j \in \bar{\beta}_q} \phi_j}{\sum_{j \in \bar{\beta}_q} \sqrt{\phi_j}} \quad \text{and} \quad \frac{\alpha_n - \sum_{j \in \beta_q \cup B_{q+1}} \alpha_j - \sum_{j \in \bar{\beta}_q \cup B_{q+1}} \phi_j}{\sum_{j \in \beta_q \cup B_{q+1}} \sqrt{\phi_j}}$$

which holds for all $0 \leq q \leq M-1$.

Let $A_q \triangleq \sum_{i \in \beta_q} \alpha_i$ so that we can write

$$A_{q+1} - A_q = \sum_{i \in B_{q+1}} \alpha_i \quad (3.24)$$

Similarly, let $Z_q = \sum_{i \in \bar{\beta}_q} \phi_i$ so that

$$Z_q - Z_{q+1} = \sum_{i \in B_{q+1}} \phi_i \quad (3.25)$$

Finally, let $D_q = \sum_{i \in \beta_q} \sqrt{\phi_i}$ so that

$$D_q - D_{q+1} = \sum_{i \in B_{q+1}} \sqrt{\phi_i} \quad (3.26)$$

From expression (3.23) we know that

$$\frac{\alpha_n - A_q - Z_q}{D_q} - \frac{\alpha_i - \phi_i}{\sqrt{\phi_i}} > 0 \quad \text{for all } i \in B_{q+1}$$

and $0 \leq q \leq M-1$. Hence we have

$$\frac{\sqrt{\phi_i} [\alpha_n - A_q - Z_q] - D_q [\alpha_i - \phi_i]}{D_q \sqrt{\phi_i}} > 0 \quad \text{for all } i \in B_{q+1}$$

and $0 \leq q \leq M-1$. Since the denominator can never be nonpositive, we have

$$[\alpha_n - A_q - Z_q] \sqrt{\phi_i} - D_q [\alpha_i - \phi_i] > 0 \quad \text{for all } i \in B_{q+1}$$

and $0 \leq q \leq M-1$. If we sum this last expression over all $i \in B_{q+1}$ we have, for $0 \leq q \leq M-1$

$$[\alpha_n - A_q - Z_q] \sum_{i \in B_{q+1}} \sqrt{\phi_i} - D_q \sum_{i \in B_{q+1}} \alpha_i + D_q \sum_{i \in B_{q+1}} \phi_i > 0$$

Now we recall relationships (3.24) - (3.26) and modify this last expression accordingly, which gives

$$[\alpha_n - A_q - Z_q] [D_q - D_{q+1}] - D_q [A_{q+1} - A_q] + D_q [Z_q - Z_{q+1}] > 0$$

for all $0 \leq q \leq M-1$.

Simplification of this expression gives, for $0 \leq q \leq M-1$

$$-D_{q+1}(\alpha_n - A_q - Z_q) > -D_q(\alpha_n - A_{q+1} - Z_{q+1})$$

or
$$\frac{\alpha_n - A_q - Z_q}{D_q} < \frac{\alpha_n - A_{q+1} - Z_{q+1}}{D_{q+1}} \quad \text{which enables}$$

us to write in fully expanded form the relationship which is sought

$$\frac{\alpha_n - \sum_{i \in \beta_q} \alpha_i - \sum_{j \in \beta_q} \phi_j}{\sum_{j \in \beta_q} \sqrt{\phi_j}} < \frac{\alpha_n - \sum_{j \in \beta_q \cup B_{q+1}} \alpha_j - \sum_{j \in \beta_q \cup B_{q+1}} \phi_j}{\sum_{j \in \beta_q \cup B_{q+1}} \sqrt{\phi_j}} \quad (3.27)$$

Expression (3.27) holds for $0 \leq q \leq M-1$. Note that in particular when $q=M-1$, the RHS of (3.27) is equal to $\frac{S_1}{N_1}$ where S_1 and N_1 are the same terms that appear in expressions (3.19) and (3.20) for ΔT^- and ΔT^+ , respectively. Furthermore, the LHS of (3.27) for $0 \leq q \leq M-1$ must be less than $\frac{S_1}{N_1}$. The LHS of (3.27) also appears in the inequality (3.23) which enables us to write:

$$\frac{\alpha_i - \phi_i}{\sqrt{\phi_i}} < \frac{S_1}{N_1} \quad \text{for } i \in B_{q+1} \quad (3.28)$$

Expanding (3.28) one step further to include the remaining parameters which appear in expressions (3.19) and (3.20), we have

$$\frac{\sqrt{\phi_i}}{\alpha_i - x_i - \phi_i} \geq \frac{\sqrt{\phi_i}}{\alpha_i - \phi_i} > \frac{N_1}{S_1} \geq \frac{N_1}{S_1 + K} \quad \text{for } i \in B_{q+1} \quad (3.29)$$

Finally, having established (3.29), we now establish that $\Delta T^+ \geq \Delta T^-$ which assures that no solution better than the one obtained by the algorithm does exist.

Considering only the k-th subset of B in (3.19) and (3.20),
define

$$\Delta T^+(B_k) = \frac{1}{\gamma} \left[\sum_{i \in B_k} \frac{x_i \phi_i}{(\alpha_i - \phi_i)(\alpha_i - x_i - \phi_i)} \right]$$

$$\text{and } \Delta T^-(B_k) = \frac{1}{\gamma} \frac{N_1^2}{S_1(S_1+K)} \left[\sum_{i \in B_k} x_i \right]$$

We can write the difference expression

$$\Delta T^+(B_k) - \Delta T^-(B_k) \stackrel{\Delta}{=} \frac{1}{\gamma} \sum_{i \in B_k} \left[x_i \left(\frac{\gamma \phi_i \sqrt{\phi_i}}{(\alpha_i - x_i - \phi_i)(\alpha_i - \phi_i)} - \frac{N_1}{S_1} \frac{N_1}{S_1 + K} \right) \right] \quad (3.30)$$

Use of expression (3.29) for $k=q+1$ guarantees that $\Delta T^+(B_k) \geq \Delta T^-(B_k)$. Since (3.29) holds for $0 \leq q \leq M-1$ the difference expression (3.30) must be nonnegative for every subset of B in expressions (3.19) and (3.20), and

$$\Delta T^+ \geq \Delta T^-$$

which completes the proof.

3.3.2 Optimal Passage Capacity Assignment Procedure, Tandem Multiplexor Case

The results of the single stage case can be extended to the case of tandem multiplexors by treating the tandem system in hierarchical fashion. Consider the general tandem configuration

of Figure 3.9. Let us define the concept of concentration level as follows: A link k is said to be located at concentration level j ($j \geq 1$) if the message traffic entering link k must be transmitted over exactly j links (including link k) of the network enroute to the central facility.

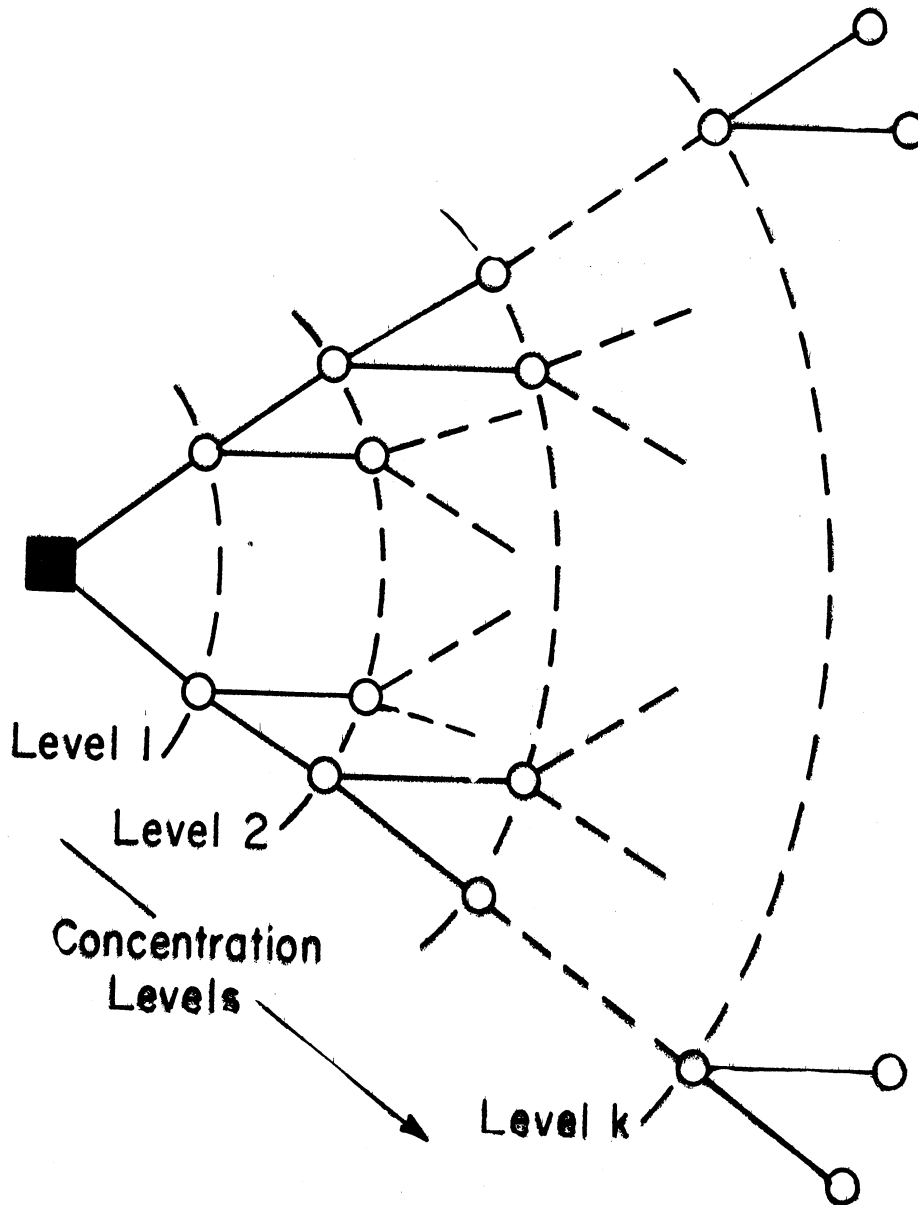


Figure 3.9 Generalized Tandem Multiplexor Schematic

Clearly, each branch of the tree constitutes a multipoint segment which can be analyzed independently. The procedure for handling the multi-branch network of Figure 3.9 will consist of an independent treatment of each of the constituent multipoint segments.

Let us initially consider only the subnetwork formed in a given segment by the highest level shared link which exists and all other links in the same segment at higher levels of concentration. Observe that we can use the single-multiplexor algorithm of section 3.3.1.1.1 to optimally assign passage capacities for the shared link in this subnetwork. Once these passage capacity values have been determined, each of the passages then functions exactly like an input link to the multiplexor at the next lower level of concentration.

Whenever passage capacity values have been optimally assigned on all links at concentration level $j+1$ which submit traffic to a multiplexed link at level j , then it is possible by using the single multiplexor algorithm to optimally assign all the passage capacities in a larger subnetwork consisting of the shared link at level j as well as all higher level links in the same segment. The algorithm presented in this section simply describes a systematic procedure for assigning passage capacities in the shared links of the network in a sequential order which assures that all shared links at level $j+1$ are considered before any attempt is made to consider those at level j .

3.3.2.1 Description of the Tandem Algorithm and Discussion¹

We now present an algorithm which enables one to optimally assign passage capacities and hence to calculate the minimum average response time in an arbitrary tree structured network where MX concentrators are used exclusively. It is assumed that all link capacity values, message arrival rates, and mean lengths are given.

In order to simplify the explanation of the tandem algorithm, let us define a computational module called ASSIGN which implements the optimal single-multiplexor assignment procedure of Section 3.3.1.1.1. From Figure 3.8 recall the inputs which were needed for such computations:

- n : The number of remote terminals
- $(\alpha_1, \alpha_2, \dots, \alpha_n)$: The link capacity values for links
1, ..., n
- $(\gamma_1, \gamma_2, \dots, \gamma_n)$: The message arrival rates
- $\frac{1}{\mu}$: The mean message length

These inputs were used by the procedure to calculate outputs consisting of the optimal passage capacity assignments for the shared link. We will denote the dependence on the input parameters by referring to the module as ASSIGN (NRT, ALFA, GAM, INVMU) where

NRT \triangleq The number of remote terminals, n

¹ A computational module called TANDMX will later be introduced to denote all of the computations performed by this algorithm.

ALFA \triangleq	A one-dimensional vector having n components $\alpha_1, \alpha_2, \dots, \alpha_n$, the link capacities for links 1, ..., n
GAM \triangleq	A one-dimensional vector having n components $\gamma_1, \gamma_2, \dots, \gamma_n$, the message arrival rates
INVMU \triangleq	The mean message length, $\frac{1}{\mu}$

Hence, it is understood that execution of the module will result in all computations described in Section 3.3.1.1.1 being performed for the currently defined set of inputs. The results of the computations will be the optimal passage capacity values on the shared link in question.

Define S_x as the set of all terminals which share link x; the cardinality of this set is $|S_x|$. Define $P_i = \{p_1, p_2, \dots\}$ as the set of all links at the next higher level of concentration which feed traffic directly into link i; the cardinality of this set is $|P_i|$.

Let h be an index on segments of the network and let NSG denote the total number of segments. Form a vector, SL, of all shared links in the network so that $SL(h, k, \ell)$ denotes the ℓ -th shared link at concentration level k in segment h. The index on concentration levels, k, has a range which depends on the segment under consideration, so that $1 \leq k \leq NK_h$ where NK_h denotes the highest level of concentration at which a shared link may be found in segment h. The index, ℓ , on shared links at a particular concentration level and in a particular segment has a range which depends on k and h so that $1 \leq \ell \leq NL_{k, h}$ where $NL_{k, h}$ denotes the total number of shared links at level k in segment h.

We now present the algorithm.

- Step 1: Set $\alpha_{j,j} = \alpha_j$ for all nonshared links j in the network:
set $h = 0$.
- Step 2: $h = h + 1$; Set $k = NK_h + 1$ where k is an index on the levels of concentration in segment h .
- Step 3: $k = k - 1$; Set $\ell = 0$, where ℓ is simply an index on the shared links at concentration level k in segment h .
- Step 4: $\ell = \ell + 1$; Set $i = SL(h, k, \ell)$, the ℓ -th shared link at concentration level k in segment h . The optimal passage capacity assignments $\alpha_{x,j}$ are already defined for links $j \in P_i$ and for terminals $x \in S_j$. Each passage on these links is functionally equivalent to a separate input to the multiplexor servicing link i . Hence, form the vector ALFA as follows:
- Step 5: Define a set $\theta = A_1 \cup A_2 \cup \dots \cup A_{|P_i|}$ where $A_j \triangleq \{\alpha_{x,p_j}\}$ for terminals $x \in S_{p_j}$ and all $p_j \in P_i$. Thus on each link p_j which feeds node i there are $|S_{p_j}|$ passages, each of which looks like a separate input to node i . The cardinality of θ will therefore be $|S_i| - 1$. Form the first $|S_i| - 1$ components of the vector ALFA by ordering the passage capacities of θ according to increasing value of the remote terminal index x for all terminals $x \in S_{p_1} \cup S_{p_2} \cup \dots \cup S_{p_{|P_i|}}$. Set the last component $ALFA(|S_i|)$ equal to the capacity of link i .
- Similarly, form the vector GAM as follows:

Step 6: Define a set of arrival rates $\Omega = B_1 \cup B_2 \cup \dots \cup B_{|P_i|}$ where $B_j \triangleq \{\gamma_x\}$ for all terminals $x \in S_{p_j}$ and all $p_j \in P_i$. Thus the set Ω contains the arrival rate for every terminal that shares any link p_j which is incident to node i . Form the first $|S_i| - 1$ components of the vector GAM by ordering the elements of Ω according to increasing value of the remote terminal index x for all $x \in S_{p_1} \cup S_{p_2} \cup \dots \cup S_{p_{|P_i|}}$. Set the last component $\text{GAM}(|S_i|) = \gamma_i$.

Step 7: Set $\text{NRT} = |S_i|$ and $\text{INVMU} = \frac{1}{\mu}$. Execute ASSIGN (NRT, ALFA, GAM, INVMU)

Step 8: For each remote terminal $x \in S_i$ define an arbitrarily ordered set:

$$Y_x \triangleq \{\text{all links which contain a passage serving remote terminal } x \text{ and are located at a higher level of concentration than link } i.\}$$

Then for each remote terminal $x \in S_i$, set $a_{x,y} = a_{x,i}$ for all $y \in Y_x$. Thus all passages serving a particular remote terminal x which shares link i are assigned a capacity equal to the passage capacity for terminal x on link i .

Step 9: Is $\ell = \text{NL}_{k,h}$? If not, go to step 4.

Step 10: Is $k = 1$? If not, go to step 3.

Step 11: Is $h = \text{NSG}$? If not, go to step 2; otherwise terminate the procedure.

Note that upon the completion of step 8 of the procedure, all passage capacities in the subnetwork which includes the shared link $SL(h, k, \ell)$ and all higher level links in the same segment are optimally assigned. This follows immediately from the fact that passages which submit traffic to the multiplexor servicing shared link $SL(h, k, \ell)$ function exactly as separate input links to the multiplexor, and as a consequence of the previously established optimality of the procedure ASSIGN.

At each new iteration of steps 5-8 a larger subnetwork within a segment is considered; at each iteration of step 11 an additional segment is considered. When step 11 is executed with $h = NSG$, all passage capacities in the network have been optimally assigned. Furthermore, after this terminal step, all passages which comprise each channel of the network have been assigned the same capacity. Thus all channel capacities are also optimally assigned. Symbolically, if C_i denotes the optimal capacity of the channel originating at remote terminal i , then $C_i = \hat{\alpha}_{i,j}$ for all $j \in Q$ where

$\hat{\alpha}_{i,j}$ denotes the optimal capacity of the passage serving remote terminal i on link j .

Q set of all links which contain a passage serving remote terminal i .

We now cite an example illustrating the usage of this procedure.

3.3.2.2 Example of Use of the Algorithm

Consider the single-segment network of Figure 3.10. Let the

capacity of all links in the network be 100 char/sec. Furthermore, let $\gamma_i = 1$ mess/sec, $i=1,2,\dots,7$ and let $\frac{1}{\mu} = 10$ char/mess. The computation proceeds as follows:

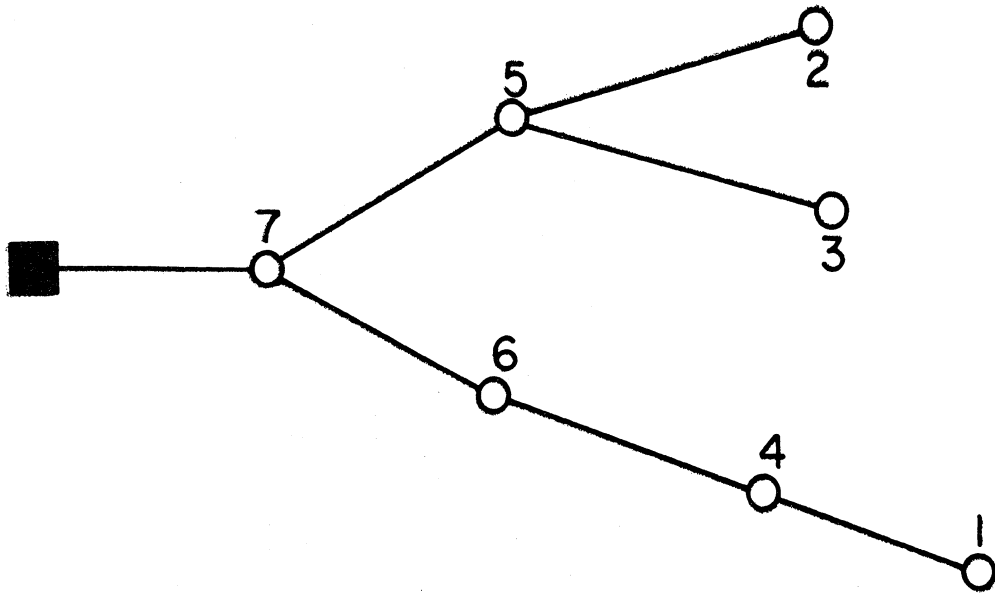


Figure 3.10 Example Network

Since $NSG = 1$ we have $NK_1 = 3$, $NL_{1,1} = 1$, $NL_{2,1} = 2$, and $NL_{3,1} =$

1. We form

$$SL(1, k, \ell) = \begin{matrix} & & \ell \rightarrow \\ & & 1 & 2 \\ 1 & & \left[\begin{array}{cc} 7 & \\ & \end{array} \right] \\ k & 2 & \left[\begin{array}{cc} 5 & 6 \\ & \end{array} \right] \\ \downarrow & 3 & \left[\begin{array}{cc} 4 & \\ & \end{array} \right] \end{matrix}$$

Step 1: $h=0$

$$\alpha_{2,2} = \alpha_2 = 100 \text{ c/s}$$

$$\alpha_{3,3} = \alpha_3 = 100 \text{ c/s}$$

$$\alpha_{1,1} = \alpha_1 = 100 \text{ c/s}$$

Step 2: $h=1; k=4$

Step 3: $k=3; \ell=0$

Step 4: $\ell=1; i=4$

$$\alpha_{2,5} = 33.33 \text{ c/s}$$

$$\alpha_{3,5} = 33.33 \text{ c/s}$$

$$\alpha_{5,5} = 33.33 \text{ c/s}$$

Step 8: $Y_2 = \{2\}, Y_3 = \{3\}$

$$\alpha_{2,2} = \alpha_{2,5} = 33.33 \text{ c/s}$$

$$\alpha_{3,3} = \alpha_{3,5} = 33.33 \text{ c/s}$$

Step 4: $\ell = 2; i = 6$

Step 5: $ALFA(1) = \alpha_{1,4} = 50 \text{ c/s}$ $ALFA(2) = \alpha_{4,4} = 50 \text{ c/s}$
 $ALFA(3) = 100 \text{ c/s}$

Step 6: $GAM(1) = \gamma_1 = 1 \text{ m/s}$
 $GAM(2) = \gamma_4 = 1 \text{ m/s}$
 $GAM(3) = \gamma_6 = 1 \text{ m/s}$

Step 7: $NRT = 3$
 $INVMU = 10 \text{ c/m}$

Execute ASSIGN(NRT, ALFA, GAM, INVMU)

The module ASSIGN computes the following:

$$\alpha_{1,6} = 33.33 \text{ c/s}$$

$$\alpha_{4,6} = 33.33 \text{ c/s}$$

$$\alpha_{6,6} = 33.33 \text{ c/s}$$

Step 8: $Y_1 = \{1, 4\}$ $Y_4 = \{4\}$

$$\alpha_{1,4} = \alpha_{1,1} = \alpha_{1,6} = 33.33 \text{ c/s}$$

$$\alpha_{4,4} = \alpha_{4,6} = 33.33 \text{ c/s}$$

Step 10: $k \neq 1$, so go to step 3

Step 3: $k = 1; \ell = 0$

Step 4: $l = 1; i = 7$

Step 5: $ALFA(j) = \alpha_{j,5} = 33.33 \text{ c/s}$ for $j = 2, 3, 5$

$ALFA(j) = \alpha_{j,6} = 33.33 \text{ c/s}$ for $j = 1, 4, 6$

$ALFA(7) = 100 \text{ c/s}$

Step 6: $GAM(j) = \gamma_j = 1 \text{ m/s}$ for $j = 1, \dots, 7$

Step 7 $NRT = 7$

$INVMU = 10 \text{ c/m}$

Execute ASSIGN(NRT, ALFA, GAM, INVMU)

The module ASSIGN computes the following:

$\alpha_{j,7} = 14.28 \text{ c/s}$ $j = 1, \dots, 7$

Step 8: $Y_1 = \{1, 4, 6\}$ $Y_2 = \{2, 5\}$ $Y_3 = \{3, 5\}$

$Y_4 = \{4, 6\}$ $Y_5 = \{5\}$ $Y_6 = \{6\}$

$\alpha_{1,6} = \alpha_{1,4} = \alpha_{1,1} = \alpha_{1,7} = 14.28 \text{ c/s}$

$\alpha_{2,5} = \alpha_{2,2} = \alpha_{2,7} = 14.28 \text{ c/s}$

$\alpha_{3,5} = \alpha_{3,3} = \alpha_{3,7} = 14.28 \text{ c/s}$

$\alpha_{4,6} = \alpha_{4,4} = \alpha_{4,7} = 14.28 \text{ c/s}$

$\alpha_{5,5} = \alpha_{5,7} = 14.28 \text{ c/s}$

$\alpha_{6,6} = \alpha_{6,7} = 14.28 \text{ c/s}$

Step 10: $k = 1$, so go to step 11.

Step 11: $h = NSG = 1$, so the procedure terminates

The optimal solution therefore demands that no more than 14.28 units of capacity be assigned to each of the passages in link 7. Thus it is not possible to make complete usage of the full 100 units of

capacity in every link; excess capacity is being wasted in all links except link 7. The minimum average response time for this network is therefore

$$T = \frac{1}{7} \left[\sum_{i=1}^7 \frac{1}{\frac{100/7}{10} - 1} \right] = 2.333 \text{ sec/mess.}$$

In Chapter 5, a design technique for systematically discarding this excess unused capacity in higher level links to reduce cost will be presented.

3.4 Procedure for the Computation of Minimum Average Response Time in Networks with MX and MSC Concentrators

In this section, we develop an analysis procedure for obtaining the minimum average response time in an arbitrary n remote terminal network configuration which contains both MX and MSC concentrators, assuming all link capacities and external arrival statistics are known. Since finite capacity buffers are used in MSC's, the value obtained is an approximation. In accordance with Assumption 4 of Section 2.3 it is assumed that the buffer size is large enough so that the resulting error is small enough to be neglected. Hence, we will not continue to make explicit reference to this negligibly small difference. Also, the forward network only will be considered, but the analysis is also pertinent for the reverse network if the parameters are properly reinterpreted.

Define $M \triangleq \{ \text{all concentrator nodes in the remote network} \\ \text{which use MSC's} \}$

$\overline{M} \triangleq$ {all remote terminal nodes in the network not
in the set M }

Recall the definition of a channel given in Section 1.2 - the longest chain of contiguous passages serving a common remote terminal which does not have any intermediately located buffers. If we knew the capacity values for all the channels in a network having n remote terminals, the general expression (2.8) for average response time, T , could be grouped as follows:

$$T = \frac{1}{\sum_{j=1}^n \gamma_j} \left[\sum_{i \in \overline{M}} \left(\frac{1}{\frac{C_i}{\gamma_i} - 1} \right) + \sum_{i \in M} \left(\frac{1}{\frac{C_i}{\lambda_i} - 1} \right) \right] \quad (3.31)$$

where C_i is the capacity of the channel which originates at terminal i and $\lambda_i = \sum_{j \in I} \gamma_j$, the set I consisting of all remote terminals whose forward path to the central node contains the i th terminal. The first term of expression (3.31) represents the contribution from those channels which transmit only external message traffic originating at a single remote terminal. The second term denotes the delay component contributed by the channels which originate at remotely located MSC's and hence transmit traffic for all terminals which feed the MSC.

In order to determine the minimum value of T as given by expression (3.31) we must determine the optimal values of C_i for all channels i . Thus far in this work, however, we have only considered optimum channel capacity assignment procedures for networks containing MX

concentrators exclusively. In this section, we will exploit the fact that remotely located MSC's serve to partition the network into a number of independent subnetworks, each one of which is "rooted" at a node in the set M or at the central site. Each subnetwork contains only MX concentrators since every MSC looks like a sink to links feeding it but like a source to the link into which it feeds traffic directly. Thus the input message traffic to a particular subnetwork consists both of external terminal traffic and concentrated message traffic from any higher level subnetworks which may exist. Message switching centers located at higher levels of concentration thus submit their total concentrated traffic loads to the next lower level subnetworks by functioning as equivalent sources.

A straightforward analysis procedure can consequently be developed to solve the problem for the entire network by solving each of the independent subnetworks using the procedure of Section 3.3.2.

Algorithm

Redefine the set M to contain all MSC's in the network, including the central node. Order the nodes of the set M so that node m_i is not located at a lower level of concentration in the structure than is node m_{i+1} . (A node located k links away from the central node is at a higher level of concentration in the tree than a node located $k-1$ links away.) The set \bar{M} is still defined to contain all nodes in the network not in the set M .

In order to facilitate the presentation of the algorithm, we postulate the existence of a computational module TANDMX which is an implementation of the procedure of Section 3.3.2.1. This module accept inputs consisting of all link capacity values, arrival rates, the mean message length parameter for an arbitrary tree structured network containing only multiplexors. The outputs consist of optimal passage capacity and hence channel capacity assignments for the entire network.

Letting i be an index on the nodes in the set M , we define for each i :

$$X_i \triangleq \{ \text{all terminals in set } \bar{M} \text{ whose path to } m_i \text{ in the forward direction does not contain any intermediate MSC's} \}$$

$$Y_i \triangleq \{ \text{all terminals in set } M \text{ whose path to } m_i \text{ in the forward direction does not contain any intermediate MSC's} \}$$

Step 1: $i = 0$.

Step 2: $i = i+1$; determine X_i and Y_i from network structure.

Step 3: Considering only the subnetwork rooted at node m_i , the capacity of any channel emanating from a node in the set $X_i \cup Y_i$ has not yet been determined. All other channel capacity values in this subnetwork have been optimally assigned at a prior execution of steps 3 - 4. Form a temporary configuration from the subnetwork under consideration

in the following manner: Replace the entire subnetwork rooted at each node $y \in Y_i$ with a single node having a Poisson source of intensity $\lambda_y = \sum_{k \in J} \gamma_k$ where J is the set of all terminals whose forward path in the unmodified subnetwork includes node y . For the purposes of determining presently unknown channel capacities, this temporary network of nodes $X_i \cup Y_i$ is functionally equivalent to the actual network in question. This temporary subnetwork also contains only source nodes and multiplexor concentrator nodes in addition to node m_i .

Step 4: Execute the module TANDMX to optimally assign all passage and hence channel capacities in this temporary configuration. Let C_j^* denote this optimal assignment of channel capacity for the channel emanating from node $j \in (X_i \cup Y_i)$ in the temporary configuration. Let C_j denote the capacity of the channel originating at node j in the original network. Set $C_j = C_j^*$ for all $j \in (X_i \cup Y_i)$.

Step 5: Let $T(m_i)$ denote the contribution to the overall network response time due to the channels whose capacities were assigned at the most recent execution of step 4. Set

$$T(m_i) = \frac{1}{\sum_{j=1}^n \gamma_j} \left[\sum_{j \in X_i} \frac{1}{\frac{C_j}{\gamma_j} - 1} + \sum_{y \in Y_i} \frac{1}{\frac{C_y}{\lambda_y} - 1} \right]$$

Step 6: If $i = |M|$, set $T = \sum_{i=1}^{|M|} T(m_i)$ and terminate. Otherwise go to step 2. T denotes the minimum average response time obtainable for the network.

Individual subnetworks to which the tandem multiplexor model can be applied are thus analyzed in sequential fashion until the entire network performance has been resolved.

3.5 The Way Station Concentrator

Consider the multidropped line of Figure 3.11 which is controlled by a way station multiplexor.

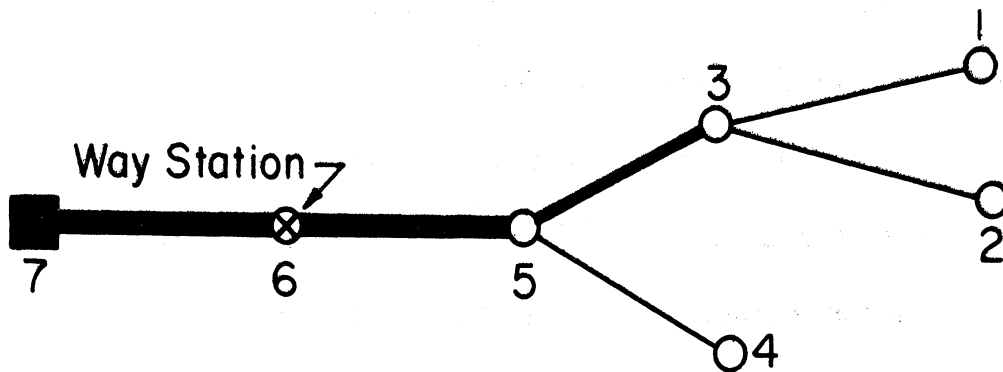


Figure 3.11 A Way Station Concentrator

Terminals 1 -6 effectively time share link 6. Although there are a variety of ways to implement the way station control procedure, it is common to allow only one of the multiplexed terminals to use the multidropped line at a time. Although there is no reason why the multiplexing could not be done on a short time slice basis (say a quantum equal to a character transmission time), party line control schemes with lengthy time slicing are quite common. For example,

if each of the terminals in Figure 3.11 is a teletypewriter (TTY), a time slice would correspond to the elapsed time from the establishment of a connection to the instant when the user at the TTY frees the line or "hangs up." This period is often referred to as the holding time. For a TTY, average holding times between 20 and 45 minutes are not uncommon. This type of operation is highly uncontrolled in the sense that when one terminal is active, no other terminal in the sharing group can obtain a connection until the user of the active terminal decides to free the line.

Efficient utilization of communications resources is difficult to obtain since only a very small fraction of any active terminal's holding time is spent for actual data transmission, and all other terminals are locked out for the holding time interval. Thus the server (communication line) is dedicated to a particular terminal, but performing useful work only a very small fraction of the time. For example, in the case of a TTY with an average holding time of 20 minutes, average input line lengths of 15 characters, and a mean arrival rate of 1 input line every 30 seconds, the average data transmission time for a line having a 200 char/sec TRIB would be 3 sec. The ratio of average transmission time to average holding time for this typical situation is $\frac{3}{1200}$.

In all discussions of average response time thus far, T has been defined as an average over all terminals where each terminal contains its own independent Poisson source generator and these source gen-

erators are considered to be simultaneously active. Clearly, this model does not apply in the case of the way station configuration since all but one of the terminals must be inactive when the line is occupied. Consequently, the average data transmission times for the active terminal are not influenced by what happens at other terminals, in contrast to the operating environment when multiplexors and MSC's are used.

Terminals in the sharing group do, however, interact in competing for the line and it is possible to formulate a queueing model which enables one to analyze the congestion resulting from this competition for access to the transmission facility. This queueing model for the way station is given in Figure 3.12. Assume that the length of time which each terminal holds the central link server is an exponentially distributed random variable with mean \bar{t}_H . Furthermore, let γ_i be the average rate of requests for use of the line from the i -th terminal. When the line is occupied, a new request cannot receive service until the end of the holding time interval for the currently active terminal. Queues of such requests for service are allowed to build up physically at the individual terminals, although for purposes of the delay analysis a common queue arises as a consequence of the control procedure which is used.

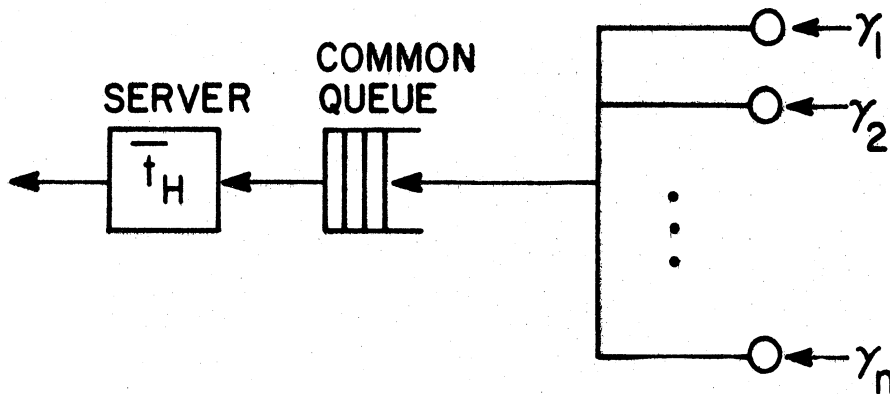


Figure 3.12 Way Station Queueing Model for n Remote Terminals

Let Q be the average time a request spends both waiting for access to the line and using it once acquired. Then an expression for Q is obtained by replacing $\frac{1}{\mu C}$ with \bar{t}_H in expression (2.3) where there are, in general, n users sharing the multidropped line.

$$Q = \frac{\bar{t}_H}{1 - \bar{t}_H \left(\sum_{i=1}^n \gamma_i \right)} = \frac{1}{\frac{1}{\bar{t}_H} - \sum_{i=1}^n \gamma_i} \quad (3.32)$$

It is interesting to note that any configuration in which the terminals of the sharing group submit a total rate of requests equal to $\sum_{i=1}^n \gamma_i$ in (3.32) will yield the same average delay as the configuration of Figure 3.12. This independence constitutes one of the most important limitations of the way station type of concentrator. For example if Figure 3.12 corresponds to a ten teletypewriter (TTY) configuration and $\gamma_i = 1$ arrival/30 minutes, then the average delay, Q , is the same as in a configuration having twenty TTY's with $\gamma = 1$ arrival every 60

minutes. It thus is not possible to reduce Q by distributing a constant load over additional input stations.

These undesirable features of the way station multiplexor can be eliminated, however, by greatly reducing the length of an average time slice in the sharing control procedure. For example, if the slicing is done on a bit or character basis, the way station becomes essentially a time division multiplexor in which the entry/termination (drop off) points are distributed along the length of the line and the one multiplexed line connects all the remote terminals. Kaplan [2] suggests that this type of distributed time division multiplexor may offer substantial advantages over existing equipment in future data transmission systems.

3.6 Telephone Type of Concentrator

In this type of concentrator, m input lines of the same speed are selectively connected to a smaller number, say n , of lines of the same speed which constitute the trunk or shared link. Multiplexing is accomplished on a call by call basis with no speed change between the input and output lines on the trunk. The way station concentrator discussed in Section 3.5 is a special case type of telephone concentrator in which there is but one line in the trunk.

For reasons similar to those discussed in Section 3.5, it is relatively difficult to obtain efficient utilization of the network's communications resources with telephone concentrators. Again,

a terminal ties up a line on the trunk for the duration of the entire holding time interval during which the line is performing useful work only a very small fraction of the time. This type of concentration scheme is similarly uncontrolled in the sense that no terminal in the sharing group can obtain the use of a busy line on the trunk until the user of the corresponding active terminal decides to free the line.

Since at most n of the m terminals in the sharing group may be active simultaneously, it is not meaningful to consider average data transmission times when the average is taken over all the terminals in the network. Thus, the Poisson source generator used throughout this work to model remote terminal behavior when all terminals in the network are considered to be simultaneously active clearly is not relevant in the case of the telephone concentrator. Since all lines in and out of the concentrator are of the same speed, and concentration is accomplished on a call by call basis, the data transmission times experienced by any one of the active terminals are not influenced by what happens at any other terminal. This is in contrast to the operating environment when MX's and MSC's are used; in that situation all terminals sharing a facility interact in the sense that channel capacities and hence transmission times are dependent on the arrival rates for each terminal in the sharing group.

All of the m terminals in the sharing group do, however, interact in competing for access to the n lines on the trunk. It is thus

possible to formulate a multiserver queueing model which enables one to analyze the congestion resulting from this competition for access to the transmission facilities. The multiserver queue with no loss model [70] of Figure 3.13 is appropriate for this special type of concentrator. We again assume that holding time for each terminal is an exponentially distributed random variables with mean \bar{t}_H , and let γ_i denote the Poisson average rate of requests from the i -th terminal for use of a line on the trunk.

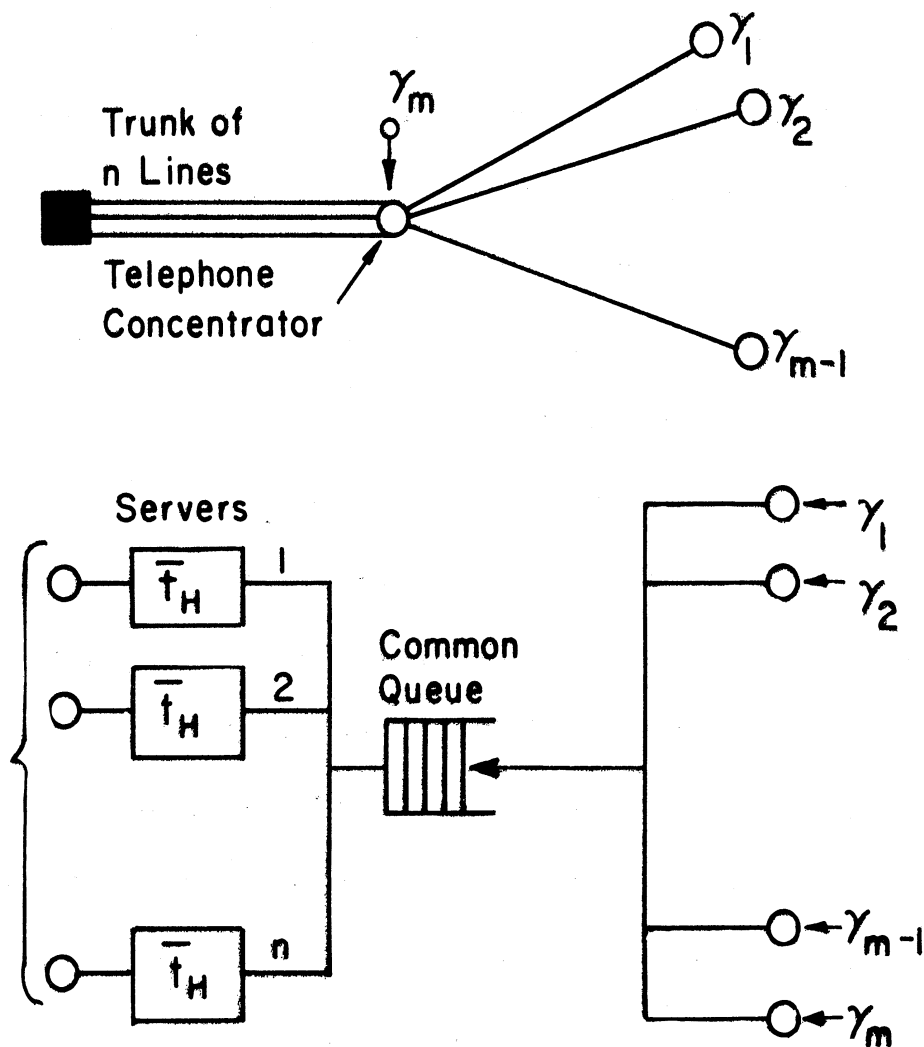


Figure 3.13 Queueing Model for Telephone Concentrator

In this model, each of the n lines in the trunk acts as one of n identical parallel servers for requests which arrive from the m remote terminals. When all trunks are busy, the control procedure monitors a queue of requests for the first line on the trunk which becomes available. The requests physically queue up at the respective remote terminals; the common queue arises as a consequence of the control procedure which is used. When all lines on the trunk are busy and one becomes free, a terminal is selected from the list of those waiting on a first come, first served (FCFS) basis. If several lines on the trunk are available to an arriving request, a free one is usually selected at random. This procedure most nearly distributes the traffic load evenly over all the trunk lines.

The average traffic intensity per trunk line is given by

$\frac{\overline{t}_H}{n} \sum_{i=1}^m \gamma_i$ which is also equal to the system utilization ρ .¹ It can be shown that the probability P_N of there being exactly N requests in the system (either undergoing service or awaiting access to a line in the trunk) at a given time is given by²

¹ We demand that $\rho < 1$.

² See, for example, Reference [70].

$$P_N = \begin{cases} \frac{(n\rho)^N P_0}{N!} & \text{for } N < n \\ \frac{(n\rho)^N P_0}{n! n^{N-n}} & \text{for } N \geq n \end{cases} \quad (3.33)$$

where

$$n\rho = \bar{t}_H \sum_{i=1}^n \gamma_i \quad (3.34)$$

and

$$P_0 = \frac{1}{\sum_{N=0}^{n-1} \frac{(n\rho)^N}{N!} + \frac{(n\rho)^n}{(1-\rho)n!}}$$

The above expressions are then used to obtain Q , the average time a request spends both waiting for access to a trunk line and using it once acquired, by the following relationship:

$$Q = Q_W + \bar{t}_H \quad (3.35)$$

where Q_W is the mean time elapsed before a free trunk line is obtained. Q_W is obtained from the relation¹ $Q_W = L_W / \sum_{i=1}^m \gamma_i$ where L_W is the mean length of the waiting line excluding those requests undergoing servicing and $\sum_{i=1}^m \gamma_i$ denotes the total rate of requests for trunk line accesses from all remote terminals. From

¹ See, for example, Reference [62], page 103.

the standard expectation formula, we have

$$L_W = \sum_{k=n}^{\infty} (k-n) P_k \quad (3.36)$$

which can be written as

$$L_W = \frac{\rho}{1-\rho} \sum_{k=n}^{\infty} P_k \quad (3.37)$$

so that

$$Q_W = \frac{\rho}{1-\rho} \cdot \frac{1}{\sum_{i=1}^m \gamma_i} \cdot \sum_{k=n}^{\infty} P_k \quad (3.38)$$

which is equivalent to

$$Q_W = \frac{\bar{t}_H}{n(1-\rho)} \cdot \sum_{k=n}^{\infty} P_k \quad (3.39)$$

Finally we have for Q , the mean waiting and servicing delay

$$Q = \bar{t}_H \left[1 + \frac{1}{n(1-\rho)} \cdot \sum_{k=n}^{\infty} P_k \right] \quad (3.40)$$

Curves for Q , normalized to the mean holdingtime, \bar{t}_H , as a function of the number of trunk lines and the system utilization are presented in Figure 3.14.

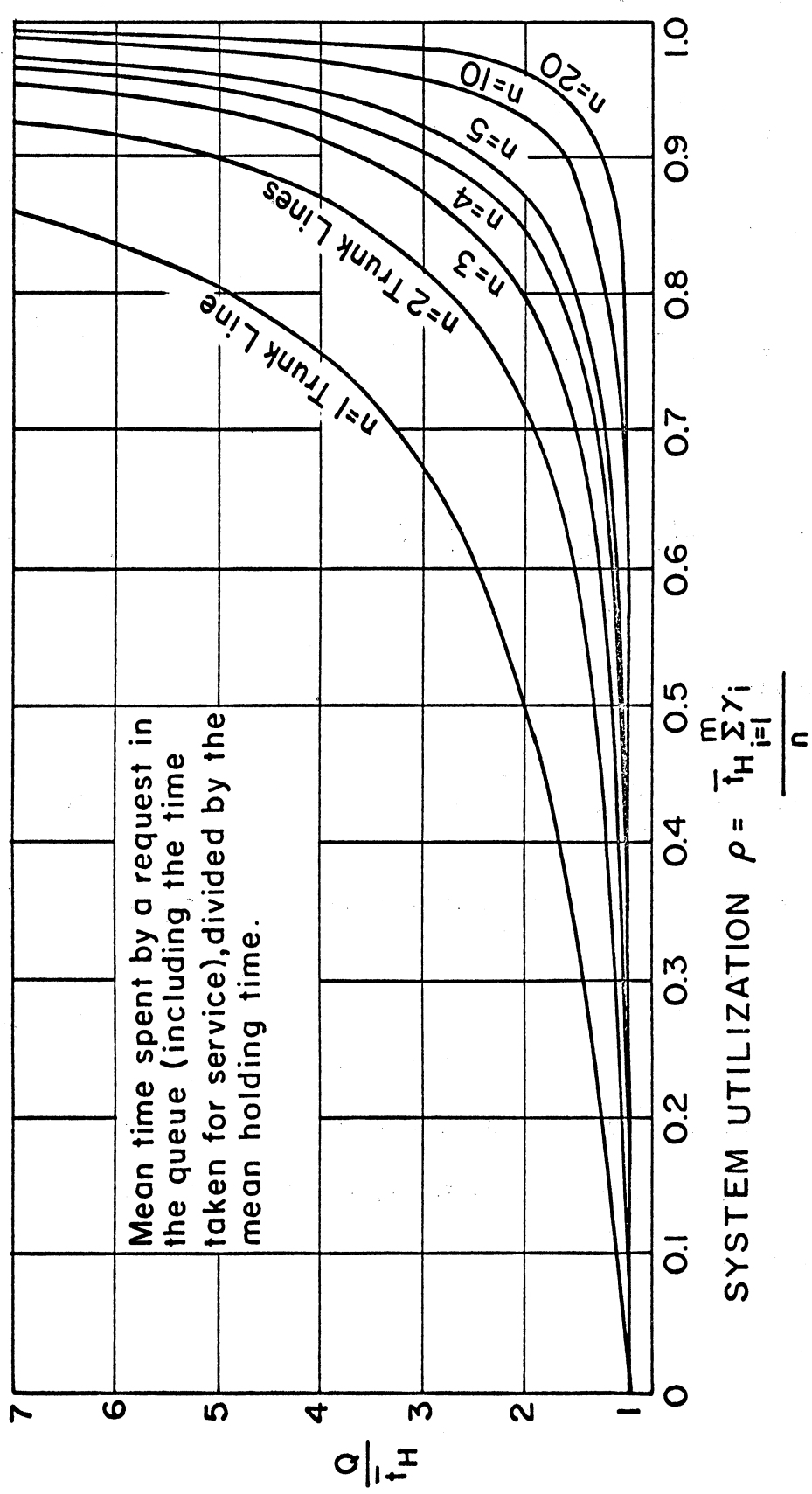


Figure 3.14 Queueing Times for Accessing and Using a Trunk Line in the Telephone Concentrator

3.7 Comparative Performance Considerations

In the last two sections, we have considered the way station and telephone types of concentrators noting that, at any time, only a fraction of the terminals in the sharing group may be simultaneously active. Furthermore, for any of the active terminals, it has been shown that the data transmission times are not influenced by the behavior of any other terminal. Queueing for access to the communications facilities however does arise, and was considered in the model of Figures 3.12 and 3.13. Direct comparison of this type of congestion for the way and telephone concentrators can be made using expression (3.40) and the curves of Figure 3.14, since the way station is a special case of the telephone type of multiplexor in which there is but one server ($n = 1$).

A number of the relatively undesirable features of these two types of concentration has been cited throughout Section 3.5 and 3.6. In recapitulation, we noted the relatively uncontrolled environment which exists in way and telephone concentration schemes in contrast to networks with MX's and MSC's where no terminal in a sharing group is every locked out by another user from accessing a communication line. It was also noted that it is impossible to reduce the average time to acquire the line in a way station system by redistributing a constant total demand over additional input terminals. In the telephone type of concentrator we noted that input and output line speeds must be identical. However, considerable motivation for the

capability of using different line speeds in a network was cited in Chapter 1. For these reasons, the way station and telephone types of concentrators will not be considered further. The balance of the effort in this section is devoted to comparative studies of the MX and MSC of concentration.

3.7.1 Message Switching Centers vs. Multiplexors in the Single Concentrator Case

To compare the cost effectiveness of MSC's and MX's, in addition to models for obtaining average response times of the devices we also need to establish meaningful cost data. Even though we are concerned only with comparing the average response times of the MX and MSC in the single-concentrator case, usage of the MSC can buy additional capability which must be considered in arriving at comparable cost functions. In particular, the features of the MSC to perform code translation, message editing, error correction, and control functions as well as to store messages in transit, result in a reduction of the data channel interference with the central computer. Hence the existence of a MSC in the remote network can increase the throughput of the central computational facility. To put it another way, dollars invested in a MSC affect the response time of the telecommunication network and also can have the effect of replacing the central CPU by a hypothetical machine which has higher throughput capability as a consequence of reduced data channel interference. The actual costs of a MSC can therefore be decreased by whatever dollar value can be

given to the increased throughput capability. Models for quantitatively determining this channel interference in the IBM/360 have been reported in references [50] and [51].

In this research, however, we have elected not to reduce MSC costs since no empirical data on the values of reduced data channel interference is readily available. Also, an arbitrarily selected nonzero value would be of dubious merit in our efforts which are intended to be as general purpose as possible. Since whatever actual value of increased throughput, if any, results from using MSC's is likely to depend significantly on the application, a better estimate of this value can only be made after a careful consideration of all factors relating to the design problem at hand.

All of these factors must be considered in arriving at meaningful cost functions which are central to any comparative studies which one might want to conduct. We now turn to the other important factor which must be considered in comparing MX's and MSC's, namely the average response time characteristics of the devices.

As will be noted, even though multiplexors are significantly less expensive concentrating devices than MSC's, their average response time characteristics are only significantly poorer than MSC's in relatively high duty cycle environments. One might be tempted to conclude that MSC's always yield lower average response times than MX's. Such intuitive notions were shown to be incorrect in some parametric studies which follow. Hence any anticipated performance

superiority of the MSC vanishes in certain low duty cycle applications.

Due to the iterative nature of the optimum passage capacity assignment procedure for MX's, it is difficult to generalize, to any great extent, on the relative performances of the two modes of concentration. There is one specific generalization which can be made to establish an inequality relating T using the MSC (denoted T_{MSC}) and T using the MX (denoted \hat{T}_{MX}). This generalization is stated in Theorem 3.2.

Theorem 3.2

If $\rho_H \leq \rho_n$, then $T_{MSC} \leq \hat{T}_{MX}$ where we define
(Figure 3.15)

$$\rho_H \triangleq \frac{\sum_{i=1}^n \frac{\gamma_i}{\mu}}{\alpha_n}, \text{ the utilization of the shared link}$$

if a MSC is used, and

$$\rho_n \triangleq \frac{\frac{\gamma_n}{\mu}}{\hat{\alpha}_{nn}} \text{ the utilization of the single-passage}$$

channel originating at the concentrator site and

$\hat{\alpha}_{nn}$ is the optimal passage capacity assignment for the single-passage channel originating at the concentrator site if a multiplexor is used.

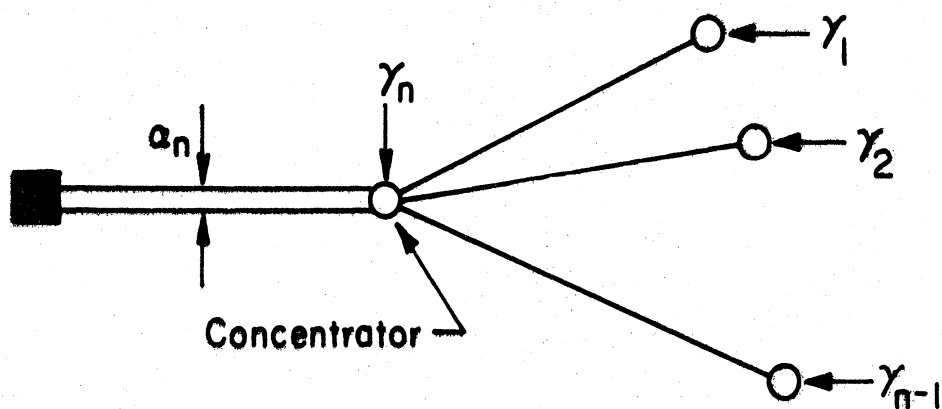


Figure 3.15 A Generalized Schematic for Single Stage Concentration

Proof: We recall expression (3.13) and (3.1) for \hat{T}_{MX} and T_{MSC} respectively, noting of course that (3.1) applies to an $n+1$ channel network but that the network Figure 3.15 only has n channels.

$$\hat{T}_{MX} = \frac{1}{\gamma} \left[\sum_{i \in B} \left(\frac{1}{\frac{\alpha_i}{\gamma_i} - 1} \right) + \sum_{j \in \bar{B}} \left(\frac{1}{\frac{\hat{\alpha}_{jn}}{\gamma_j} - 1} \right) \right] \quad (3.41)$$

and

$$T_{MSC} = \frac{1}{\gamma} \left[\sum_{i=1}^{n-1} \left(\frac{1}{\frac{\alpha_i}{\gamma_i} - 1} \right) + \frac{1}{\frac{\alpha_n}{\sum_{j=1}^n \frac{\gamma_j}{\mu}} - 1} \right] \quad (3.42)$$

If a multiplexor is used, then B denotes the set of all remote terminals whose passage capacity values on the shared link are equal to the capacity of their respective input links. \bar{B} is the complementary set, so that $B \cup \bar{B}$ is the set of all remote terminals in the network. The $\hat{\alpha}_{jn}$ are the optimal passage capacity assignments for terminals in the set \bar{B} . Form the difference expression,

$$\hat{T}_{MX} - T_{MSC} = \frac{1}{\gamma} \left[\sum_{j \in \bar{R}} \left(\frac{1}{\frac{\hat{\alpha}_{jn}}{\gamma_j} - 1} - \frac{1}{\frac{\alpha_j}{\gamma_j} - 1} \right) + \left[\frac{1}{\frac{\hat{\alpha}_{nn}}{\gamma_n} - 1} - \frac{1}{\frac{\alpha_n}{\sum_{i=1}^n \frac{\gamma_i}{\mu}} - 1} \right] \right]$$

where \bar{R} is the set of all terminals in \bar{B} except the n -th terminal at the concentration site.

The first difference component, the sum over $j \in \bar{R}$ is guaranteed strictly positive since the optimal passage capacity assignment procedure includes in \bar{R} only terminals for which $\hat{\alpha}_{jn} < \alpha_j$. The hypothesis about ρ_H and ρ_n guarantees the positiveness of the second difference component and the proof is complete.

3.7.1.1 Parametric Variations, Some Case Studies

In this section, an analysis is made for a particular ten terminal single stage concentrator (Figure 3.15 with $n=10$). In an attempt to draw some meaningful general conclusions, a wide range of input link capacity values was covered in analyzing the performance of this configuration using both a MSC and MX at the concentrator site.

Figures 3.16 - 3.23 are plots of T_{MSC} and \hat{T}_{MX} as a function of the utilization of the shared link, for various combinations of input link capacity values. Expressions (3.41) and (3.42) were used to compute \hat{T}_{MX} and T_{MSC} respectively. Curves were obtained for models in which all input links have the same capacity as well as for models in which no two input links have the same transmission capacity.

3.7.1.2 Conclusions of Comparative Case Studies

Probably the most significant observation which can be made is that the performance differential of the MSC and MX types of concentrators in all instances appears to depend critically on the ratio $\rho_H/\bar{\rho}_i$ where

$$\rho_H \triangleq \frac{\sum_{i=1}^n \frac{\gamma_i}{\mu}}{\alpha_n}, \text{ the utilization of the shared link}$$

$$\bar{\rho}_i \triangleq \frac{\sum_{i=1}^{n-1} \frac{\gamma_i}{\mu \alpha_i}}{n-1}, \text{ the average utilization of a low speed input link.}$$

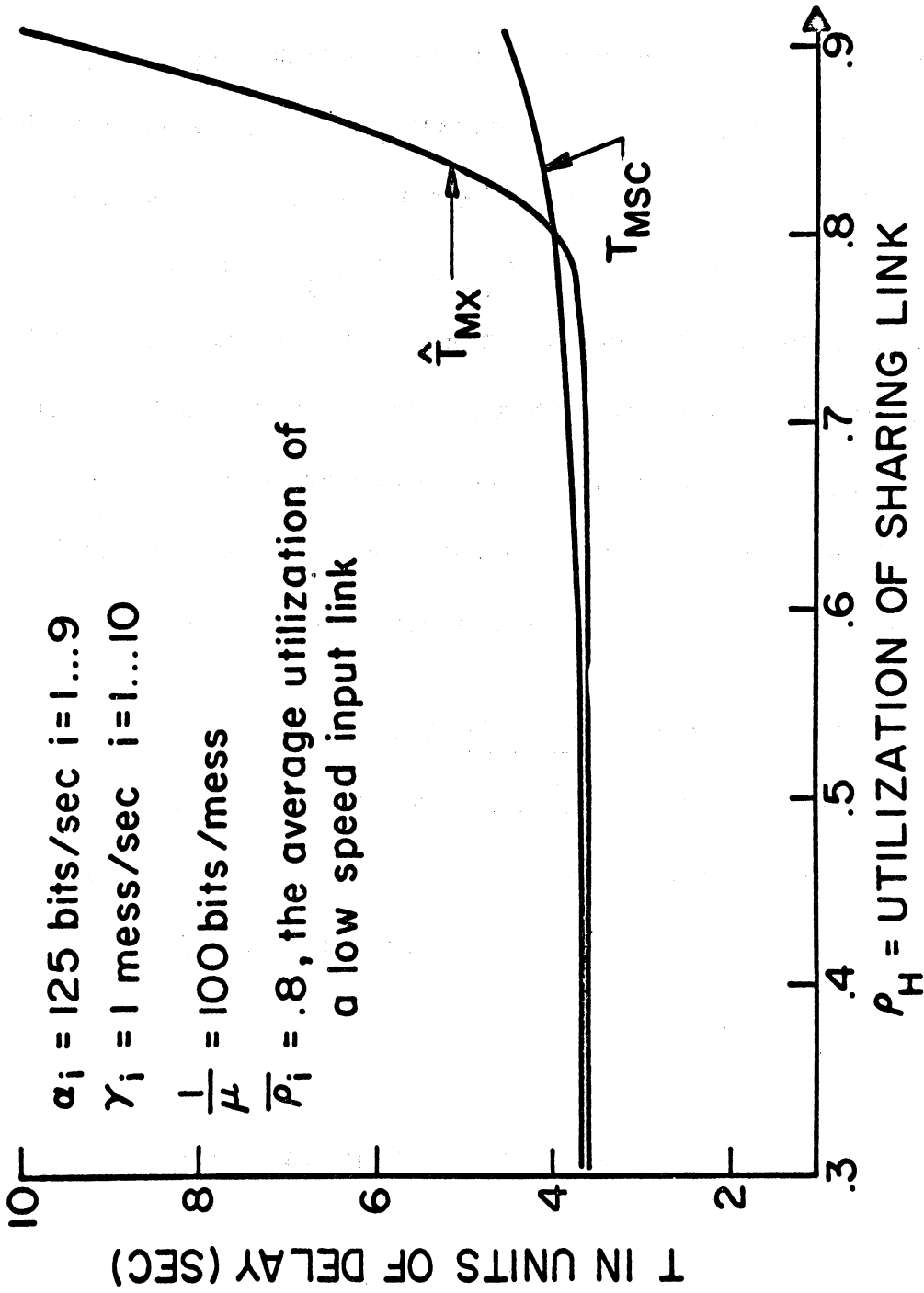


Figure 3.16 Comparative Study #1 of MX and MSC Concentrators

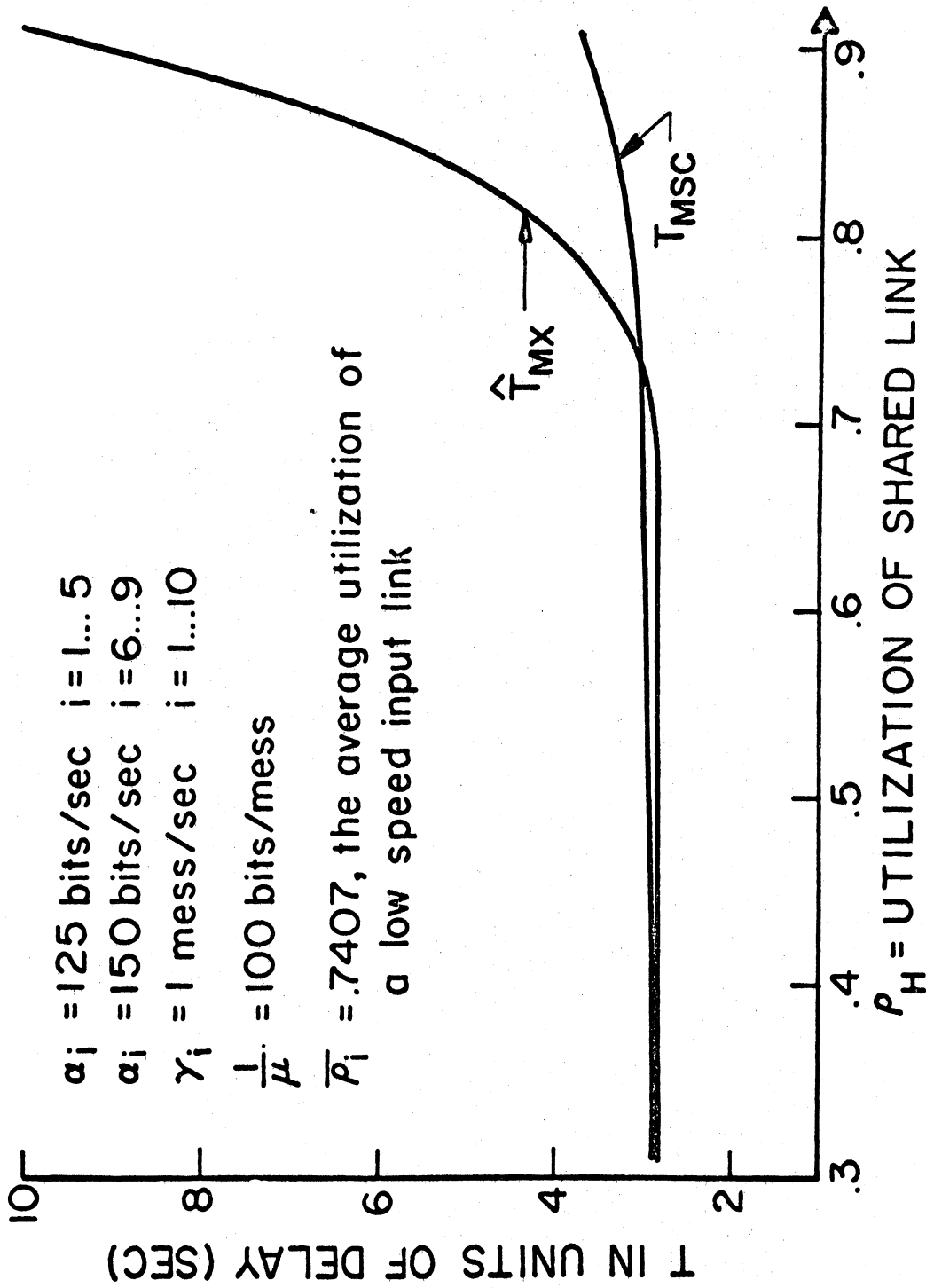


Figure 3.17 Comparative Study #2 of MX and MSC Concentrators

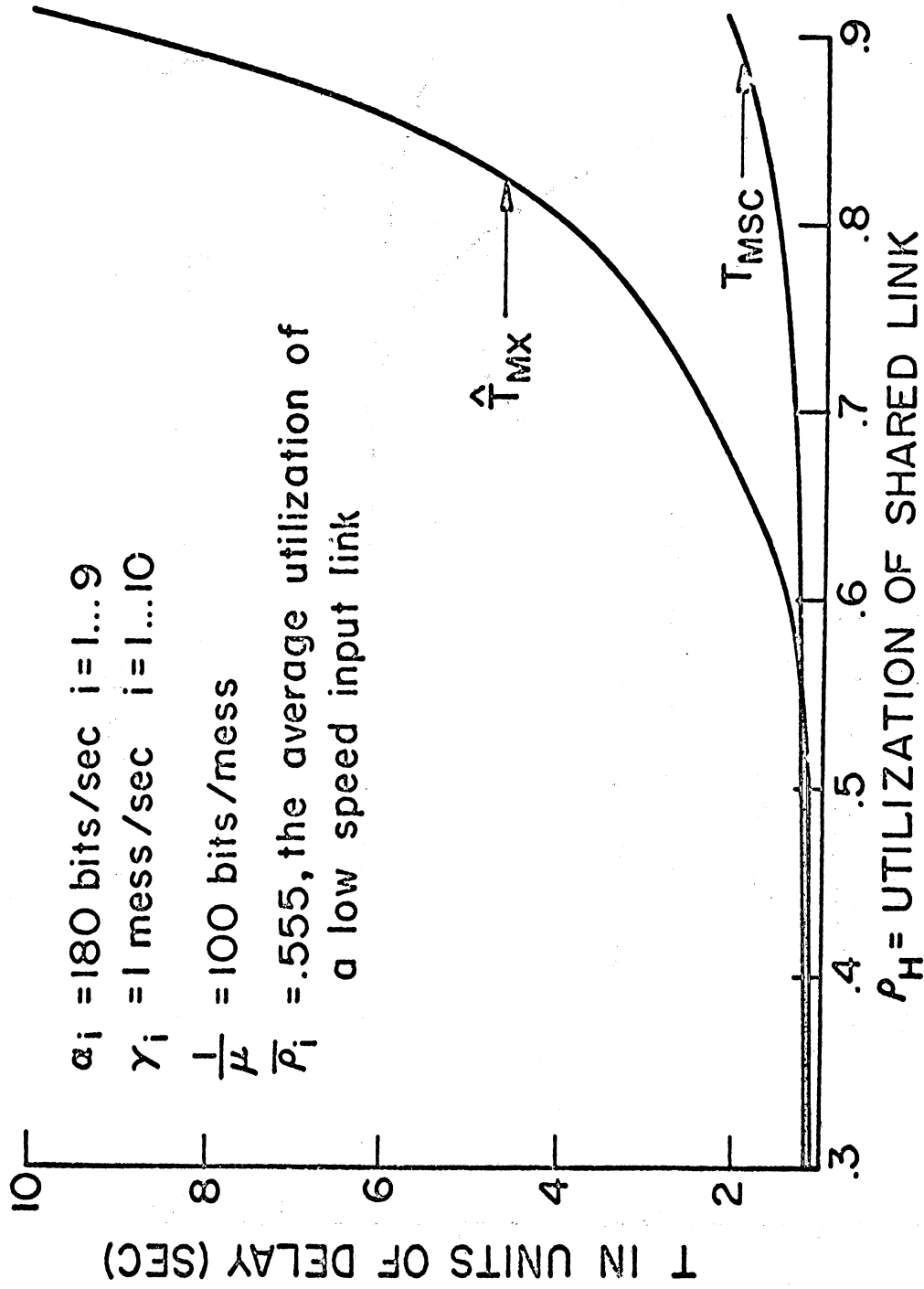


Figure 3.18 Comparative Study #3 of MX and MSC Concentrators

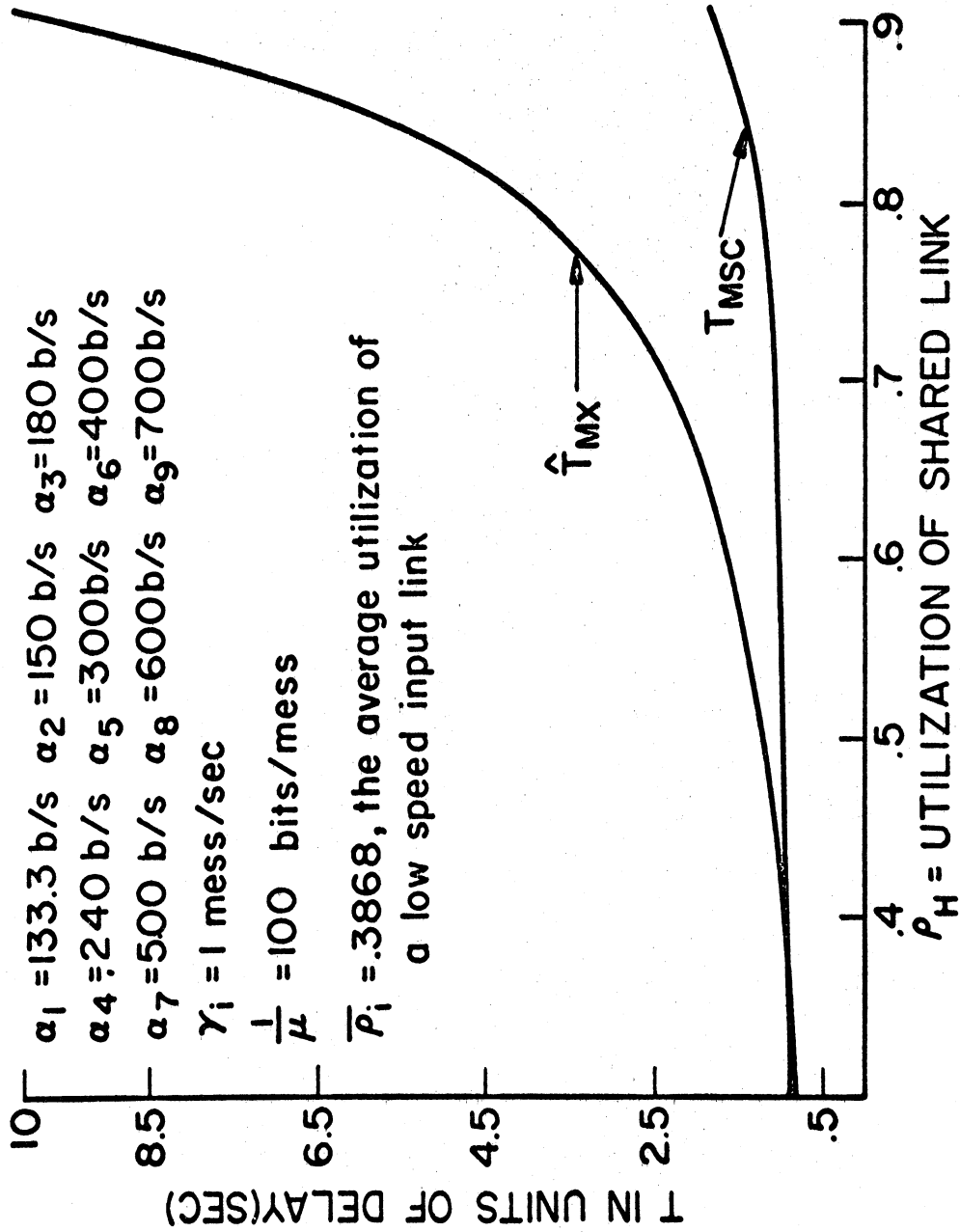


Figure 3.19 Comparative Study #4 of MX and MSC Concentrators

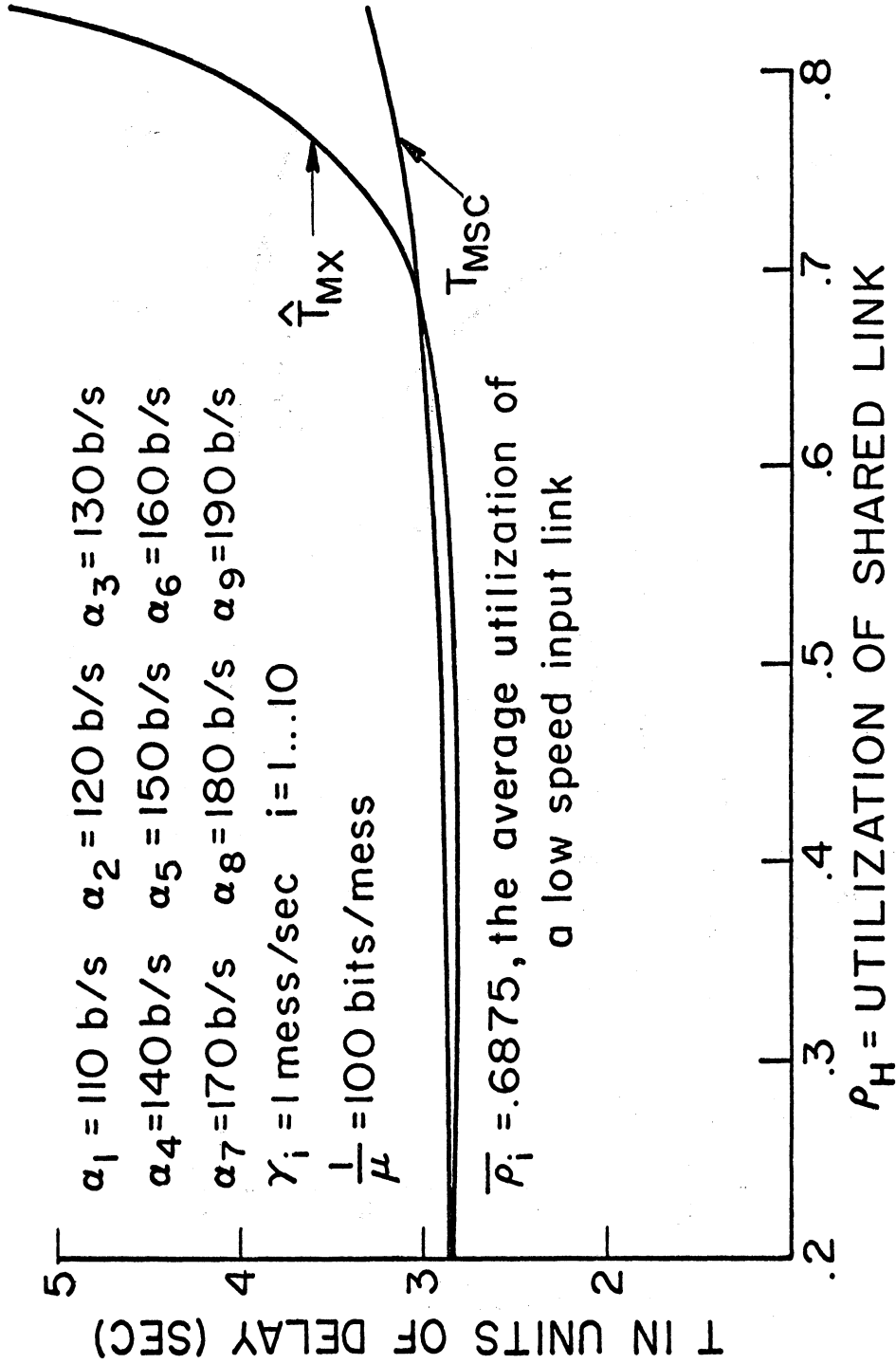


Figure 3.20 Comparative Study #5 of MX and MSC Concentrators

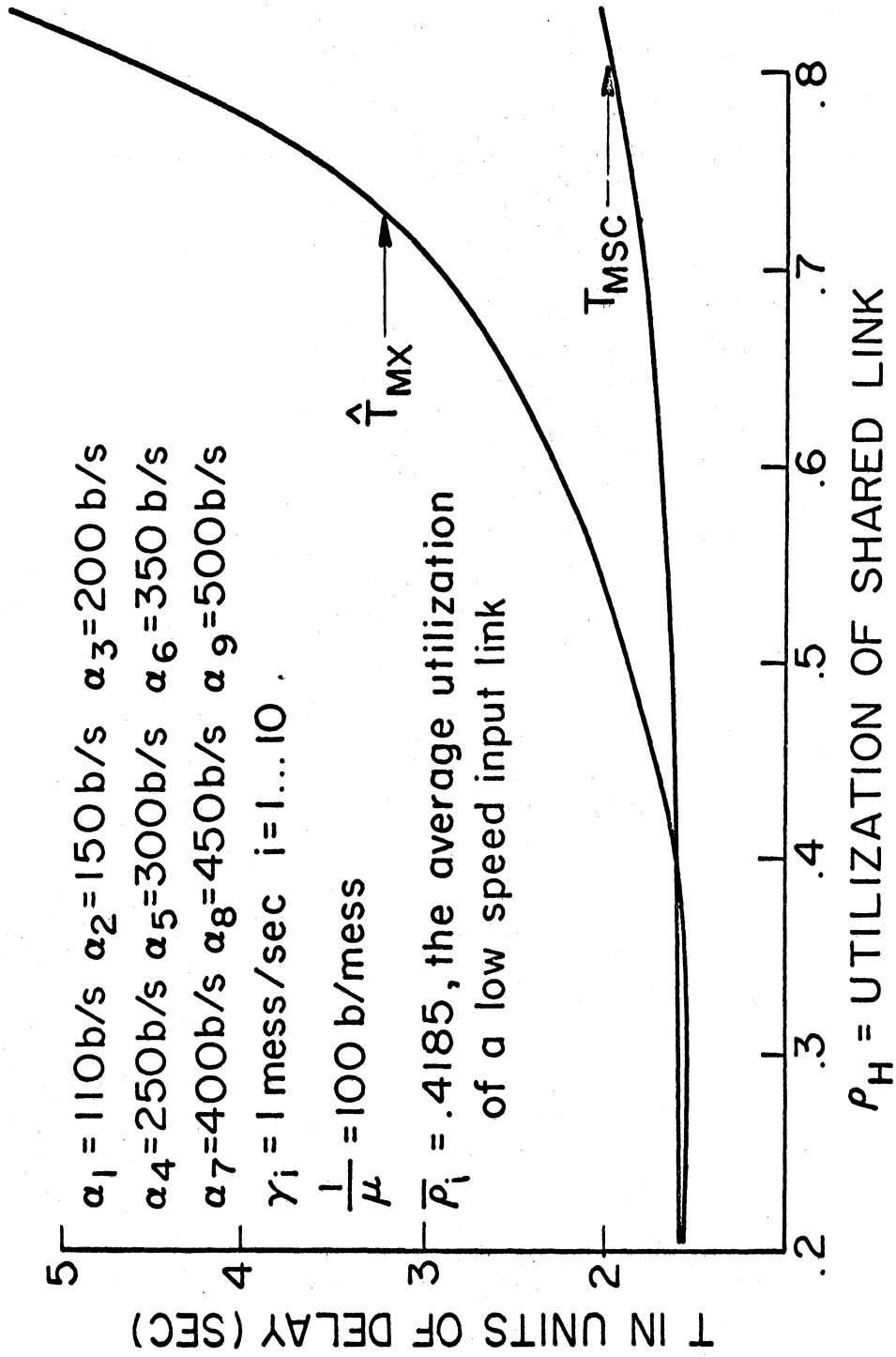


Figure 3.21 Comparative Study #6 of MX and MSC Concentrators

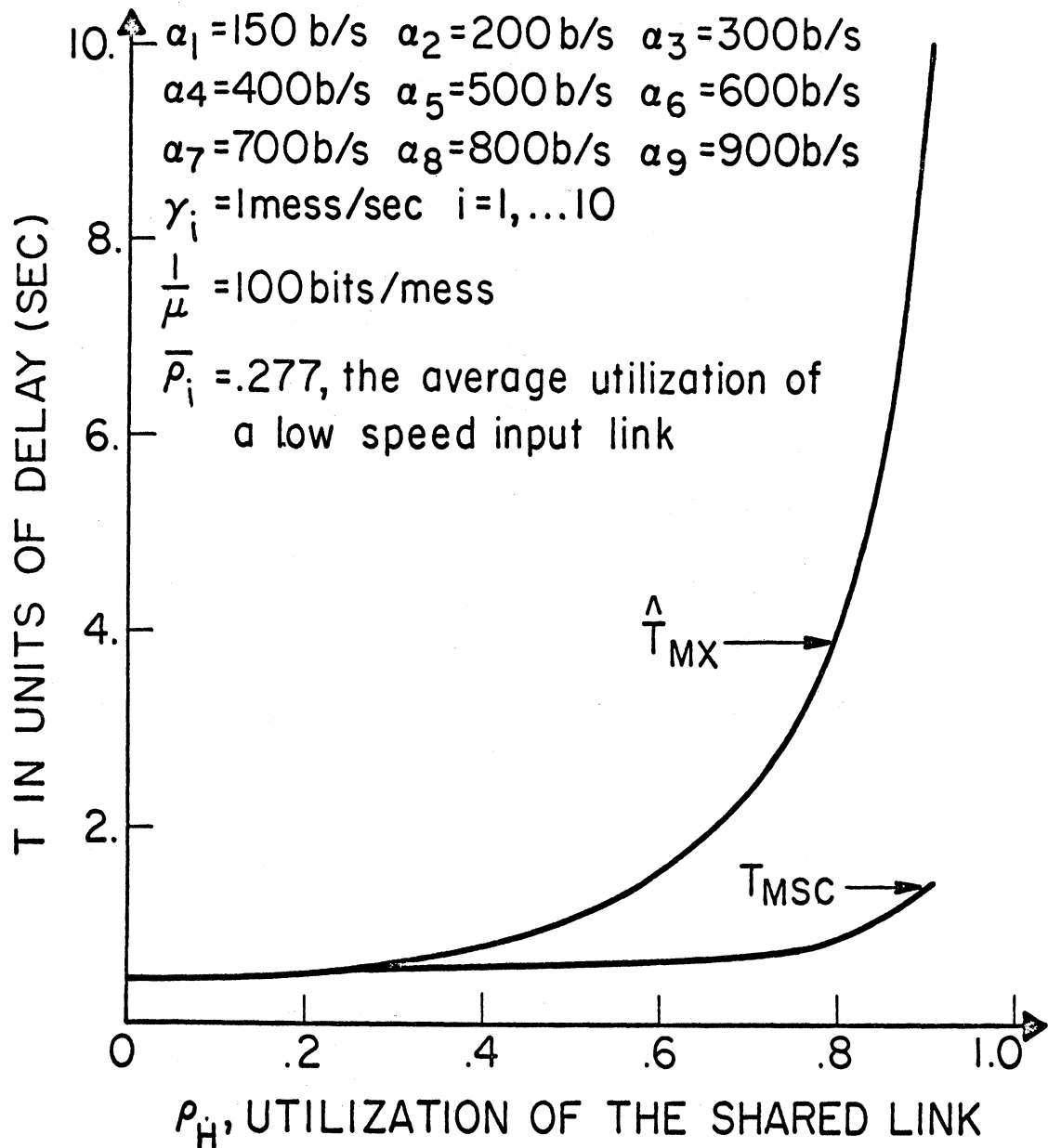


Figure 3.22 Comparative Study #7 of MX and MSC Concentrators

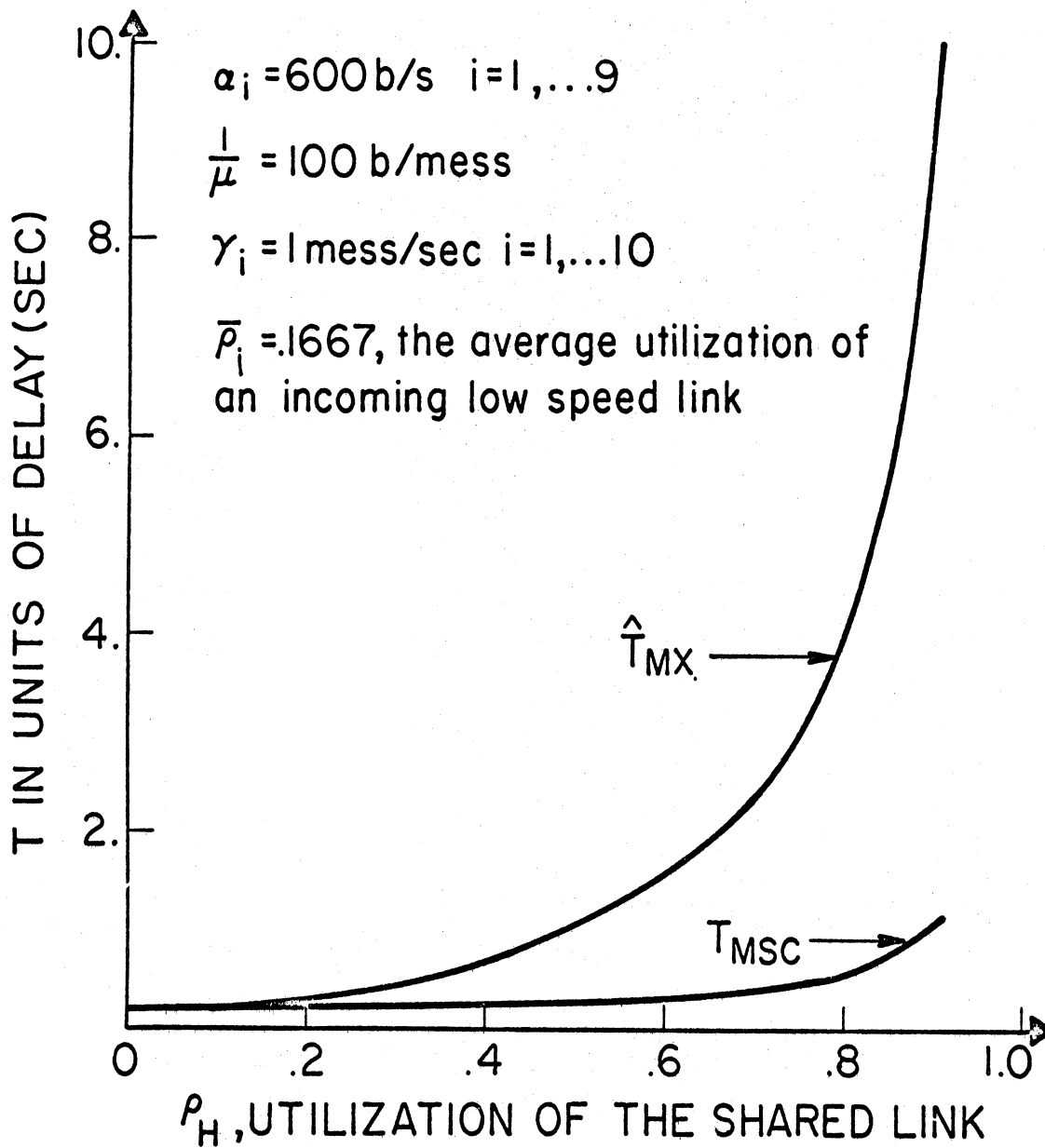


Figure 3.23 Comparative Study #8 of MX and MSC Concentrators

Whenever $\rho_H > \bar{\rho}_i$, $T_{MSC} < \hat{T}_{MX}$ and whenever $\rho_H < \bar{\rho}_i$, $T_{MSC} > \hat{T}_{MX}$.

Thus, although we have not established a proof, we may conclude from observation that the performance of the two concentrators is approximately the same whenever $\rho_H \cong \bar{\rho}_i$. For the somewhat pathological situations in which the shared link has a lower utilization ρ_H than the average utilization of a low speed input link, the MSC does not have any performance superiority over the MX concentrator.

The magnitude of the performance differential clearly depends not only on the ratio $\rho_H/\bar{\rho}_i$ but also on ρ_H itself. It can be observed that $\hat{T}_{MX} - T_{MSC}$ always increases as ρ_H is increased above $\bar{\rho}_i$, but the rate of change in the increase depends on the particular values of $\rho_H/\bar{\rho}_i$ and ρ_H .

Concerning the asymptotic values of \hat{T}_{MX} and T_{MSC} as $\rho_H \rightarrow 0$, they must be equal, since the equality of

$$\lim_{\alpha_n \rightarrow \infty} \hat{T}_{MX} = \lim_{\alpha_n \rightarrow \infty} T_{MSC} = \frac{1}{\gamma} \left[\sum_{i=1}^{n-1} \frac{1}{\frac{\alpha_i}{\gamma_i} - 1} \right]$$

can easily be established from expression (3.41) and (3.42).

Chapter IV
COST FUNCTIONS AND EXISTING
NETWORK DESIGN PROCEDURES

4.1 Introduction

In the remaining portions of this work, the efforts of the previous chapters are integrated into a procedure to be used for the design of centralized telecommunication networks. The independent variables to be treated are network topological structure, link capacity values, and types of data concentrators. The procedure will be geared to obtain solution configurations which must have average response times not exceeding a certain maximum acceptable value.

Although the procedures presented in this work will not always obtain theoretically optimal solutions, the design techniques represent significant improvements over existing state-of-the-art procedures; they incorporate the key tradeoffs which exist between all of the critical design variables.

Before plunging headlong into a treatment of the most general problem formulation, it is essential that the important relationships between cost and performance be explicitly characterized for each of the design variables. It is likewise essential that the characteristics and limitations of existing network design procedures be well established before any efforts are made to consider additional variables in the design problem. For these reasons, the entirety of Chapter 4

will be devoted to a detailed presentation of existing design procedures and to the establishment of realistic cost functions.

After these tasks are completed in Chapter 4, we will turn in Chapter 5 to treat the important concepts of feasibility and improvability for given network configurations. These concepts are essential to an understanding of the design tradeoffs and to the efficient allocation of resources in a teleprocessing network. The optimal passage capacity assignment procedure developed in Chapter 3 is shown to have important connotations insofar as feasibility and improvability are concerned. Use of this procedure will enable one to determine the exact amount of excess or unused capacity in the links of a network. The balance of Chapter 5 is devoted to interpreting these results in the light of existing design procedures; the consistent nonoptimality and resulting improvability of solutions generated by existing procedures is readily established. Finally, the results of all prior chapters are incorporated into a procedure for solving the most general design problem in Chapter 6.

As the initial step toward accomplishing all these objectives the cost functions which will be used throughout the balance of the work are established.

4.2 Cost Functions for Link Capacity and Data Concentrators

We now consider the two cost functions which are of utmost concern to the network designer - the cost of the various discrete transmission capacities usually available and the costs for the multiplexor

and MSC types of data concentrators.

4.2.1 Link Capacity Cost Function

The cost function of Figure 4.1 for link capacity in a unit length link has been established using data which is reasonably consistent with recent common carrier tariffs but does not reflect the extraordinary and seemingly unnecessary degree of complexity inherent in the existing tariff statutes of the common carriers¹.

For the convenience of mathematical modelling, it has been assumed that the cost of a link is linearly proportional to its length. The cost function, plotted as approximately a straight line on a logarithmically scaled grid reflects the nonlinearities or economies of scale which exist in the cost of capacity. That is, the cost per unit of capacity is lower in links which have the higher capacity values. The cost function of Figure 4.1 is plotted for a zero redundancy ($q=0$) link (See Section 1.3.1); for different values of q one obtains a new cost function simply by scaling the abscissa parameter, capacity, by the multiplicative factor $1 - q$.

4.2.2 Data Concentrator Cost Functions

In a realistic formulation of a data concentrator cost function, the device cost should depend, at least, on the number of line interfaces (or equivalently, the number of links incident to the concentrator) and on the aggregate speeds of the devices being concentrated. Since

¹ As an illustration of this point, see Reference [75].

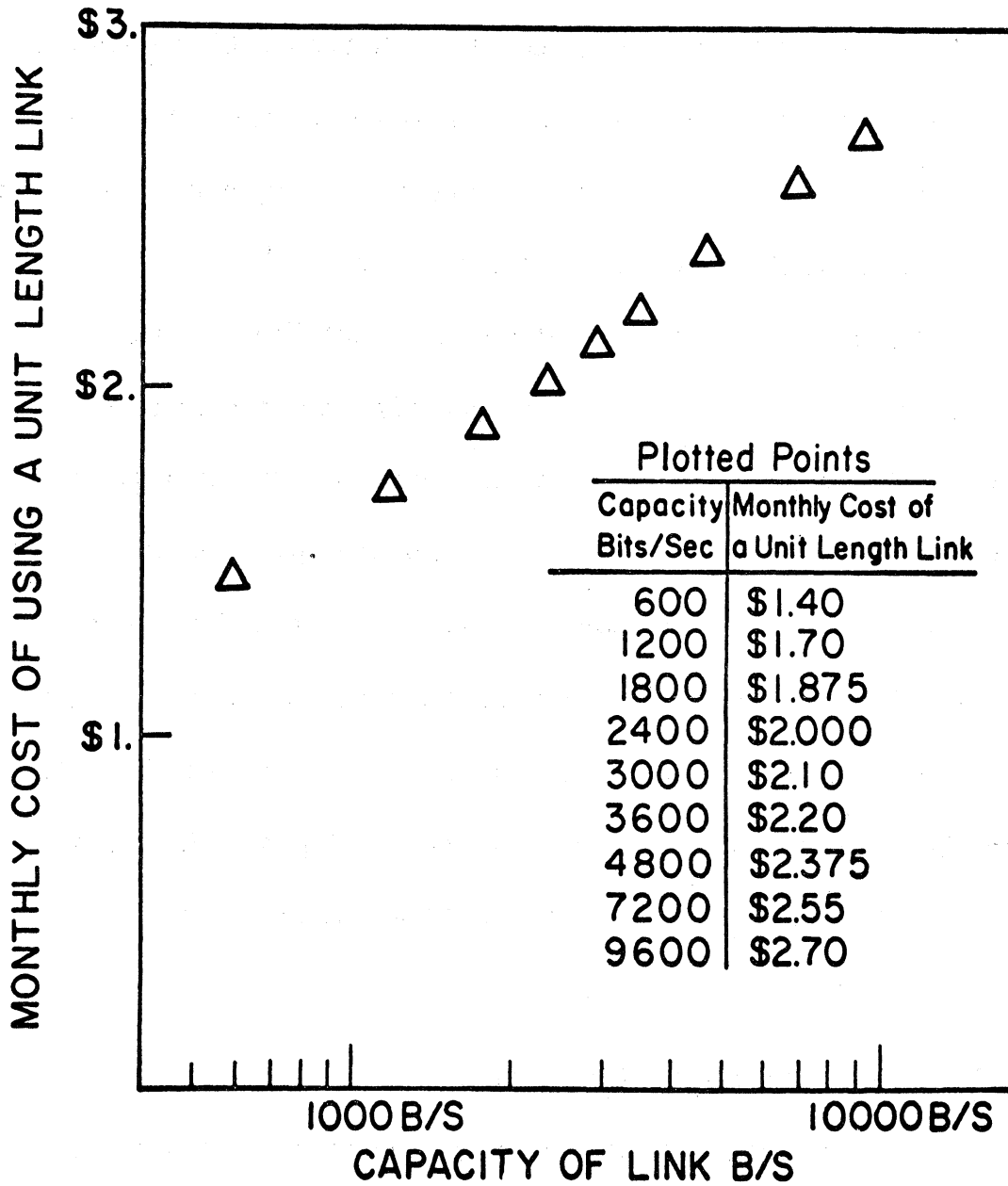


Figure 4.1 Cost vs. Capacity Function in a Unit Length Link

the types of devices may vary from one concentrator node to another, it is most meaningful to normalize this aggregate speed parameter with respect to some commonly used device. The teletypewriter has been used in this work.

Thus we will characterize the aggregate speeds of incoming devices in terms of the number of teletypes which would require the same amount of device-handling capability. Hence the nomenclature "equivalent number of teletypes" is used to describe the total device-handling speed capability which a given concentrator must have.

It is most realistic to reflect the cost dependence of concentrators on these two parameters in the following manner:

- Depict the dependence on the number of line interfaces in the manner shown in Figures 4.2 and 4.4, assuming a single concentrator is sufficient to handle the number of equivalent teletypes which must be concentrated.
- Whenever the speed limitations of a single concentrator are insufficient to handle the aggregate speeds of incoming devices at a particular concentrator node, compensate for this insufficiency by adding the minimum number of additional concentrator equipments which provide adequate device handling capability.

The maximum device-handling capability of a single MX or MSC concentrator has been chosen as 100 equivalent teletypewriters for reasons which are explained in Section 4.2.2.2. For mathematical

modeling convenience, and in the absence of a more realistic alternative, the same "saturation" level has been selected for the MX and MSC concentrators. Since the line scanning requirements are by far the most dominant of the limiting effects in either case, this assumption does not seem unrealistic.

4.2.2.1 Multiplexor Types of Concentrators

Figure 4.2 depicts the cost function which has been postulated for a single multiplexor concentrator. The function was compiled on the basis of data gathered from various manufacturers¹. The cost function reflects standard amortization rates, and state of the art concentrator technology as well as the realistic fixed cost component for the first two multiplexed channels which are formed. The first additional 1-8 multiplexed channels may be formed at an incremental concentrator cost of \$25 per channel, whereas any additional multiplexed channels above 8 cost \$12.50 per channel.

For example, the total monthly cost of the multiplexor concentrator at node B of Figure 4.3(a) would be \$100. At node I of Figure 4.3(b) the monthly cost of the multiplexor concentrator would be \$275. For Figure 4.3(c) the concentrator at node K would cost \$312.50 per month. If in any of these situations, a single MX were not adequate to handle the devices being concentrated, the costs would be simply scaled upward by $\$100 \times N$ per month, N being the minimum number of

¹The author would like to acknowledge the cooperation of the Comshare Corporation, Ann Arbor, Michigan, which provided especially useful data.

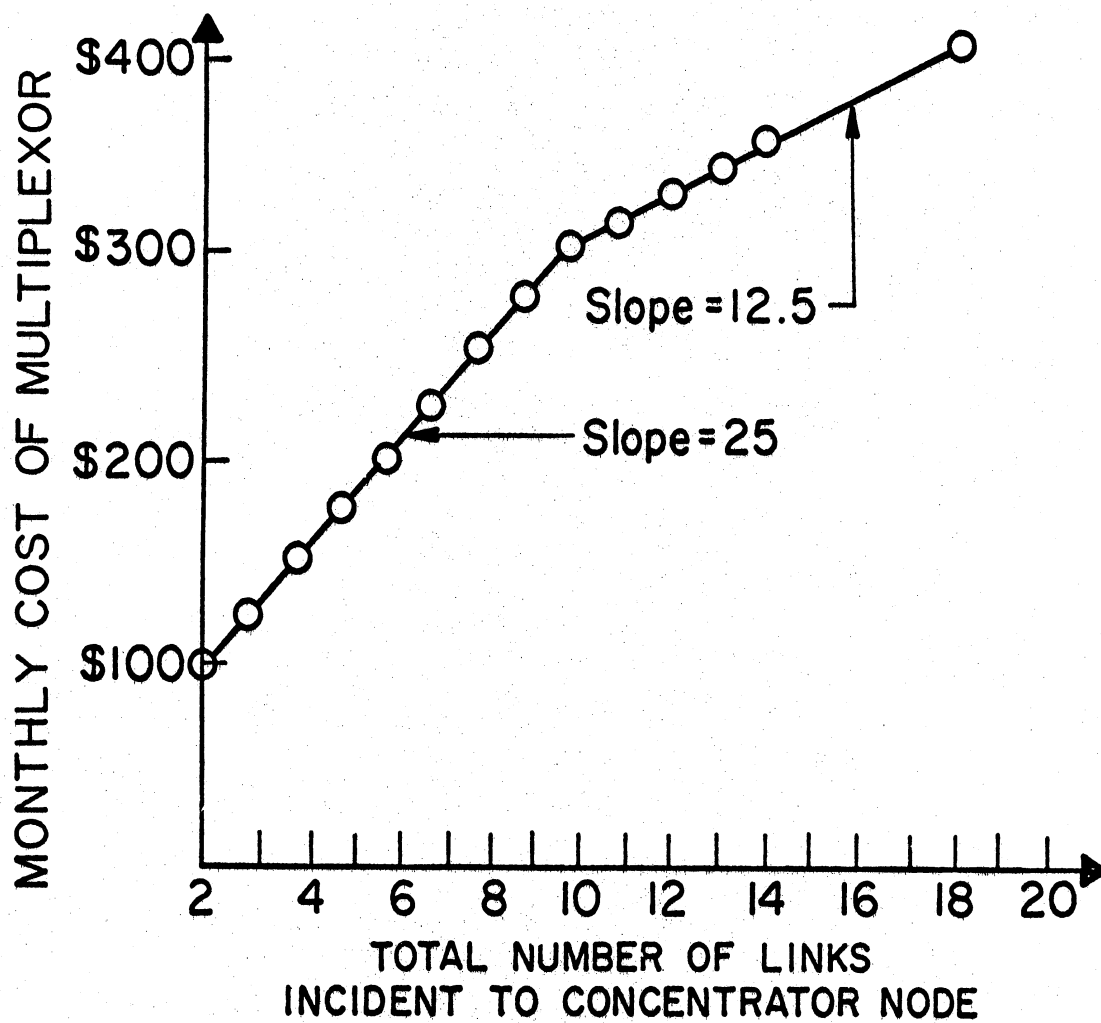
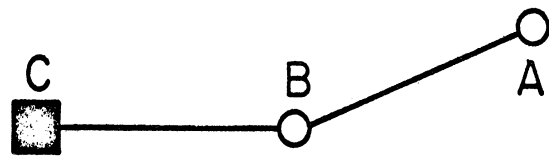
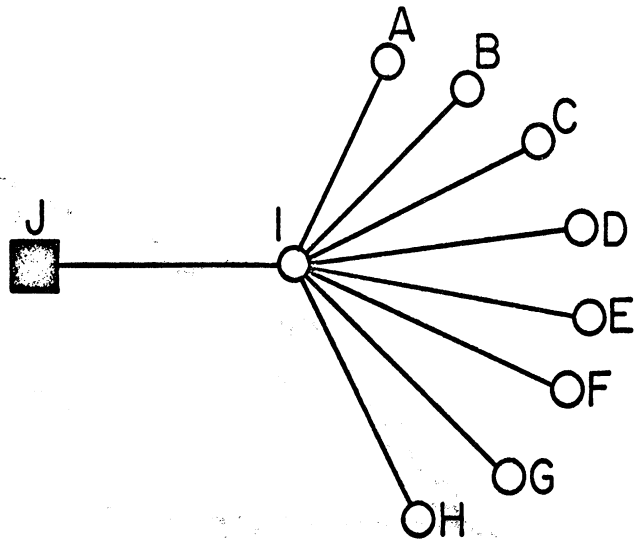


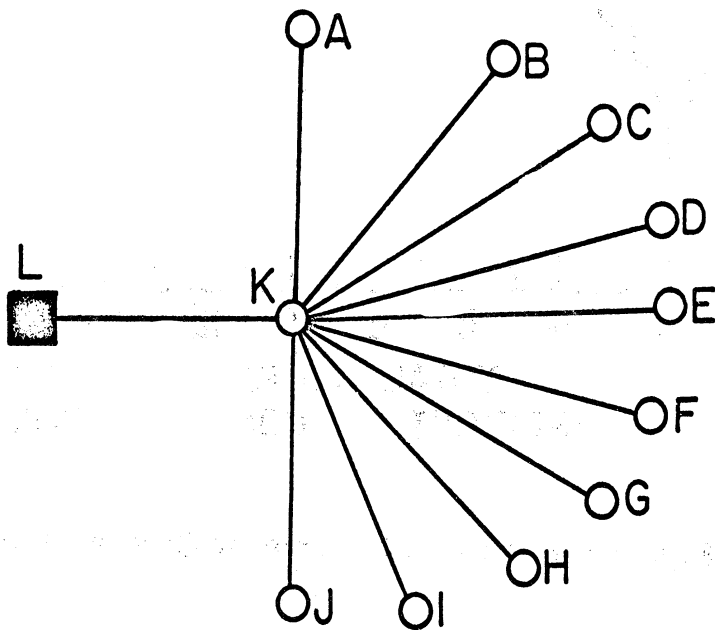
Figure 4.2 Cost of Multiplexor Concentrators



(a)



(b)



(c)

Figure 4.3: Some Example Configurations

additional MX's which would be capable of handling the required load, if added at the concentrator node.

4.2.2.2 Message Switching Center Concentrator Costs

Since the message switching concentrators contain computational capabilities and are, in general, electronically more sophisticated than multiplexors, they are significantly more costly and hence most cost-effective at nodes with large numbers of incident links. The relatively large fixed cost component on the MSC cost curve of Figure 4.4 reflects the fact that software, processor unit, storage, and control logic costs usually dominate the cost of the MSC device. The results of the simulation studies described in Section 3.2.2 indicate that the buffer storage capacities of typical minimally-configured general purpose computers which might be used in such applications are far more than adequate to satisfy assumption 4 of Section 2.3, even in the most highly congested cases. A close look at minimal hardware configurations typically used in these applications will reveal that the devices become saturated or limited by processor speed constraints (execution speeds) long before buffer saturation becomes a factor.¹ Stated another way, the maximum

¹ For example, in the Digital Equipment Corporation Model 680 communications processor [79] handling 110, 8 bit per character - 110 baud teletypewriters, about 92% of the processor's time will be required just for successful line scanning operations. Control software, including packing and unpacking routines, requires not more than 1500 words of memory which leaves approximately 2500 words or 3750 characters of memory left for buffer storage. The simulation studies of Chapter 3 revealed that only 4000 bits or 500 characters of buffer are needed to satisfy the "infinite buffer" assumption in a typical 90 teletypewriter case.

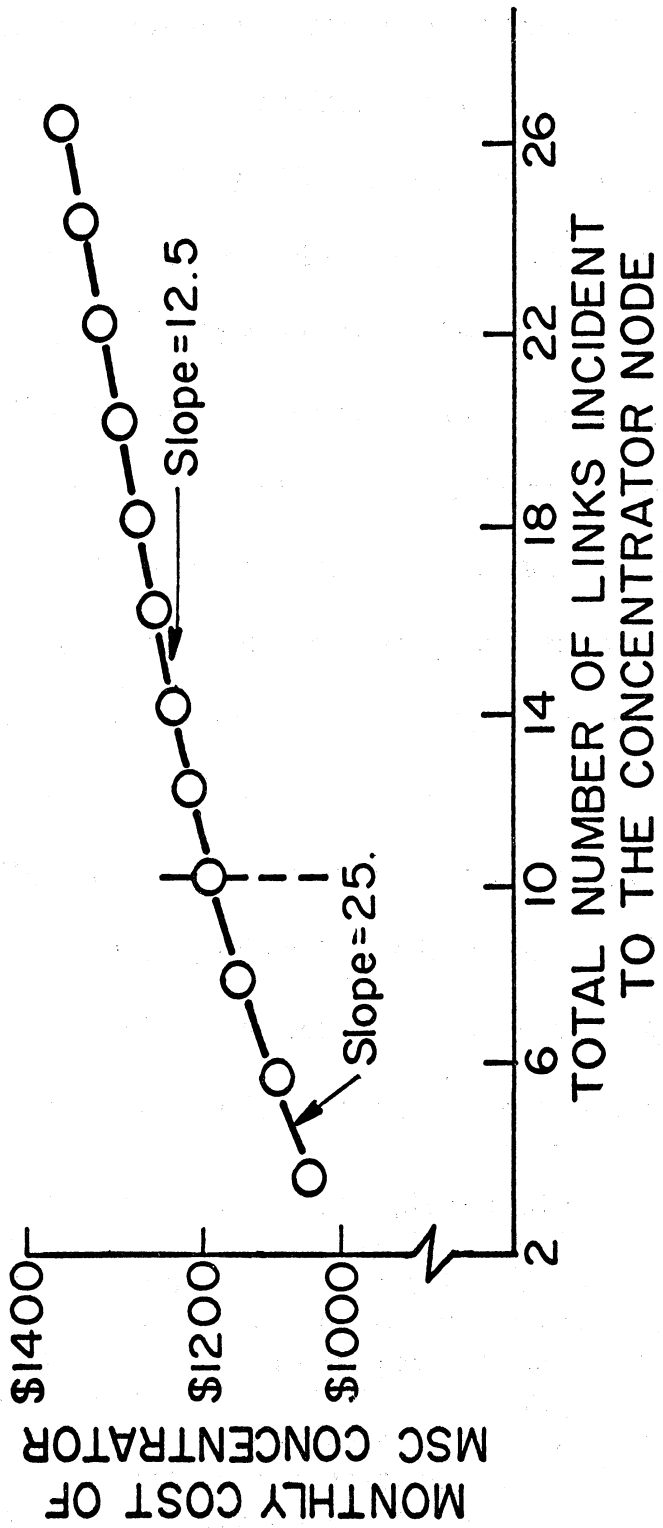


Figure 4.4 MSC Cost Function

number of remote terminals which can be serviced by a given MSC concentrator is limited by the speed of the CPU and not by buffer size limitations. For this reason, the MSC cost function of Figure 4.4 does not indicate a dependence on the size of the buffer. Since state-of-the-art hardware technology indicates that MSC's are typically capable of handling the equivalent of about 100 teletypewriters, this value has been used for the limiting number of equivalent teletypes which a given MSC can handle.

Thus any MSC which concentrates fewer than the equivalent of 100 teletypewriters will not require any cost components other than those of Figure 4.4 since the speed limitations of a single device are not constraining in this instance.

Having established the cost functions for these critical design variables, we complete our efforts in characterizing the important design tradeoffs by presenting existing design procedures in detail. A working knowledge of these procedures and their inherent limitations is essential to an understanding of the material in ensuing chapters.

4.3 The Esau Williams Algorithm¹

In this section we will consider the Esau Williams procedure in substantial detail. This particular design algorithm easily represents the most realistic and significant contribution to the teleprocessing network design effort to date. However, it is concerned with only the one particular tradeoff between topological structure and line cost.

¹ See Reference [53].

The solution configuration obtained will, in general, be suboptimal. Furthermore, no consideration is given to response time, data concentrator costs or performance, economies of scale in the cost of capacity, or the effects of using different capacity values for the different links of a structure.

Nonetheless, this algorithm provides a good deal of insight into the design tradeoffs which must be considered as topological structure is varied. It should be emphasized that the chief limitations of the procedure all stem from the fact that several significant variables in the design problem have not been considered. In Chapters 5 and 6 of this research, it will be shown that the solutions obtained by the Esau Williams procedure are consistently inferior to those of procedures developed in this work which consider the additional important design variables.

The objective of the Esau Williams algorithm is to obtain a configuration which connects all remote terminals to the central facility and is "reasonably optimal" as far as the cost of communication lines is concerned. The completely point-to-point network structure is assumed to be the most expensive configuration since each terminal is connected directly to the central facility. The basic gambit is to systematically form multipoint segments in an iterative fashion by deleting some central link (a link incident to the center) and adding a less costly noncentral link which preserves the connectedness of the network. The formation of multipointed branches at each iteration

reduces line costs by an amount equal to the difference in the costs of the central link deleted and the noncentral link added. For example, in Figure 4.5(a), it is possible to multipoint terminals C and D on line C and, as shown in Figure 4.5(b), realize a line savings of $k(\ell_{DE} - \ell_{CD})$ where k is the cost per unit length of that particular capacity link.

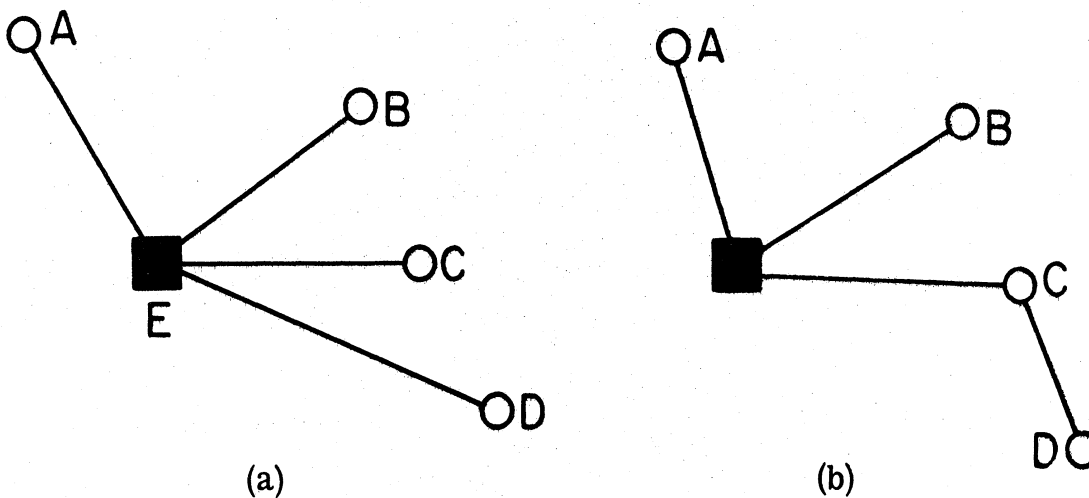


Figure 4.5 An Example Illustration of an Iteration in the Esau Williams Design Algorithm

A still greater savings could be achieved by placing all three terminals B, C, and D on the same multipoint segment having central link C. In general, if a savings is possible, it is assumed attainable by removing a central link and adding a noncentral link which preserves the network connectivity.

To determine the order in which modifications should be made, define $\Delta L(X, YZ)$ as the line savings associated in removing central

link X and adding link YZ, where Y is assumed to be in the same segment as X and Z is not in the same segment as X. For each particular central link in a configuration, there may be several ΔL 's, one for each noncentral link that would restore connectedness if added after the deletion of the given central link.

At the beginning of the procedure there are as many segments as there are remote terminals. Then as multipointed segments are formed, the number of central links (or equivalently, segments) will decrease. Let each segment therefore be identified by the remote terminal which is connected directly to the central facility. At any point in the procedure, assume all of the ΔL 's are computed for a given segment, and let $\hat{\Delta L}$ denote their maximum. Moreover, let $\hat{\Delta L}$ denote the maximum for all segments. For example, if $\hat{\Delta L} = \Delta L(D, DC)$ in the structure of Figure 4.5(a), then the configuration would be modified as shown in Figure 4.5(b). Iterations of this type are then repeated.

The method assumes that maximum line loading factors are given and will not allow any configurations to be formed in which a central link would be loaded in excess of this maximum value. Removal of a central link may render certain ΔL 's invalid. For example, if $\hat{\Delta L}$ for segment D in Figure 4.5(a) is $\Delta L(D, DC)$, this ΔL is not defined for configurations such as Figure 4.5(b) where central link D does not exist.

In computing ΔL 's for segment X in relation to the segment containing Z, all remote terminals in both segments must be considered. For example, in Figure 4.6 in computing ΔL for segment A in relation to segments C and E, the following ΔL 's would be computed, traffic volumes permitting:

$$\Delta L(A, AC)$$

$$\Delta L(A, AD)$$

$$\Delta L(A, BC)$$

$$\Delta L(A, BD)$$

$$\Delta L(A, AE)$$

$$\Delta L(A, BE)$$

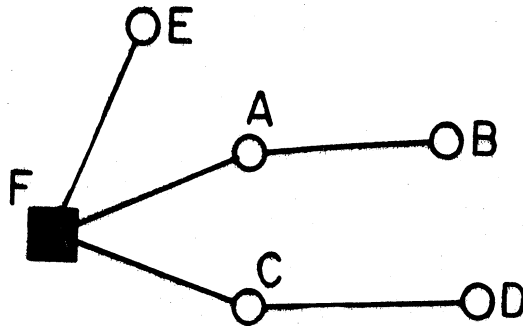


Figure 4.6 An Example Configuration

The remaining ΔL 's for Figure 4.6 would be computed as follows:

For segment E:

$$\Delta L(E, EA)$$

$$\Delta L(E, EB)$$

$$\Delta L(E, EC)$$

$$\Delta L(E, ED)$$

For segment C: $\Delta L(C, CA)$
 $\Delta L(C, CB)$
 $\Delta L(C, CE)$
 $\Delta L(C, DA)$
 $\Delta L(C, DB)$
 $\Delta L(C, DE)$

The algorithm simply consists of choosing the structural modification at each step in the procedure which gives rise to the maximum ΔL for all segments. We now present the algorithm.

Algorithm

Let k be an index on the structures generated by the procedure so that in a network having a total of n terminals including the central facility, the k -th structure will contain exactly $n-k$ central links.

Figure 4.7 depicts this variation of topology as a function of the index k .

Let X be a variable which ranges over the set of central links in a given structure k . Let Y be a variable which ranges over the set of all remote terminals in segment X . Let Z be a variable which ranges over the set of all remote terminals not in segment X . For initialization purposes, set $k=1$ and let structure 1 be the completely point-to-point network:

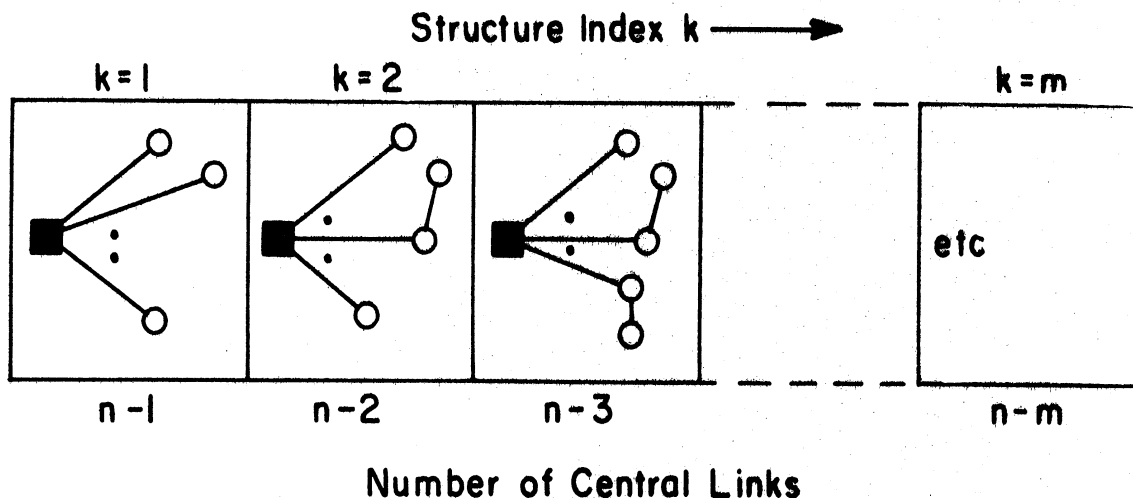


Figure 4.7 Variation of Topological Structure Using the Esau Williams Design Algorithm

- Step 1: Define the range of the variable X ; set all $\hat{\Delta L}(X) = 0$.
Initialize the variable X .
- Step 2: Define the range of the variable Z ; initialize the variable Z .
- Step 3: Define the range of the variable Y ; initialize the variable Y .
- Step 4: Can the segment containing Z handle the traffic of segment X ?¹ If not, go to step 8.
- Step 5: Compute $\Delta L(X, YZ)$. Is $\Delta L(X, YZ) > \hat{\Delta L}(X)$? if not, go to Step 7.
- Step 6: Set $\hat{\Delta L}(X) = \Delta L(X, YZ)$; preserve the X, Y , and Z which give this $\hat{\Delta L}$.
- Step 7: Having all Y 's been considered? If not, augment the Y variable by one and go to Step 5.

¹ The authors do not characterize traffic in concise mathematical terms. Presumably this traffic constraint is stated in terms of the steady flow capacities for the links.

- Step 8: Having all Z's been considered? If not, augment the Z variable by one and go to Step 3.
- Step 9: Having all X's been considered? If not, augment the X variable by one and go to Step 2.
- Step 10: Search the $\hat{\Delta L}(X)$'s over all X to find $\hat{\Delta L}$. Is this $\hat{\Delta L} > 0$?
If not, terminate the procedure.
- Step 11: Let the values of the variables X, Y, Z which give $\hat{\Delta L}$ be denoted X^* , Y^* , Z^* . Form structure k+1 from structure k by deleting central link X^* and adding noncentral link Y^*Z^* . Augment k by 1. Terminate if k equals the number of remote terminals in the network. Otherwise, go to step 1.

It has been previously noted that this algorithm does not obtain the theoretically minimum line cost network. This is so since it is assumed that a "greedy" procedure which maximizes the amount of line savings at each iteration will lead to the best network after all iterations have been completed. Since large numbers of possible structures are never considered, there can be no assurance that the true minimum line cost network will be found.

There are several other sources of nonoptimality inherent in the Esau Williams approach to network design, some of which have been noted previously. For example, the Esau Williams design technique assumes that only one line speed is available. The authors, however, point out that it is not obvious which speed of line will form the most economical configuration. They also note that a combination

of lines operating at different speeds will usually be most economical. This is so because it is desirable to match the link capacity value to the traffic which a link carries. Unfortunately, no further treatment of this important issue is given by Esau and Williams. Furthermore, since data concentrator costs are not considered, the solutions obtained will not in general be truly optimal whenever concentrators are used. We will address these issues in greater depth in the following chapters.

An alternate network design procedure which also has a limited degree of utility is the well-known minimum spanning tree algorithm of Kruskal [76]. This algorithm like the one due to Esau and Williams, does not address the issues of data concentrator costs, variable link capacity values, or of network performance. In certain special design situations where none of these factors happens to be important, however, this algorithm can be used quite effectively.

Unlike the Esau Williams algorithm, the minimum spanning tree algorithm does obtain a theoretically optimal solution to the problem of minimizing line costs in a network. Another significant difference between the procedures is the fact that the Esau Williams method enables the designer to specify maximum acceptable line loadings and thereby prohibit the formation of configurations having overloaded links. With the minimum spanning tree algorithm, no such feature is available. However, the chief limitations of the minimum spanning tree procedure stem from the fact that it is usually not justifiable to ignore the aforementioned factors other than line costs in realistic

formulations of the design problem.

4.4 The Minimum Spanning Tree Algorithm¹

The minimum spanning tree algorithm can be used to obtain the theoretical minimum line cost network which provides an information path from each terminal to the central processor location. Assume that the cost $d(e_i)$ of using link e_i is known for all possible links in the network of n nodes. This implies that only one capacity value can be considered for a given link.

Form the sequence of links $E = \{e_1, e_2, \dots, e_N\}$ such that $d(e_i) \leq d(e_j)$ for $j > i$. S denotes the set of links included in the solution; R denotes the set of those links rejected.

Step 0: $i=0$, $S = \emptyset$, $R = \emptyset$ where \emptyset is the null set

Step 1: $i=i+1$

Step 2: Consider link e_i . If e_i does not form a cycle with the set S of links already chosen for the solution configuration, add e_i to the set S . Otherwise permanently reject link e_i by adding it to the set R .

Step 3: Is the cardinality of S equal to $n-1$? If not, go to step 1.

Otherwise terminate the procedure.

At the end of the procedure, the set S contains the links which comprise the optimal solution. The following generalization of the MST algorithm, due to Prim [77], should be noted for the sake of completeness.

¹ See Kruskal, Reference [76].

Theorem 4.1

The minimal cost connecting network obtained from the MST algorithm also minimizes all monotonically nondecreasing symmetric functions of link costs among all connecting networks with no cycles.

Realistically, the cost of a network usually depends on link costs and concentrator costs. However in some practical situations where the costs of data concentrators are "small" in relation to the cost of links, the MST algorithm produces a reasonably good approximation to the true optimal solution considering both line and concentrator costs. We have chosen to illustrate this point by assuming relatively small concentrator costs when using the minimum spanning tree as one of the design strategies for the comparative case studies in Chapter 7.

Chapter V

EFFICIENT UTILIZATION OF THE LINK CAPACITY RESOURCE

5.1 Introduction

Having completed the characterization of the cost functions and the discussions of existing design procedures, we are now ready to consider the important issue of how link capacity resources should be allocated in a network whose structure and types of concentrators are known. This particular problem will become quite significant in the network design efforts which follow in Chapter 6 when topological structure and types of concentrators are finally treated as variables.

Attention is placed on examining this link capacity allocation problem in the context of the Esau Williams and minimum spanning tree procedures. It will be shown that since these techniques do not treat link capacity as a design variable, their solutions will not generally obtain efficient utilization of the link capacity resource.

We will illustrate that the optimum passage capacity assignment procedure originally developed in Chapter 3 can be used to determine the exact amount of excess or unused capacity, if any, which exists in the links of a given network. The existence of excess capacity in a network indicates that a given configuration is definitely non optimal in a cost sense since lower capacity values could be used in one or more links without exceeding a required minimum response time.

We commence these efforts by summarizing the key issues to be treated in this chapter in the form of two specific questions which the network designer must be able to answer. These are:

- 1) Is it possible to reduce one or more link capacity values in a given topological structure without changing the average response time and if so, how should link capacities be systematically reduced so that maximum cost savings are realized?
- 2) If a particular network configuration must have an average response time which does not exceed T_{\max} , a given maximum acceptable value, how should link capacity values be assigned in the least costly manner so that the average response time constraint is not violated?

In accordance with the realistic cost-capacity function presented in Section 4.2, it will be assumed that a multiplicity of discrete capacity values from the set $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_N\}$ are available, where typical values are $\Phi_k = 600(1-q) \times k$ bits/sec for $k = 1, 2, \dots, N$ and $1-q$ is an efficiency factor discussed in Section 1.3.1¹. N is the total number of values available.

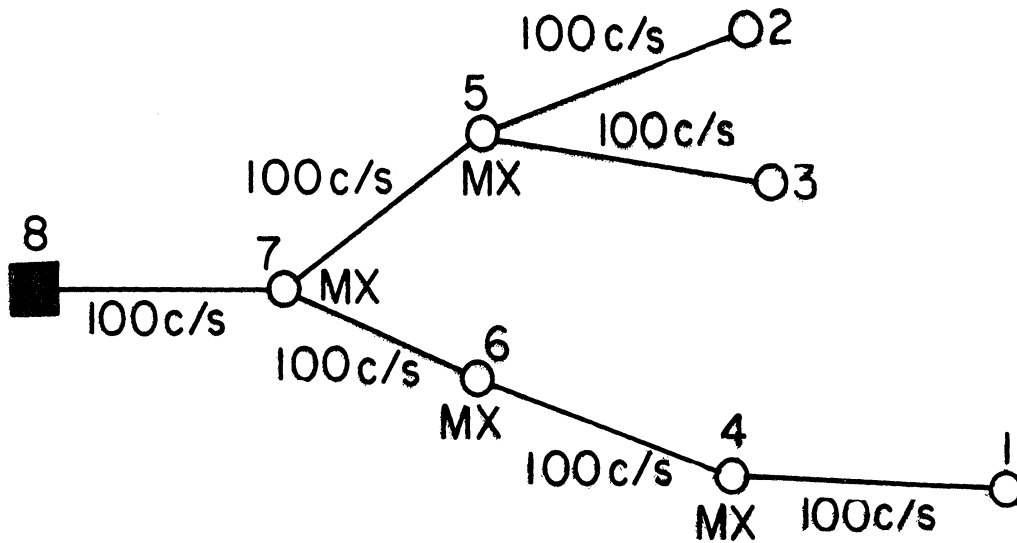
We answer the first of these two questions by recalling the implications of the optimum passage capacity assignment procedure.

¹ This range is consistent with that proposed by the USA Standards Committee X3 Computers and Information Processing in Reference [78]. For voice grade application, $N = 16$. No standards have yet been issued for values above voice grade.

5.2 Excess Link Capacity Removal Procedures which Preserve a Given Average Response Time

Consider an arbitrary network structure in which the link capacity values and types of concentrators are specified. Recall that the computational routine of Section 3.4 determined optimal channel capacity assignments throughout a network by equalizing the capacities of all passages which comprise the channels in the network. In each link, however, the sum of all optimally assigned passage capacity values will not necessarily be equal to the total link capacity. Thus if one or more MX concentrators are used in the network, it may be possible to reduce the capacity values in one or more links without changing the average response time. This operation is not possible, however, if MSC concentrators are used exclusively.

For example, we observed in the single link capacity value example of Section 3.3.2.2 (which is duplicated in Figure 5.1) that a great deal of capacity was being wasted in the less-congested links at higher levels of concentration. The optimal passage capacity assignment procedure used no more than 14.28 c/s in links 1, 2, and 3; no more than 28.55 c/s in link 4; and no more than 42.84 units in links 5 and 6. Thus the capacity links 1, 2, and 3 could be reduced to 14.28 c/s without affecting the network's average response time.



$\hat{T} = 2.333$ sec/mess, the minimum average response time
 $\gamma_i = 1$ mess/sec $i = 1, \dots, 7$ $\frac{1}{\mu} = 10$ char/mess

Figure 5.1 Network for the Example of Section 3.3.2.2

Similarly, values as small as 28.56 c/s in link 4 and 42.84 c/s in links 5 and 6 would not change the performance from that obtained when 100 c/s is used for all links. Thus, any fraction of the link capacity not required in the solution given by the optimal passage capacity assignment procedure may be discarded.

Stated mathematically, when considering the i th link in a given topological configuration, let $\Phi_{z(i)}$ denote the given capacity value for link i ; $z(i)$ is merely an integer indicating which particular value from the set Φ is used for the i th link.¹ Let S_i denote the set of all

¹ We continue to confine our discussions to situations in which one of the discrete link capacity values from the set $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_N\}$ is being considered. N denotes the total number of discrete values available and the components are ordered so that $\Phi_x \leq \Phi_{x+1}$ for all integers $1 \leq x \leq N-1$.

remote terminals whose forward message traffic is transmitted over link i without passing through any intermediately located MSC's.^{1, 2} Furthermore assume that the computational procedure of Section 3.4 has been executed to obtain the minimum average response time. In determining the channel capacities $\{C_i\}$ of expression (3.31) needed to calculate this response time value, optimal passage capacities $\{\hat{\alpha}_{gi}\}$ are computed for all remote terminals $g \in S_i$ and for all links i .

Define for the i th link of the structure in question:

$$\Phi_{x(i)} \triangleq \Phi \left[\sum_{g \in S_i} \hat{\alpha}_{gi} \right]$$

where $\Phi[y]$ is used to denote the smallest capacity value from the set Φ which is not less than y ; $x(i)$ is merely an integer which denotes the particular value from the set Φ which is equal to $\Phi \left[\sum_{g \in S_i} \hat{\alpha}_{gi} \right]$ for link i . Stated another way, let y be the sum of the optimal passage capacity values in link i . Then $\Phi_{x(i)}$ is the smallest value of capacity which could be used for link i instead of $\Phi_{z(i)}$ that would not change the average response time. Thus, if $x(i) < z(i)$ then by

¹ If a MSC is used at terminal i , S_i contains only the remote terminal i . If a MSC is used at terminal j and $j \in S_i$ then no terminals which feed traffic into node j from higher levels are included in S_i . Since a MSC functions exactly as a single independent source, the passage capacity assignment procedure assures that the effective arrival rate of this equivalent single source includes all external arrival traffic from nodes at higher levels of concentration in the same segment.

² This definition is perfectly consistent with the definition of S_i given previously in the algorithm of Section 3.3.2.1; only we have now generalized to define S_i for links in networks containing both multiplexors and MSC's.

using $\Phi_{x(i)}$ instead of $\Phi_{z(i)}$ for the capacity of the i th link of the configuration, the total network cost is reduced without changing the average response time. A cost savings is thereby realized, and is equal to the different between the costs of having $\Phi_{z(i)}$ units of capacity and of having $\Phi_{x(i)}$ units in link i . The total cost savings associated with this excess-capacity trimming operation is obviously the sum of the savings which accrue from each of the links whose capacity can be so reduced.

In conclusion, it should again be noted at this excess capacity trimming operation only involves changes in the link capacity values; the structure, types of concentrators and average response times for the configuration are not changed as the cost reduction is effected.

There are however instances of design applications in which the designer will only be concerned with guaranteeing that the average response time is less than some maximum acceptable value, T_{\max} . Thus he may want to consider improvement procedures which do not necessarily preserve the average response time but allow it to vary as long as this average response time constraint indicated by T_{\max} is satisfied. We now consider this particular topic.

5.3 Link Capacity Variation Procedures which Allow Average Response Time to Vary within Constrained Limits

At this juncture, we direct our efforts toward the development of a systematic procedure for selecting link capacity values so that a given network configuration has an acceptable average response time and a minimum line cost.

In a network which has n remote terminals and for which N discrete values of link capacity are available to the designer, there are exactly N^n possible combinations of link capacity assignments, if average response time constraints do not exclude any of these possibilities. We seek the minimum cost set of capacity assignments which satisfy the average response time constraint; hence some fraction of the N^n total combinatorial possibilities may not be feasible regardless of cost. Furthermore, if some feasible solution can be determined whose cost is "fairly close" to the true minimum, then it may be possible to ignore a sizeable fraction of the feasible solutions - those whose cost must exceed the cost for the "good approximation."

A technique is proposed for the solution of this problem which enables a number of these assignment combinations to be excluded or ignored without any consideration. These rejections correspond to assignments which must definitely be either nonoptimal or infeasible and hence not of interest. The original combinatorial problem involving as many as N^n link capacity assignments is immediately transformable into one of reduced dimensionality. Before going further, we consider this transformation in detail.

It is assumed that some arbitrary feasible assignment of link capacities $\Phi_{z(i)}$ is given; the subscript $z(i)$ refers to the particular element of the set Φ which denotes the capacity of the i th link.

It is also assumed that the types of data concentrators are known.

The first step in the desired direction consists of using the optimal passage capacity assignment procedure in the manner described in Section 3.4. Having determined the minimum average response time obtainable for this set of starting capacity assignments, excess capacity is trimmed in the manner described in the previous section. We have now determined the set of $\Phi_{x(i)}$ where $\Phi_{x(i)}$ is the smallest value from the set Φ which could have been used instead of $\Phi_{z(i)}$ for the i th link without altering the average response time.

Further link capacity reductions are now effected in a successive manner which reduces the network cost and increases the average response time at each step of the reduction operation. At each step of the procedure, the capacity of each link in the network is reduced to the next lower value if the reduction would not load a given link in excess of ρ_{\max} , the maximum acceptable loading factor. If such an overload would result, the capacity value of such a link is not changed.

Formalizing this procedure, we let h be an index on the number of discrete reduction steps. Define

$$v_{i,h} \triangleq \text{Link capacity of the } i\text{th link, obtained by reducing the capacity from } \Phi_{x(i)} \text{ to } \Phi_{y(i,h)} \text{ where } y(i,h) = \max \{b(i), x(i) - h\} \text{ and } b(i) \text{ is the subscript which indicates the smallest capacity value from the set } \Phi \text{ which will not overload the } i\text{th link } (b(i) \geq 1). \quad (5.1)$$

This variation can be depicted schematically in Figure 5.2. The monotonic behavior of the cost and average response time variables as the index h is varied enables one to readily determine the largest value for h for which the capacity assignments will result in an acceptable average response time.

For any given value of h , let $T(\{v_{i,h}\})$ denote the average response time, as obtained by the method of Section 3.4 for the network having link capacity assignments $\{v_{i,h}\}$. Define \hat{h} to be the largest value of the index h for which $T(\{v_{i,h}\}) \leq T_{\max}$, the maximum acceptable average response time.

If $y(i, \hat{h}) = b(i)$ for $i = 1, 2, \dots, n$, then the optimal assignment of link capacity values has been found since no less expensive feasible possibility exists.

For certain, there exist at least the following number, $A(\hat{h})$ of nonoptimal assignments:

$$A(\hat{h}) = \left\{ \prod_{i=1}^n (N - y(i, \hat{h}) + 1) \right\} - 1 \quad (5.2)$$

These correspond to all capacity assignments in which one or more link i capacity values exceed $v_{i, \hat{h}}$ but none are less than $v_{i, \hat{h}}$ for the i th link. All such assignments must clearly be more costly than the one indicated by $\{v_{i, \hat{h}}\}$, $i = 1, 2, \dots, n$. Similarly there will be at least the following number of assignments that are assuredly infeasible:

$$B(\hat{h}) = \prod_{i \in \theta} (y(i, \hat{h}) - 1) \quad (5.3)$$

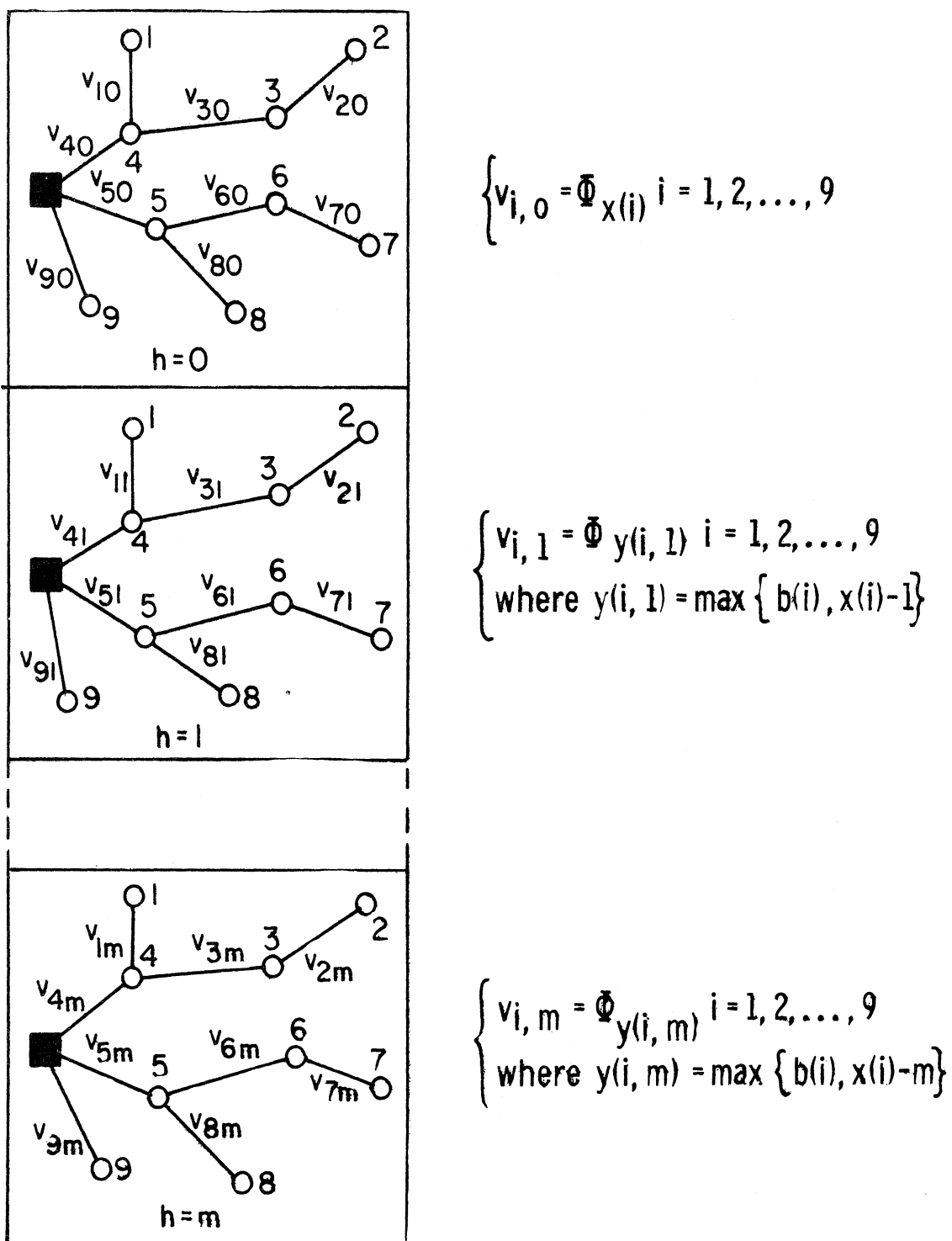


Figure 5.2 Illustration of Link Capacity Variation Procedure.
 Cost and Average Response Time Behave
 Monotonically as h is Varied.

where θ is the set of links i for which $y(i, \hat{h}) > 1$.¹ These correspond to those capacity assignments in which no link capacity values are larger than for a combination known to be infeasible.

At this point, we have a feasible solution of assignments indicated by the $\{v_{i, \hat{h}}\}$. If a less costly feasible solution exists, it must be found with an assignment combination not included in either of those subsets whose cardinalities are $A(\hat{h})$ and $B(\hat{h})$, respectively.

To be sure, the problem of finding any feasible capacity assignments better than the one given by the $v_{i, \hat{h}}$ (for $i=1, \dots, n$) is no less combinatorial in nature than was the original problem. However this smaller problem is more amenable to efficient solution for the following reasons:

1. The number of possible assignments which must be considered in the original problem has been reduced.
2. In the smaller problem, any assignment combinations having a cost exceeding that for the $\{v_{i, \hat{h}}\}$ solutions need not be considered.
3. In the smaller problem, easy-to-calculate lower bounds on average response time can be used to rule out certain groups of assignment combinations which could not possibly satisfy the average response time constraint.

¹ Since all links for which $y(i, \hat{h}) = 1$ must have the same capacity value for all assignment combinations counted in $B(\hat{h})$, θ includes only those for which $y(i, \hat{h}) > 1$.

Branch and bound techniques such as those described in reference [80] must be used in the solution of the smaller problem. In this reference [80] Lawler and Wood characterize these types of optimization techniques as procedures which involve intelligent searches of the space of all feasible solutions. More specifically, the authors note the essential features of such methods:

"There are many constrained optimization problems for which direct methods of solution either do not exist, or are not efficient. Problems may be of this nature because the objective function or constraints are non-convex, or some or all of the variables are restricted to discrete values. Branching and bounding enables us to solve these 'difficult' problems by applying existing methods of solution to 'easy' problems".

The basic gambit in such branch and bound algorithms consists of repeatedly partitioning the space of all feasible solutions into smaller and smaller subsets. At each step, a lower bound is calculated for the cost of the solutions within each subset. After each partitioning, those subsets with a bound that exceeds the cost of a known feasible solution are excluded from all further partitioning. The partitioning continues until a feasible solution is found such that its cost is no greater than the bound for any subset.

It should not be surprising that there is no one particular algorithm which is the best for all situations. As is pointed out by the authors, the selection of the right algorithm will depend on the computer available and most importantly on the particular problem at hand. The interested reader should consult reference [80] for a

detailed discussion of the tradeoffs involved in this selection.

For illustrative purposes, however, we will now consider an example problem and formulate a branch and bound type of algorithm which can be used to obtain its solution. This particular algorithm is meant to have a fairly general degree of utility, even though the efficiency with which it executes will depend on the application under consideration.

5.3.1 An Example Solution Using Backtrack Programming

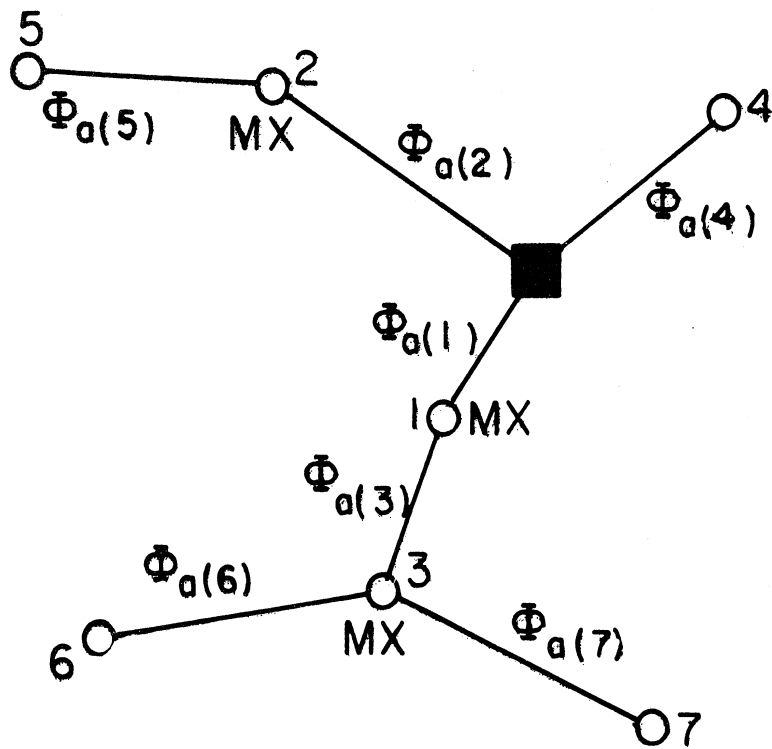
In this section, we will present a computational procedure for obtaining the minimum line cost assignment of link capacity which satisfies a response time constraint in a given network. We will use the $\{v_{i,h}^{\hat{}}$ solution to make the computational routine as efficient as possible.

Consider the fairly "simple" example network of Figure 5.3 with only seven discrete capacity values $\Phi = \{\Phi_1, \dots, \Phi_7\}$ where $\Phi_i = (600)(.75)^i$ and $i = 1, 2, \dots, 7$.¹ There are a total of $7^7 = 823,543$ possible assignment combinations.

One widely used method of branching and bounding known as backtrack programming [81] is a representative technique which we shall consider for this example.

Let $\Phi_{a(i)}$ denote the capacity value for the i th link for a particular value of the variable $a(i)$; the $\{a(i)\}$ are integer variables each of

¹ The structure under consideration appears again in Chapter 7 as the solution configuration for one of the case studies.



$\Phi_a(i)$ is the capacity of link i where $\Phi = \{\Phi_1, \dots, \Phi_7\}$
 and $\Phi_j = (600)(.75)^j$ for $j = 1, 2, \dots, 7$.

Figure 5.3 An Example Network

which denotes the capacity value from the set Φ which is being used for the i th link capacity. Let $D(a(1), a(2), \dots, a(7))$ denote the line costs for the capacity assignment $\{\Phi_{a(1)}, \Phi_{a(2)}, \dots, \Phi_{a(7)}\}$. Let $T(a(1), a(2), \dots, a(7))$ denote the average response time for the same capacity assignment. The optimization problem is to find a vector $\{a(1), a(2), \dots, a(7)\}$ which minimizes $D(a(1), a(2), \dots, a(7))$ subject to the constraint that $T(a(1), a(2), \dots, a(7)) \leq T_{\max}$. Each of the variables $a(i)$ can assume any integral value in the range $1 \leq a(i) \leq 7$ $i = 1, 2, \dots, 7$.

The brute force way is to form each of the 7^7 sample vectors, evaluate each one in D and T and see which one produces the smallest D and has $T \leq T_{\max}$. Backtrack programming is designed to obtain the same answer in far fewer than 7^7 trials.

The basic idea behind backtracking is to build up the sample vector one component at a time and use bounds to see whether the vector being formed still has any chance of success. If the partial vector¹ $\{a(1), \dots, a(j), \dots\}$ is already seen to be inherently suboptimal, then 7^{7-j} possible test vectors can be ruled out in one fell swoop without having to examine them individually. The effectiveness of a backtrack program is determined largely by the skillfulness with which the bounds are selected and programmed, since weak bounds

¹ The notation $\{a(1), \dots, a(j), \dots\}$ means that the first j components of the vector are specified and the last $7-j$ are unspecified. Thus in computing bounds we are free to force all the unspecified variables to assume appropriate extremal values so that the bounds may be calculated.

will lead to too many cases being considered.

More specifically, note that the number of test vectors which can possibly be ruled out is determined by the value of j in the partial vector under consideration. If such exclusions can be made, the maximum number will result if the decision can be reached with a minimally specified partial vector (one with a minimum number of components). Hence the manner in which the links of the network are ordered for consideration in the partial vector will clearly affect the efficiency of the procedure. Before considering this question further, however, it is necessary to define the precise bounds which will be used. Once the bounds are characterized, then we will be in a position to complete the discussions concerned with ordering the components of the partial vector.

The lower bound on cost, $D_{LB}(a(1), \dots, a(j), \dots)$ is defined to be the line cost of the capacity assignment $\{\Phi_{a(1)}, \dots, \Phi_{a(j)}, \Phi_1, \dots, \Phi_1\}$. This bound does not depend on any network parameter values except line costs. However for the response time bound there will be a dependence on the types of concentrators used. Since the most general case of a network containing both MX and MSC concentrators is most readily treated by considering several all-multiplexor subnetworks independently, we will first develop the response time bound for the all-multiplexor case. If a network has MSC's then the appropriate modifications which will be presented in the next section should be used. For the all multiplexor network, let $T_{LB}(a(1), \dots, a(j), \dots)$ be

defined as follows:

$$T_{LB} \triangleq \frac{1}{\gamma} \sum_i \left\{ \sum_h \left(\frac{\sqrt{\frac{\gamma_h}{\mu}} \sum_k \sqrt{\frac{\gamma_k}{\mu}}}{\Phi_{a(i)} - \sum_k \frac{\gamma_k}{\mu}} \right) \right\} \quad (5.4)$$

where the i summation is taken over all central links and the h and k summations over all remote terminals which share central link i . If $\Phi_{a(i)}$ is not defined (if $i > j$) then set this capacity value to the maximum possible by setting $a(i) = 7$. We now discuss briefly how this bound for average response time has been obtained.

5.3.1.1 Derivation of Response Time Lower Bound

Expression (5.4) pertains only to networks having MX concentrators exclusively, so we shall discuss the case first. Then we will consider modifications for those situations where one or MSC's are used. The approach to be taken here will involve calculating the overall network lower bound from lower bounds developed for each of the constituent segments in the network.

Note that we can express the average response time T for any network as follows:

$$T = \frac{1}{\gamma} \sum_{\substack{\text{central} \\ \text{links } i}} \gamma^{(i)} T^{(i)} \quad (5.5)$$

where $\gamma^{(i)} = \sum_{j \in I} \gamma_j$, I being the set of all remote terminals in the segment having central link i , and $T^{(i)} = \frac{1}{\gamma^{(i)}} \sum_k \left(\frac{1}{\frac{C_k}{\gamma_k} - 1} \right)$.

The k summation is taken over all channels in the network segment for which i is the central link. To obtain a lower bound for T we will simply derive expressions for minimum values for each of the $\gamma^{(i)}$ $T^{(i)}$ terms in expression (5.5).

We obtain these minimum values by considering each segment individually and noting what happens in the tandem multiplexor assignment procedure of Section 3.3.2.1 when passage capacity values are being assigned for central link i . In a given segment, the passage capacities on the central link are the last ones to be assigned by the module ASSIGN. One of two situations occurs at Step 2 of the ASSIGN module when $k = 1$. If $B_1 = \emptyset$, that is, if no passages on the central link are input limited, then the minimum average response time for the entire network segment, $\hat{T}^{(i)}$, is given directly by expression (3.7) after the parameters are properly reinterpreted:

$$\hat{T}^{(i)} = \frac{1}{\gamma^{(i)}} \left\{ \sum_h \frac{\sqrt{\frac{\gamma_h}{\mu}} \sum_k \sqrt{\frac{\gamma_k}{\mu}}}{\Phi_{a(i)} - \sum_k \frac{\gamma_k}{\mu}} \right\} \quad (5.6)$$

where the h and k summations are taken over all remote terminals which share central link i and $\Phi_{a(i)}$ is the value of the i th component in the capacity vector $\{\Phi_{a(1)}, \dots, \Phi_{a(j)}, \Phi_7, \dots, \Phi_7\}$.

On the other hand if $B_1 \neq \emptyset$ when the module ASSIGN executes step 2 for $k = 1$, then the RHS of expression (5.6) does not represent the average response time $T^{(i)}$ for the segment in question. However,

it does represent a lower bound for $T^{(i)}$. This can be shown by recalling that the module ASSIGN performs the task of determining passage capacity values $\{\alpha_{ji}\}$ for all terminals j which share link i so that

$$T^{(i)} = \frac{1}{\gamma^{(i)}} \left[\sum_{j \in J} \left(\frac{1}{\frac{\min\{\alpha_{ji}, \alpha_j\}}{\gamma_j}} \right) + \frac{1}{\frac{\alpha_{ii}}{\gamma_i} - 1} \right]$$

is minimized. J is the set of all terminals excepting terminal i which share link i ; α_j denotes the current capacity value for the passage serving terminal j on one of the links feeding traffic directly to link i .

Recall that in the proof of Theorem 3.1 we introduced the function

$$T^* = \frac{1}{\gamma^{(i)}} \sum_{j \in I} \left(\frac{1}{\frac{\alpha_{ji}}{\gamma_i} - 1} \right)$$

which has the property that $T^* \leq T^{(i)}$ for any capacity assignment $\{\alpha_{ji}\}$, $j \in I$ where I is the set of all terminals, including terminal i , that share link i . It was also shown that the minimum value which T^* can have is given by the RHS of expression (5.6). Since $T^* \leq T^{(i)}$, it follows that $T^{(i)}$ can likewise never have a value less than the one given by expression (5.6).

Since $\gamma^{(i)}$ is just a constant, the minimum value of each of the $\gamma^{(i)}T^{(i)}$ product terms in expression (5.5) is determined simply by multiplying both sides of (5.6) by $\gamma^{(i)}$. Thus expression (5.4) for the overall network lower bound T_{LB} is given by summing $\gamma^{(i)}\hat{T}^{(i)}$ values over all central links:

$$T_{LB} = \frac{1}{\gamma} \sum_i \left\{ \sum_h \left(\frac{\sqrt{\frac{\gamma_h}{\mu}} \sum_k \sqrt{\frac{\gamma_k}{\mu}}}{\Phi_{a(i)} - \sum_k \frac{\gamma_k}{\mu}} \right) \right\}$$

where the i summation is taken over all central links and h and k summations over all remote terminals sharing the i th central link.

If none of the passages on any central link are limited at higher levels of concentration by application of the hierarchical assignment procedure, then T_{LB} is the concise value of average response time, not just a lower bound. Thus expression (5.4) is the best possible bound (is an exact expression) for the average response time in these situations and is a significant computational improvement over the hierarchical procedure for considering each concentrator node in the network. This derivation of expression (5.4) applies only for networks which use MX concentrators exclusively.

If a combination of the MX and MSC type concentrators is used, the above derivation needs to be modified somewhat. Recall that in

Section 3.4 we noted that MSC's partition the network into independent subnetworks, each of which contains only source, sink, and MX concentrator nodes. Not surprisingly, therefore, the lower bound T_{LB}^* in these cases can be readily calculated from

$$T_{LB}^* = \sum_i T_{LB_i}$$

where T_{LB_i} is the expression (5.4) lower bound which would be calculated for the i th independent subnetwork and the i summation is taken over all such subnetworks.

If no MX concentrators are used at all (MSC's are used exclusively) the calculation of lower bounds for average response time is not necessary. In this case expression (2.8) can be used to directly obtain the actual average response time since channel and link capacity values are equivalent throughout the network. This completes the derivations of the lower bounds and enables us to now formally describe the computational procedure.

5.3.1.2 The Computational Procedure

The manner in which the component variables of the partial vector $\{\Phi_{a(1)}, \dots, \Phi_{a(j)}, \dots\}$ are ordered has been shown to significantly affect the efficiency of the computations. Now that the expressions for the response time bound have been developed, we are in a position to complete the discussion of how the partial vector should be ordered since this is the first step which must be undertaken in implementing the algorithm. We shall present the algorithm for the network using

multiplexors exclusively. Extensions of this algorithm for the case where one or more MSC's are used would be straightforward now that we have discussed how the response time bound calculations should be modified in these situations.

In order to minimize the number of bound calculations (for D_{LB} and T_{LB}) it is desirable to give those links which affect these bounds most significantly the lowest indices in the partial vector. For the response time bound T_{LB} presented in expression (5.4) this is best accomplished by assigning the lowest indices of the partial vector to central links in the network since central link capacity values appear in expression (5.4) and others do not. For the cost bound D_{LB} , the longest links should be given the lowest indices in the partial vector since they will have the most significant affect on the bound. Still another factor which must be considered in this ordering arises as a consequence of the operation performed at Step 5 of the algorithm to be presented. For reasons which will become evident shortly, those links having the capacity value Φ_1 in the $\{v_{i,h}\}$ solution should be given the highest indices in the partial vector. The ultimate decision on how the components of the partial vector should be ordered must be reached by intelligently compromising all of these factors. Since no hard and fact rules can be proposed which will be perfectly general, we will assume that some intelligent ordering which considers these tradeoffs is made as the first step for any particular problem at hand.

Let CMIN be a variable whose value is the cost of the network (lines and concentrators) for the least expensive capacity assignment considered up to any point in the procedure. CMIN is initialized by setting it equal to the total cost of the network having the $\{v_{i,h}^{\wedge}\}$ capacity assignment; the $\{v_{i,h}^{\wedge}\}$ assignment is obtained using the method described in Section 5.2. At the end of the procedure, CMIN is simply the total cost of the solution configuration.

At step 4 in the algorithm to be considered, any assignment which is in the set counted in expression (5.3) corresponds to one which is known a priori to be infeasible. This check can readily be made by comparing certain components of the assignment vector $\{\Phi_{a(1)}, \dots, \Phi_{a(7)}\}$ with the $\{v_{i,h}^{\wedge}\}$ solution components. Specifically, if $\Phi_{a(i)} < v_{i,h}^{\wedge}$ is true for those links i for which $v_{i,h}^{\wedge} > \Phi_1$, then the assignment vector in question is known a priori to be infeasible.

Similarly, at step 5 in the algorithm to be considered, any assignment which is in the set counted in expression (5.2) corresponds to one which is known a priori to be nonoptimal. Thus, if each component of the assignment vector $\{\Phi_{a(1)}, \dots, \Phi_{a(j)}, \Phi_1, \dots, \Phi_1\}$ is not smaller in value than its corresponding component in the $\{v_{i,h}^{\wedge}\}$ solution, then the present partial assignment $\{\Phi_{a(1)}, \dots, \Phi_{a(j)}, \dots\}$ is assuredly a nonoptimal partial assignment and no bounds need even be computed. This is why we seek to assign those link i for which $v_{i,h}^{\wedge} = \Phi_1$ to the highest index positions in the partial vector.

We now present the algorithm in detail; some example results follow this description.

Algorithm Description

- Step 1: Set $j = 1$
- Step 2: Set $a(j) = 1$
- Step 3: Is j less than the number of links in the network? If so, go to step 5.
- Step 4: Is $\Phi_{a(i)} < v_{i,h}^{\wedge}$ for those links i for which $v_{i,h}^{\wedge} > \Phi_1$? If so, go to step 8. This step transfers the execution to step 8 if the present assignment $\{\Phi_{a(1)}, \dots, \Phi_{a(7)}\}$ is known to be infeasible.
- Step 5: Is each component of the assignment vector $\{\Phi_{a(1)}, \dots, \Phi_{a(j)}, \Phi_1, \dots, \Phi_1\}$ greater than or equal in value to its corresponding component in the vector $\{v_{i,h}^{\wedge}\}$? If so, go to step 7. This step transfers the execution to step 7 if the present partial assignment $\{\Phi_{a(1)}, \dots, \Phi_{a(j)}, \dots\}$ is known to be nonoptimal.
- Step 6: Compute $D = D_{LB}(a(1), \dots, a(j), \dots)$. If D is less than $CMIN$, go to step 9.
- Step 7: $j = j - 1$. If $j \leq 0$, terminate the procedure.
- Step 8: $a(j) = a(j) + 1$. If $a(j)$ exceeds 7, the number of available link capacity values, go to step 7. Otherwise, go to step 3.
- Step 9: Using expression (5.4) compute $T = T_{LB}(a(1), \dots, a(j), \dots)$. If $T \leq T_{\max}$, go to step 12.

- Step 10: If $a(j) = 7$, the number of available link capacity values, go to step 7.
- Step 11: Is link j a central link? If not, go to step 7. Otherwise set $a(j) = a(j) + 1$ and go to step 3.¹
- Step 12: If j is equal to the number of links in the network, go to step 14.
- Step 13: Set $a(j+1) = 1$; $j = j + 1$. Go to step 3.
- Step 14: Using the algorithm described in Section 3.4, compute the average response time for the current link capacity assignment. If it exceeds T_{\max} , go to step 10. Otherwise, set $C_{\min} = D$. Save the current values of the variables $a(1) \dots a(7)$ by setting $ISAV(i) = a(i)$ for $i = 1, \dots, 7$. $ISAV$ is simply a vector used for storing intermediate results. Set $j = j - 1$ and go to step 8.

The cost of the optimal solution is the value of C_{\min} following termination. $ISAV(i)$ contains the subscript of the component from the set Φ which should be used for the link capacity value in the i -th link to obtain the minimum cost feasible capacity assignment.

5.3.1.3 Results of the Example

We will now consider the case where $T_{\max} = .25$ and the $\{v_{i,h}^{\wedge}\}$ solution is given as follows: $v_{1,h}^{\wedge} = v_{2,h}^{\wedge} = v_{3,h}^{\wedge} = v_{4,h}^{\wedge} = 1350$ b/s;

¹ Unless link j is a central link, increasing $\Phi_{a(j)}$ will not affect T_{LB} as obtained from expression (5.4). Hence there is no need to consider further increasing the capacity value for link j if it is not incident to the central node.

$$v_{5,h}^{\wedge} = v_{6,h}^{\wedge} = v_{7,h}^{\wedge} = 450 \text{ b/s.}$$

The traffics and distances used are the same as those considered in the case study analysis of this network in Section 7.3. For these capacity assignments, CMIN initially has the value \$4200.

The above algorithm was implemented on the IBM 360/67 computer at the University of Michigan. The optimal assignment of link capacities is shown in Table 5-A. The monthly cost of this optimally assigned network is \$4102, which represents approximately a 2.5% improvement in the cost over the \$4200 figure for the $\{v_{i,h}^{\wedge}\}$ approximation.

Link	Optimal Capacity Value
1	1800 b/s
2	900 b/s
3	1350 b/s
4	1350 b/s
5	450 b/s
6	450 b/s
7	450 b/s

Table 5-A

The Optimal Link Capacity Assignment
for the Network of Figure 5.3
When $T_{\max} = .25$

It is interesting to note that in this example, approximately 25 seconds of CPU time were used to obtain the $\{v_{i,h}^{\wedge}\}$ solution. An additional 25 seconds of CPU time were used to find the optimum

capacity assignments shown in Table 5-A. It is further interesting to note that in obtaining the optimal assignment from the $\{\hat{v}_{i,h}\}$ approximation, step 14 was executed only 217 times. Hence it was necessary to actually perform the average response time calculations for only 217 assignment combinations in making this final improvement to the $\{\hat{v}_{i,h}\}$. To see how the initial knowledge of the $\{\hat{v}_{i,h}\}$ solution compared with an arbitrary starting solution as far as computational requirements for the branch and bound solution are concerned, backtracking was also used for an arbitrary starting feasible assignment in which all link capacities were set to 1800 b/s. This time, the optimal assignment was found in about 60 sec of CPU time; 552 executions of step 14 were required.

In conclusion, the apriori knowledge of the $\{\hat{v}_{i,h}\}$ solution reduced the computation time by slightly more than a factor of two in this particular case. Reductions of this magnitude may not appear to be appreciable in this case. However, in larger networks they can be vastly more significant, as will be seen in additional examples which follow. In the next section, we will discuss the computational requirements of this optimization technique in a more general context.

5.3.2 Computational Requirements and Summarization

In discussing computational limitations of backtracking and similar optimization techniques, we have seen that one important fact stands out above all others. Namely, the efficiency of these procedures depends almost solely on the application - specifically, on the size of

the network, the tightness of the constraints, the structure of the network, and on how close the cost of the $\{v_i, \hat{h}\}$ solution is to the cost of the optimal assignment which is to be found. It should be quite clear that there exist a wide variety of problems which cannot be solved in a reasonable length of time using this procedure. Knowledge of the $\{v_i, \hat{h}\}$ solution helps reduce the computational requirements in large problems, but sometimes this reduction is not even sufficient to render the problem solvable. We conclude these discussions of backtrack procedures by classifying the spectrum of problems according to their combinatorial size and expected computational requirements.

In the problem at hand, it has been pointed out that the number of possible capacity assignments is N^n in a network having n links when N values of capacity are available. If $N = 7$, then we have approximately 10^5 assignments in a network with 6 links, about 10^{10} assignments in a network with 12 links, about 10^{15} assignments in a network with 18 links, etc.

By classifying the problems according to their computational requirements, we can estimate the likelihood that branch and bound optimization techniques will obtain solutions in reasonable amounts of time. Based on the experiences of this research, for the seven capacity value cases one can classify as "small" those problems having approximately 10 or fewer links. By small, we mean to imply that backtrack procedures are highly likely to be capable of obtaining solutions for problems of this size, regardless of the starting capacity

assignment, constraint values, or other factors.

Similarly, we would classify as "medium" those problems having somewhere between 12 and 15 links and reasonable constraint values similar to those considered in the examples of Chapter 7. Experience with examples showed that solvability depends on how many of the $\{\hat{v}_{i,h}\}$ components are greater than Φ_1 and also on how much greater than Φ_1 these values are. Loose constraint values, say for $T_{\max} > 3.0$, usually resulted in all components of the $\{\hat{v}_{i,h}\}$ solution having capacity values close to Φ_1 . Thus we would place in this category those networks which are probably solvable if one has a good starting solution (all $\{\hat{v}_{i,h}\}$ components close to Φ_1) and is willing to use significant amounts of computer time.

In "large" networks having more than, say 15-20 links, the chances for being able to obtain an optimal solution using backtracking appear to depend almost totally on having good starting solutions, a moderately loose constraint value, and/or some structural characteristic which can be exploited. For example, if a particular network has a topological structure which frequently results in the value of the response time bound also being the actual response time, then the efficiency of the algorithm will be notably improved. Since the network structure also determines how sensitive the bounds are to changes in the link capacities, the efficiency of the algorithm again is seen to depend on the structure of the network under consideration. To substantiate this point, in the nineteen link network for the case study of Section 7.4,

an attempt was made to improve upon the $\{v_{i,h}^{\wedge}\}$ solution when $T_{\max} = .75$. In 5 minutes of backtracking on the 360/67, no solution better than the $\{v_{i,h}^{\wedge}\}$ was found. Due to finite computational resources, no attempts were made to go beyond this point.

On the other hand, in the case study of Section 7.5 a network having twenty-two links is considered. The $\{v_{i,h}^{\wedge}\}$ assignment for a particular structure when $T_{\max} = 2.0$ was improved to the optimal solution using only 13 seconds of backtracking. The cost differential between the $\{v_{i,h}^{\wedge}\}$ and the optimal assignments was approximately 15% a savings well worth obtaining in 13 additional seconds of CPU time!

To summarize, it appears that in those networks which we have classified as "large", it is not, in general, practical to use these optimization procedures except in a restricted way. They can be programmed and allowed to run for a given length of time. If the optimum is found in that acceptable length of time, the problem is solved. If not, possibly an improved solution which is not known for certain to be the optimum may be found. Otherwise, one must accept the $\{v_{i,h}^{\wedge}\}$ solution or one obtained in another manner.

In Chapter 6, it will be shown that the computational requirements for determining the $\{v_{i,h}^{\wedge}\}$ solutions are vastly superior to those associated with this backtracking optimization technique. The growth of these requirements is of the order of n^3 where n is the number of terminals in the network. This type of algorithm growth is considered

to be relatively good for combinatorial problems.¹

In the efforts of the next chapter where structure is treated as a design variable, we will be confronted with the task of solving this same optimization problem many times over for different structures which are considered. The readily obtainable $\{v_{i,h}^{\wedge}\}$ approximate solutions having these desirable algorithmic growth properties are used in describing the basic algorithm to assure that the design procedure can be used in networks with more than 20 terminals. Then a simple step can be added to the basic design procedure, enabling these approximate solutions to be upgraded to optimal solutions in those cases where the branch and bound algorithms can be meaningfully used.

In concluding Chapter 5, we comment briefly on the results of Section 5.2 and 5.3 in particular reference to the Esau-Williams and minimal spanning tree algorithms. It is not difficult to show why it is extremely important to treat link capacity as a design variable in order to obtain efficient allocations of the network resources.

5.4 Implications of Variable Capacity Values with Respect to Existing Design Procedures

As presented in Chapter 4, the Esau-Williams and minimum spanning tree algorithms are both concerned with minimizing line costs when a single capacity value is available for a particular link. Even though some method of concentration must often be used, neither of these procedures considers either the costs or performance of

¹ See for example, Reference [82].

concentrators in obtaining a solution. Hence for certain ranges of T_{\max} , the maximum acceptable average response time, these solutions may not even be feasible.

Even if these solutions should happen to be feasible, however, the single link capacity restriction does not enable these procedures to balance link capacity values with traffics. Hence it is highly likely that the methods described in Sections 5.2 and 5.3 can be used to effect significant reductions in line costs. Further evidence of this improveability potential comes from observing that the Esau-Williams and minimum spanning tree procedures consider but a single capacity assignment from the space of N^n possible assignment combinations which can exist.

Thus there can be little doubt now about why the solutions obtained by these design procedures are usually relatively poor in the cost sense if multiple link capacity values are available. In Chapter 6, we turn our efforts in the direction of solving the most general formulation of the design problem where structure, link capacity, and type of concentrator are treated as design variables.

Chapter VI

NETWORK DESIGN PROCEDURES IN THE GENERAL CASE

6.1 Introduction

In Chapter 6, the efforts of the previous chapters are integrated into a design model in which network topology, type of data concentrator, and link capacity value are all treated as variables. In addition, we consider an average response time constraint which is specified in terms of a maximum value which can be tolerated.

The fundamental approach to be taken consists of extending the Esau-Williams design procedure so that all of the critical design variables are considered. The Esau-Williams algorithm, in general, will obtain a suboptimal solution to the problem of finding the topological structure which minimizes line costs. As has been noted previously, only one speed of line may be considered and no consideration is given to concentrator costs or network performance except to prohibit the formation of structure having lines loaded in excess of some maximum acceptable value.

It will be shown in Chapter 6 that by removing these seemingly unnecessary limitations a suboptimal design technique can be developed which represents a significant improvement over the Esau-Williams procedure. It is an improvement in a twofold sense - a greater degree of reality is built into the model and it will consistently obtain better solutions to design problems by making more efficient utilization of the system resources. As was originally pointed out in Chapter 2, the

important macroscopic level design variables of interest in this work are:

- Topological structure
- Link capacity values
- Response time or performance of the network
- System reliability
- Control procedures
- Types of data concentrator
- System cost

A design approach is proposed which enables these parameters to all be considered at some meaningful level of reality.

The basic philosophy has centered on the development of a multi-dimensional design parameter space in which the dependent variables (total system cost and average response time) behave predictably as the independent variables (network topology, link capacity values, types of data concentrators) are allowed to vary in a prescribed, restricted fashion. The design is carried out for a particular assumed level of channel redundancy and for network control procedures consistent with the assumptions of Section 2.3; reliability and control procedures are considered to this extent.

Important tradeoffs in the overall network design process are systematically and efficiently assessed by exploiting the monotonic behavior of key design variables in this multidimensional design parameter space which is described in Section 6.3. The individual points in this

space correspond to network configurations whose performance and cost can be readily determined, whenever desired, using the previously developed models. Knowledge of the constraints that average response time may not exceed some maximum value and that the solution is desired to be minimum cost, coupled with this monotonic behavior of key variables in certain regions of the space enable the least costly solution in this restricted space to readily be determined. Since this restricted design parameter space does not contain all possible parameter combinations for the overall problem, the solution configuration will thus be only a "reasonable suboptimal" solution to the overall design problem (as was discussed in Section 1.6).

Due to the multiplicity of tradeoff relationships which must be considered in the solution of the general network design problem, the model will be most readily comprehensible if presented in two stages. In the first stage we will extend the Esau-Williams equal link capacity procedure to treat the type of data concentrator as a design variable, and to consider a constraint which demands that the average response time of the network not exceed T_{\max} , a given acceptable value. In the second stage, the first design model will be upgraded to consider the most realistic design situation in which a number of link capacity values are available to the designer.

We commence the efforts of this section by presenting an extension to the Esau-Williams procedure in which all key parameters are treated as variables with the exception of link capacity.

6.2 Network Design When Only One Link Capacity Value is Available

In this section, we propose an extension of the Esau-Williams procedure to solve the restrictive class of design problems where only topological structure and the type of data concentrator are treated as variables. We impose the constraint that the average response time may not exceed a given maximum acceptable value, T_{\max} .

At this juncture, we consider the situation where only one link capacity value is available; in Section 6.3 of this chapter this restriction will be removed from the problem formulation. The solutions obtained by the proposed procedures will be suboptimal, in general, as are the Esau-Williams solutions. However, it will be shown that the design extension of this section paves the way for a variable link capacity network synthesis procedure whose solutions, albeit suboptimal, are consistently less costly by significant amounts than those obtained using existing design procedures. Furthermore, additional levels of reality have been considered in obtaining these improved solutions.

Before proceeding, it is meaningful to summarize several of the noteworthy characteristics of the procedure to be presented. Since the proposed procedure is a straightforward extension of the Esau Williams algorithm, those characteristics also pertain to the basic Esau Williams procedure as well.

- The most obvious (cost-saving) improvements to any network with many point-to-point segments consist of multipointing those terminals which are clustered farthest from the central facility. The procedure assures that the most costly central

links are discarded first, if it is possible to preserve connectedness by adding significantly cheaper non-central links. Once it has been ascertained that significant savings can be realized by deleting a certain central link, the procedure permanently discards from further consideration any network structure which contains the given central link. The number of long links is thus kept relatively small. Martin [44] points out that this feature is to be desired in network design algorithms, in general. Nonetheless, this permanent rejection of all structures having one of the discarded links as a central link can lead to nonoptimum structures in special cases, as will be seen in certain examples of Chapter 7.

- At each iteration, the procedure rejects a large number¹ of non-optimal configurations. Only the maximum cost savings structural alteration is made at each step; all others are permanently rejected.
- If this procedure is used in a special manner which does not consider the costs of data concentrators and the average response time, it is equivalent to the basic Esau Williams algorithm. Esau and Williams report in reference [53] that the use of their procedure has always produced solutions for practical examples that could not be improved upon manually.
- The approach has favorable combinatorial ramifications (discussed in Section 6.2.3) which enable it to be computationally effective in networks having large numbers of remote terminals.

¹ A relationship for the number of structures considered at each step is developed in Section 6.2.3.

- The philosophy of the algorithm is to be greedy at each iteration with respect to the overall objective of obtaining the least costly network. By being greedy at each iteration we mean that the largest possible step in the direction of reduced total system cost is taken. This action forces the aforementioned desirable algorithm growth properties at the expense of a theoretically optimal solution.

The basic approach to be taken in solving the problem of this section will be to again generate a sequence of topological structures (Figure 6.1) in a greedy manner like the one used by Esau and Williams, but now considering only such alternatives that would result in an acceptable network response time. It will be shown that significant improvements in network cost and/or performance can sometimes be achieved by treating the type of data concentrator as a variable in the design procedure. We commence these efforts with a discussion of how the type of data concentrator variation will be effected.

6.2.1 How the Data Concentrator Type is Varied

Since it is desired to obtain the least costly configuration considered which has an acceptable average response time and since MSC concentrators are significantly more costly than the MX types, it is logical to use MX concentrators exclusively in the structures of Figure 6.1 if it is possible to satisfy the response time constraint. As the successive structures are generated, using MX's exclusively, it may happen that

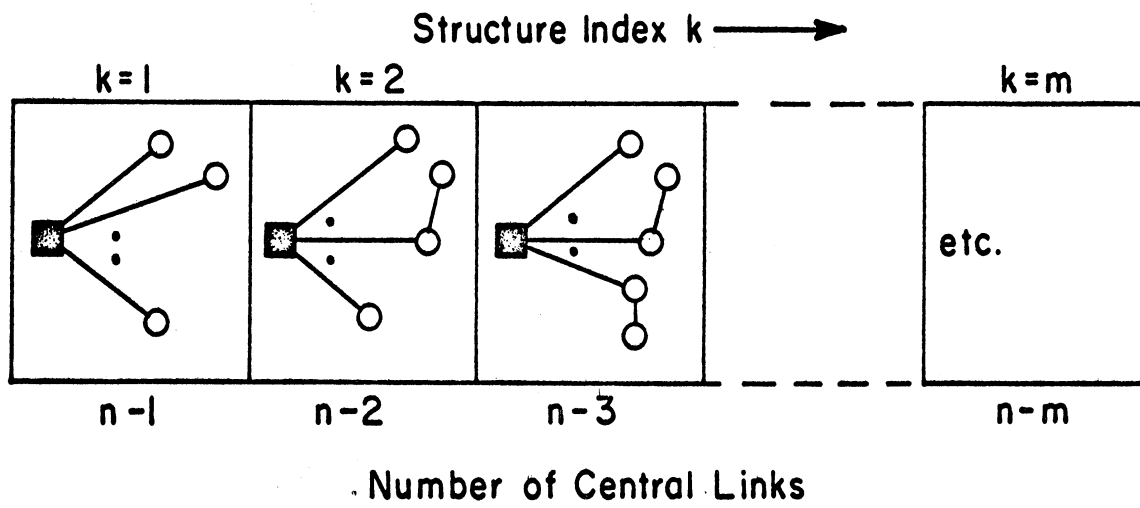


Figure 6.1. Variation of Topological Structure

at a particular step no further modifications can be made which would have acceptable response time properties. If only MX concentrators can be used, there is no way to realize possible additional line savings associated with certain alternatives which fail to satisfy the response time constraint but are otherwise acceptable. Recall that in Section 3.7 it was shown that in many instances an MX concentrator may be replaced by a MSC with a resulting significant reduction of average response time. Thus it may be possible to replace a certain MX with a MSC in the last configuration which used MX's exclusively and, in so doing, enable additional structures having acceptable average response times to be generated by further iterations of the procedure. Since each of these structures would have successively lower line costs, the cumulative line savings resulting from these additional iterations can offset the cost of upgrading the MX to a MSC. Hence one or more network configurations can be generated which use a MSC and have lower total system cost than the least costly all-MX configuration. In general, it is desired to use the minimum number of MSC's possible in these replacement operations so that the cost of upgrading can be offset with minimal cumulative line savings at ensuing steps.

Thus far we have not considered the question of where to place a MSC in a configuration, given that satisfaction of the performance objective demands that one or more additional MSC's must be used. Since the cost functions of Figures 4.2 and 4.4 will be used throughout the balance of this work, we note that upgrading any particular concen-

trator node from a MX to a MSC will always involve one of two incremental expenditures (\$900 or \$1800 per month, to be exact).¹ The particular values of these increments are not significant, only the fact that they depend solely on the equivalent number of terminal devices being concentrated.

At some step in the procedure, further savings may still exist to be realized, but no structural modifications not involving upgrades can be made which yield a new configuration having an acceptable average response time. We can refer to this situation as being one where no remaining valid alternatives can lead to an acceptable new configuration; the concept of valid alternative is introduced to describe a network modification which could lead to a less costly acceptable network configuration at some future step. We will precisely define what is meant by a valid alternative in Section 6.2.2.1.1. However for the present it will suffice for the reader to think of it as a structural modification not involving any upgrades that may be characterized by a central link deletion and a non-central link addition operation which, if implemented, would lead to a connected network having:

1. no overloaded links and
2. either lower line costs or lower total system cost

than in the unmodified configuration.²

¹ We assume that no node requires more than two concentrators to handle its total remote terminal device load.

² A detailed discussion of these conditions will be presented in Section 6.2.2.1.1 where valid alternatives are concisely characterized.

In order to ascertain that one or more upgrades must be made, each valid alternative is individually checked to determine whether it would produce an acceptable configuration without upgrades of concentrators. An upgrade must be made when all such attempts fail. In considering all of the valid alternatives, the net savings (line savings minus incremental concentrator costs) associated with each such alternative is calculated. One of these alternatives has maximum net savings, ignoring the fact that it would not result in an acceptable average response time. Given that at least one MX concentrator must be upgraded, it is therefore reasonable to choose a MX concentrator node which renders the average response time acceptable for the configuration indicated by this maximum savings valid alternative, if possible. By so doing the net cost (cost of the upgrade less net savings realized from implementing this maximum savings valid alternative) of forming the next structure in the sequence of Figure 6.1 is minimized over all valid alternatives. All other alternatives which are definitely non-optimal are thereby rejected.

If there are several such nodes which can be upgraded and render the response time acceptable for this maximum savings valid alternative, clearly the node should be selected which yields the smallest value of average response time in the configuration having one fewer central links. In so doing, we choose the feasible configuration whose average response time is farthest from the constraint value, T_{\max} . This selection is the configuration least likely to lead to a requirement for

additional costly MSC's at ensuing steps.

On the other hand, if there are no MX concentrator nodes which, if upgraded, could render the response time acceptable for this maximum savings valid alternative, then a similar attempt should be made to make the response time acceptable for the alternative having the next largest net savings. If this valid alternative can be made acceptable by upgrading a single MX, then the next configuration is formed by implementing the structural modification which this particular alternative suggests. Otherwise, the alternative having the next largest net savings should be considered, and so forth. With this strategy, we thus select the alternative that leads to the minimum differential cost in forming the next structure in the sequence of Figure 6.1. All other definitely non optimal (in a cost sense) valid alternatives are thereby ruled out.

At this point, the reader may be wondering why cost is consistently given priority over response time in determining where to place the MSC's in the network. This question is best answered by a simple restatement of our previously announced intention to obtain the least costly network which has an acceptable average response time by being greedy and taking the largest possible step in the direction of minimum total system cost at each iteration. Of course, in so doing, we accept only a reasonable suboptimal solution in exchange for desirable combinatorial growth properties which will be discussed in Section 6.2.3.

It may happen that at some point in the procedure a single MX to MSC upgrade will not make the response time acceptable for any valid alternatives. At such a point an attempt is made to consider those upgrades of two MX's which might make one of the valid alternatives acceptable. These double upgrades can take place in two ways, either by performing an upgrade at one node whose concentration requirements require two multiplexors or by performing MX to MSC upgrades at each of two different concentrator nodes which individually require only a single multiplexor.

Knowing that at least two upgrades must take place, we again attempt to minimize the differential cost of forming the next structure in the sequence of Figure 6.1. Precisely the same strategy is used as for the single upgrade situation - namely that of attempting to make the average response time acceptable for the valid alternative which would form the next structure with the least differential cost. All that remains is to indicate how the node(s) for the pairwise upgrade should be selected.

First, we form a temporary configuration by modifying the most recently defined structure of Figure 6.1 in accordance with the alternative which would lead to maximum net savings (line savings minus incremental concentrator costs, assuming no concentrator types are changed) and no overloaded links. Then for this temporary configuration we define the following parameters:

- T_{TEMP} \triangleq The average response time of this temporary configuration, obtained using the computational procedure of Section 3.4.
- u' \triangleq The particular node of all those in the temporary configuration which requires two multiplexors that would, if upgraded to two MSC's result in the maximum reduction in average response time. Denote the value of this maximum reduction by $\Delta T(u')$; it is obtained by using the computational procedure of Section 3.4. If no such u' exists, set $\Delta T(u') = 0$.
- u'' \triangleq The particular MX node of all those in the temporary configuration requiring a single concentrator that would, if upgraded to a MSC, result in the maximum reduction in average response time. Denote the value of this maximum reduction by $\Delta T(u'')$; it is obtained by using the computational procedure of Section 3.4. If no such u'' exists, set $\Delta T(u'') = 0$. Now upgrade, if possible, node u'' in the temporary configuration and let

u''' \triangleq The particular MX node of all those in this modified temporary configuration which require a single multiplexor that would, if upgraded to a MSC, result in the maximum reduction in average response time. Denote the value of this maximum reduction in average response time by $\Delta T(u''')$; it is again obtained by using the computational procedure of Section 3.4. If no such u''' exists, set $\Delta T(u''') = 0$.

Now if

$T_{\text{TEMP}} - \max \{ \Delta T(u'), \Delta T(u'') + \Delta T(u''') \} \leq T_{\text{max}}$ then we know that at least one double upgrade exists which will make this valid alternative acceptable. Given this fact, then to minimize the likelihood of having to add additional MSC's at ensuing steps, u' should be upgraded if $\Delta T(u') \geq \Delta T(u'') + \Delta T(u''')$. Otherwise u'' and u''' should be upgraded. As long as the above inequality is satisfied, we have obtained the next structure. But what if it is not satisfied for the present configuration whose average response time is T_{TEMP} ?

As was done in the single upgrade case, we repeat this procedure, redefining the above four parameters for a new temporary structure.

This new temporary structure is formed by modifying the most recently defined network of Figure 6.1 in the manner suggested by the valid alternative which has the next largest net savings, assuming no concentrator types are changed. Valid alternatives having successively smaller net savings are considered in similar fashion until either a double upgrade can be made for one of them or no more valid alternatives exist.

If at some step, no valid alternative can be rendered acceptable by two MX-to-MSD upgrades, the procedure is terminated. The contingency for needing to upgrade three or more MX's at a single step has not been considered, since it is reasonable to assume that additional line savings could not offset the costs of these three or more relatively expensive upgrades. Usage of the procedure in practice revealed that single upgrades were always adequate. We conclude this section with a brief classification of these structures of Figure 6.1 in terms of the number of MSD's which each uses.

If a total of m structures are generated, then the first m_0 of these will use MX concentrators exclusively, that is exactly 0 MSD's are used. Since it is, by definition, not possible to get an acceptable performance in configuration $m_0 + 1$ using MX's exclusively, at least 1 MSD concentrator must be used to obtain an acceptable average response time. Let m_1 denote the total number of structures in the sequence starting with configuration $m_0 + 1$, for which the average response times are acceptable using exactly 1 MSD somewhere in the network. By extending this notation we may classify the m structures

of the sequence according to the number of MSC concentrators used.

Thus we have

$$m = m_0 + m_1 + m_2 + \dots + m_k \quad (6.1)$$

where m_i = the number of different network configurations having exactly i MSC concentrators.

At this juncture, we now turn to integrating all of these operations into an algorithm which precisely describes how the design is carried out. Any uncertainties which the reader may have developed during the above discussion should become resolved.

6.2.2 The Design Technique

Throughout the balance of this section it is presumed that we seek the least costly structure which satisfies the average response time constraint. Since only a subset of all possible structures can be considered, we will obtain, in general, a reasonable, suboptimal solution. The least costly configuration of those considered is easily preserved as the algorithm proceeds; explicit reference to this step of "remembering" this least costly configuration has not been made. We now define the parameters or variables of the design model.

6.2.2.1 Parameters of the Model

To facilitate an orderly presentation of the rather lengthy list of parameters for the model, this list has been partitioned into two sections. Parameters in the first group are those needed to precisely characterize the concept of alternatives and validity which have been used so

frequently. The balance of the parameters are then considered in the second group.

6.2.2.1.1 Parameters for Characterizing Structural Alternatives and Validity

- Let $T_{\max} \triangleq$ The maximum acceptable average response time.
- $\Delta L(X, YZ) \triangleq$ The line savings associated with removing central link X and adding non-central link YZ, where Y is assumed to be in the same segment as X and Z is not in the same segment as X.
- $\Delta C(X, YZ) \triangleq$ The incremental concentrator costs of deleting central link X and inserting non-central link YZ, if multiplexors are always used whenever a new concentrator node is produced by the change. Again, Y is assumed to be in the same segment as X, and Z is not in the same segment as X. If any concentrating nodes would require one more (or fewer) concentrator devices after the structural change than before, then $\Delta C(X, YZ)$ is defined to include the proper adjustments in cost necessary to reflect the addition (or deletion) of the appropriate type of concentrator device.
- Throughout the algorithm it is implicitly understood that a concentrator node requiring two concentrating devices will use two devices of the same kind (MX or MSC).

$\Delta D(X, YZ) \triangleq \Delta L(X, YZ) - \Delta C(X, YZ)$, the net cost savings

associated with deleting central link X and inserting non-central link YZ for some particular structure.

Let X be a variable which ranges over the central links of a particular structure, and

Y be a variable which ranges over the remote terminals which use central link X, and

Z be a variable which ranges over the remote terminals which do not use central link X.

Let the joint variations of these X, Y, and Z variables (which are effected in a manner to be described) generate three tuples or triples, each of which consists of a single value for X, Y, and Z. A certain triple denotes a structural modification which could be made in terms of a central link X which should be deleted and the corresponding non-central link YZ to be added to a given structure. We will also refer to these triples as alternatives.

As the X, Y, and Z variables are jointly varied over their entire ranges in the manner to be described, we are interested in considering only those alternatives that might ultimately produce a configuration having lower total system cost than any considered thus far. Such triples will be denoted as valid alternatives. We now shall give a precise definition of valid alternative and discuss the various conditions which comprise the definition of validity. A triple of X, Y, and Z values constitutes a valid alternative if each of the following conditions holds true:

1. $\Delta D(X, YZ) > 0$; if not $\Delta L(X, YZ) > 0$.
2. If the deletion addition operation indicated by X, YZ were performed, the total message traffic flow on the central link which Z shares would not exceed $\Phi_Z^* \rho_{\max}$, where Φ_Z^* is the capacity of the central link used by the remote terminal at node Z.¹
3. Deleting central link X and adding non central link YZ will preserve the connectedness of the network.²

All possible triples are generated, one at a time, in the manner to be described in the algorithm of Section 6.2.2.2. As each one is generated, it is checked for validity by testing for the conditions 1 - 3 above. If valid, it is preserved by entering it as the $j+1$ st component of a vector V_k which will be defined as a vector of all triples corresponding to valid alternatives. It is assumed that j is an index on the components of the vector V_k and that exactly j components have been defined up to the current point in the procedure. On the other hand, if the current triple is not valid, the next one is immediately generated and checked for validity, etc. We now briefly discuss the first two validity conditions before completing our development of the parameter list.

¹See expression (6.2).

²This represents no problem since by defining the allowable ranges of the X, Y, and Z variables as we have done, each triple must describe a modification which would preserve connectedness.

Discussion of Validity Condition #1

Observe that when $\Delta D(X, YZ) > 0$, additional net savings can be realized with this alternative so that further progress in the direction of reduced total system cost can be made. Hence such an alternative should not be ruled invalid.

If on the other hand, $\Delta D(X, YZ) < 0$, but $\Delta L(X, YZ) > 0$, the implication is that positive line savings exist but the incremental concentrator cost of the modification would exceed the value of the line savings. Even so, we enter it as a valid alternative since making this one modification could pave the way to offsetting positive savings modifications at future steps because some line savings alternatives still exist. If both quantities ΔD and ΔL are negative, however, it is not possible to obtain further cost savings modifications and the triple is not considered valid.

It does not suffice to replace both of these checks by a single test for $\Delta L(X, YZ) > 0$ since in a few isolated cases it can happen that $\Delta D(X, YZ) > 0$ with both $\Delta C(X, YZ) < 0$ and $\Delta L(X, YZ) < 0$. In these cases we clearly would not want to declare the present triple invalid, since further net savings modifications still exist.

Discussion of Validity Condition #2

Any single-step structural modification (one where a central link is deleted and a non-central link added so that network connectedness is preserved) would result in merging two segments of the network into a single segment. The central link on the merged segment therefore must carry the combined message traffic of both segments for which the

merger is being contemplated. Since a communication link is not allowed to be loaded in excess of ρ_{\max} , it may happen that some alterations would, if performed, overload the central link which remains after the merging. Thus any such alteration which would result in an overloading is an invalid alternative regardless of the associated cost savings. Stated mathematically, if the current triple indicates that segment X should be merged into the segment containing Z, then this alternative is not valid unless

$$\sum_{i \in S_X} \frac{\gamma_i}{\mu} + \sum_{i \in S_Z} \frac{\gamma_i}{\mu} \leq \Phi_Z^* \rho_{\max} \quad (6.2)$$

where S_X is the set of terminals which share central link X before the two links are multipointed and S_Z is the set of terminals which share the segment containing node Z before multipointing. Of course, Φ_Z^* is the capacity of the central link which would remain in the merged segment.

We now conclude our presentation of the model parameters by considering the remaining variables.

6.2.2.1.2 The Remaining Parameters¹

Let

k be an index on the structures generated by the procedure.

NST denote the total number of structures which are generated by the procedure.

¹ All references to the step numbers in these definitions refer to the algorithm presented immediately following the definitions.

- V_k be a vector of all valid alternative modifications for structure k , as defined at step 8.
- I_k be an index on the alternatives in the vector V_k .
- R_k be a vector of all alternative modifications in the vector V_k which are attempted by the procedure but fail (see step 16) because of unacceptable average response times.
- q be an index on the alternatives in the vector R_k .
- NQ denote the cardinality of the vector R_k .
- TRY be a boolean variable whose value will be set to unity (see step 16) as soon as any attempt has been made to generate structure k without adding any MSC's; the value of TRY remains zero, otherwise.
- u^* be a variable, introduced at step 19, which denotes the particular MX node of all those in the currently defined temporary configuration requiring a single concentrator that would, if upgraded, result in the maximum reduction in average response time
- $\Delta T(u^*)$ be the value of the reduction in average response time if node u^* is upgraded; it is obtained by using the computational algorithm of Section 3.4.¹

The variables u' , u'' , u''' , $\Delta T(u')$, $\Delta T(u'')$, $\Delta T(u''')$ which are introduced at step 22 are defined exactly as they were in the discussions

¹ Specifically, a computational module called GTRESP will be introduced shortly to describe the implementation of this algorithm from Section 3.4.

of Section 6.2.1; the temporary configuration referred to in the definitions of Section 6.2.1 is equivalenced to the temporary structure currently being considered as step 22 is executed.

Throughout the discussions of the procedure, the existence of a computational module called GTRESP is postulated. This module is an implementation of the algorithm presented in Section 3.4 for obtaining the average response time in a general network for which the types of data concentrators and link capacity values are known.

6.2.2.2 The Algorithm

- Step 1: Set $k = 0$; establish the completely point to point network as the first structure in the sequence to be generated.
- Step 2: $k = k + 1$; terminate the procedure if k exceeds the number of remote terminals in the network.
- Step 3: Initialize the variable TRY by setting it to zero.
- Step 4: Set j and $q = 0$. Define the range of the X variable for structure k . Initialize X.
- Step 5: Define the range of the Z variable from the nodes of structure k which do not share central link X. Initialize Z.
- Step 6: Define the range of the Y variable from the nodes of structure k which share central link X. Initialize Y.
- Step 7: Can the segment containing Z handle the traffic of segment X? (This test is made using expression (6.2)). If not, go to step 10.

- Step 8: Compute $\Delta D(X, YZ) = \Delta L(X, YZ) - \Delta C(X, YZ)$. If $\Delta D(X, YZ) > 0$ or if $\Delta L(X, YZ) > 0$ then increment j and note the current triple as being valid by entering it as the j th component of the vector V_k . This is done by setting $X_j = X$, $Y_j = Y$, and $Z_j = Z$ where X_j, Y_j and Z_j denote the components of the j th triple entered in V_k .
- Step 9: Have all the Y's been considered? If not, augment the Y variable by one and go to step 8.
- Step 10: Have all the Z's been considered? if not, augment the Z variable by one and go to step 7.
- Step 11: Have all the X's been considered? If not, augment the X variable by one and go to step 5. Otherwise, set $I_k = j$ where I_k is the number of valid alternatives for structure k .
- Step 12: If $I_k = 0$ and $TRY = 0$, go to step 25 since no more structural changes can be made which will lower the total system cost. If $I_k = 0$ and $TRY = 1$, go to step 17. Otherwise, search the components of the vector V_k to find the triple (denoted $\hat{X}_j, \hat{Y}_j, \hat{Z}_j$) which maximizes $\Delta D(X_j, Y_j, Z_j)$ over all triples in the vector V_k .
- Step 13: In structure k only the two segments which contain nodes \hat{X}_j and \hat{Z}_j would be affected if the indicated structural modification were performed. Treating the segment \hat{X}_j as an entire network, use the GTRESP module to obtain its average response time which we will denote as $T(\hat{X}_j)'$. Obtain $T(\hat{Z}_j)'$, the average

response time of the segment containing \hat{Z}_j , taken as an entire network, in an identical manner. Then treating the segment which would result if the segments containing \hat{X}_j and \hat{Z}_j were merged as an entire network,¹ use the GTRESP module to obtain its average response time, which we will denote $T(\hat{X}_j \cup \hat{Z}_j)$. If the indicated structural modification were performed, we can express the resulting change, ΔT , in average response time as:

$$\Delta T = \frac{T(\hat{X}_j \cup \hat{Z}_j) (\gamma_x + \gamma_z)}{\gamma} - \frac{T(\hat{X}_j) \gamma_x}{\gamma} - \frac{T(\hat{Z}_j) \gamma_z}{\gamma}$$

where γ_x and γ_z are the total arrival rates of messages to terminals in the segments containing \hat{X}_j and \hat{Z}_j , respectively.

Compute this ΔT .

Step 14: Is $T^{(k)} + \Delta T \leq T_{\max}$, where $T^{(k)}$ is the average response time for the k th structure? If not go to step 16.

Step 15: Form structure $k+1$ by deleting central link \hat{X}_j and adding non-central link $\hat{Y}_j \hat{Z}_j$ to structure k . Add an additional concentrator of the appropriate type at all concentration nodes which require this additional device due to the increased traffic resulting from the merger. Of course, if any concentrator nodes should require two devices in structure k but only one in structure $k+1$, the superfluous device(s) should be eliminated

¹ Whenever the merging of two segments requires the addition of a new concentrator, multiplexors are always used. Upgrades of MX's to MSC's are not considered until steps 18-23.

from the $k + 1$ st configuration. Set $T^{(k+1)} = T^{(k)} + \Delta T$ and go to step 2.

Step 16: Augment q . Eliminate from the vector V_k the triple whose components are $\hat{X}_j, \hat{Y}_j, \hat{Z}_j$ which gave rise to the unacceptable response time by entering it as the q th component of the vector R_k . This is accomplished by setting $X_q = \hat{X}_j, Y_q = \hat{Y}_j,$ and $Z_q = \hat{Z}_j$ where $X_q, Y_q,$ and Z_q denote the components of the q th triple entered in R_k . R_k is the vector of all initially valid alternative modifications that are attempted but fail because of unacceptable average response times. Decrement I_k by 1 and set $TRY = 1$. Go to step 12.

Step 17: Save the current value of q by setting $NQ = q$. Reset $q = 1$.

Step 18: Form a temporary configuration by deleting central link X_q and adding non-central link $Y_q Z_q$ to structure k , where the subscript q denotes the links for the q th alternative in the vector R_k . Multiplexors are always used at any new concentrator nodes which arise as a consequence of the modifications. Also, there may be certain concentrator nodes in structure $k + 1$ that require one more (or fewer) concentrator device of the same type than in structure k . At each node so affected, adjust the number of MX or MSC devices used to the minimum adequate level.

- Step 19: Using the module GTRESP, calculate the average response time for this configuration and let it be denoted by T_{TEMP}^1 . Determine u^* and $\Delta T(u^*)$ from their definitions in the parameter list of Section 6.2.2.2. If $T_{TEMP} - \Delta T(u^*) > T_{max}$, go to step 21.
- Step 20: Upgrade the concentrator(s) at node u^* from MX to MSC. Structure $k+1$ has now been defined by this upgrading operation to the temporary structure formed at the last iteration of step 18. Set $T^{(k+1)} = T_{TEMP} - \Delta T(u^*)$ and go to step 2.
- Step 21: Augment q by one. If q does not exceed NQ go to step 18. Otherwise, reset $q = 1$.
- Step 22: Form a temporary configuration by deleting central link X_q and adding non central link $Y_q Z_q$ to structure k^2 , where again the subscript q denotes the links for the q th triple in the vector R_k . Using the module GTRESP, calculate the average response time for this temporary configuration and

¹ At a previous execution of step 13 for structure k , this value was already calculated. It would have been possible to store all calculations from Step 13 in a table and avoid recalculating them again here by using a table look up to retrieve the desired average response time. However, to simplify the explanation of the algorithm, and since the recalculation is a straightforward, simple matter, we have elected to recalculate T_{TEMP} .

² Again, whenever the merging of two segments produces new concentrator nodes, multiplexors are always used. Concurrently, the number of concentration devices used (one or two) at each node in this temporary configuration is adjusted to the minimum necessary value (whenever any such adjustments are necessary).

denote it by T_{TEMP} . Determine u' , u'' , u''' , and $\Delta T(u')$, $\Delta T(u'')$, $\Delta T(u''')$ from their definitions of Section 6.2.1.

If $T_{TEMP} - \max \{ \Delta T(u'), \Delta T(u'') + \Delta T(u''') \} > T_{max}$, go to step 24.

Step 23: If $\Delta T(u') \geq \Delta T(u'') + \Delta T(u''')$, upgrade the pair of MX's at node u' in the temporary configuration. Otherwise upgrade the single MX's at nodes u'' and u''' . In either case, structure $k+1$ has been defined by this upgrading operation to the "temporary" structure. Set $T^{(k+1)} = T_{TEMP} - \max \{ \Delta T(u'), \Delta T(u'') + \Delta T(u''') \}$ and go to step 2.

Step 24: Augment q by 1. If q does not exceed NQ , go to step 22.

Step 25: Set $NST = k$ and terminate the procedure.

Following termination, the value of NST indicate the number of structures which have been generated.¹

It is of interest to summarize the characteristics of the upgrading operations in steps 18-23 and reiterate the desirable features of the approach being used. These discussions pertain, of course, only to those adjacent structures in the sequence of Figure 6.1 which contain different numbers of MSC's.

- Letting $\Delta D(\hat{X}_j, \hat{Y}_j, \hat{Z}_j)$ be the net savings associated with forming structure $k+1$ from structure k without upgrading any concen-

¹ The algorithm herein described does not continue if at some iteration, three or more MX - to - MSC upgrades would be necessary to make acceptable some alternative for structure k . It is reasonable to assume that additional line savings could not offset the costs of three or more relatively expensive MSC's which would have to be added at this single iteration.

trators, we can express the net cost differential between structures k and $k+1$ as: $\$900 - \Delta D(\hat{X}_j, \hat{Y}_j, \hat{Z}_j)$ if one upgrade is made, or $\$1800 - \Delta D(X_j, Y_j, Z_j)$ if two upgrades are made. Thus in choosing the $\hat{X}_j, \hat{Y}_j, \hat{Z}_j$ alternatives such that $\Delta D(X_j, Y_j, Z_j)$ is maximized, the procedure selects the valid modification which gives the minimum net cost differential between structures k and $k+1$, and all other non-optimal alternatives are rejected.

- If any of several MX concentrator nodes could be upgraded for the same minimum differential cost, that alternative is selected which results in the lowest average response time for structure $k+1$.

We conclude by performing a combinatorial analysis which shows the algorithm to have a growth property less than n^3 where n is the number of nodes in the network. This observation is shown to have favorable connotations as far as computational requirements are concerned.

6.2.3 Number of Structures Considered by the Procedure; Combinatorial Implications

In this section we will precisely determine the algorithmic growth properties of the proposed procedure.¹ Justification for the procedure is not our primary concern in this section; rather we seek only to objectively discuss the strategies from a combinatorial standpoint.

¹ This analysis is also pertinent for the Esau Williams algorithm; the authors, however, do not explicitly consider this issue in reference [53].

The procedure always starts with a configuration of $n-1$ central links where n is the total number of nodes in the network. At each iteration, a central link is deleted and some non-central link is added; hence the maximum number of iterations to completion is $n-2$, and at each iterate the maximum net savings alternative is chosen. While $n-1$ different configurations may actually be generated by the procedure (including the star configuration), a great deal more topologies are considered, but discarded, since some other structural configuration having a lower cost and exactly the same number of central links can be chosen.

Concerning this number which are considered by the algorithm, following general analysis of the procedure may be made:

In obtaining structure #2 of Figure 6.1:

The procedure considers all point labeled spanning trees having $n-2$ central links and selects the one having minimum total cost.

In obtaining structure #3 of Figure 6.1:

Say central link k_1 was deleted in obtaining structure #2 and that deleting link k_2 gives maximum savings at the current point in the procedure. The algorithm permanently rejects any structure which contains k_1 or k_2 or both as central links. These central links are the long central links which were deleted at prior iterations; their permanent rejection is usually a very sensible step to take to reduce the combinatorial dimensions of the problem.

Nonetheless it can lead to suboptimal solutions in certain situations as will be seen in Chapter 7.

In obtaining structure #j+1 of Figure 6.1:

Algorithm has up to this point eliminated all point labeled trees having central links k_1, k_2, \dots, k_{j-1} or any combination thereof. At this iteration all point labeled trees having $n - j - 1$ central links, of which one is the k_j -th link, are eliminated. The cumulative elimination including that for the current iteration therefore, consists of any topologies having link k_1 or k_2 or $\dots k_j$ as a central link. These are again the long central links which have been deleted at prior iterations and resulted in the maximum savings. By this greedy permanent rejection, we gain needed computational efficiencies in the form of desirable algorithm growth properties. But as has been previously noted, we sacrifice the optimal character of the solution.

Also of interest is the maximum number of valid alternatives which might exist at each iteration in the procedure of Section 6.2. This number establishes an upper bound on the number of computations which must be performed at each iteration. The computational requirements of the procedure can be directly related to a summation of this value for each of the $n-2$ iterations in the procedure.

Let $M_j \triangleq$ Maximum number of valid alternatives which might be considered at the j th iteration of the procedure.

- $M \triangleq \sum_{j=1}^{n-2} M_j$, the total number of valid alternatives for all iterations.
- $(X_1, X_2, X_3, \dots, X_{n-j}) =$ the set of all nodes which are connected directly to the central node, just prior to beginning the j th iteration.
- $N_j(X_i) =$ The number of remote-terminal nodes which comprise the multipoint segment having X_i as a central link just prior to beginning the j th iteration.

A method for relating M to the total number of nodes, n , in a network would be useful in describing the growth characteristics of the procedure. Since any node in the segment having central link X_i could be connected to any other remote terminal node not in the same segment, the maximum number of such alternatives that exist if central link X_i is to be deleted is given by $N_j(X_i) [n - N_j(X_i) - 1]$. The term $[n - N_j(X_i) - 1]$ denotes the number of remote terminal nodes not in the segment having central link X_i .

Since any of the central links could possibly be deleted at iteration j , the maximum total number, M_j , is given by a summation over the values for the individual central links. Thus we have

$$M_j = \sum_{i=1}^{n-j} N_j(X_i) [n - N_j(X_i) - 1]$$

$$M_j = n(n-1) - \sum_{i=1}^{n-j} N_j(X_i)^2 - (n-1)$$

$$M_j = (n-1)^2 - \sum_{i=1}^{n-j} [N_j (X_i)^2]$$

To obtain the maximum total number of alternatives, M , we have

$$\begin{aligned} M &= \sum_{j=1}^{n-2} \left[(n-1)^2 - \sum_{i=1}^{n-j} [N_j (X_i)^2] \right] \\ &= (n-2)(n-1)^2 - \sum_{j=1}^{n-2} \left[\sum_{i=1}^{n-j} [N_j (X_i)^2] \right] \end{aligned}$$

This number M represents the maximum number of unique topological configurations which are considered by the algorithm in selecting the last $n-2$ configurations which comprises the sequence of structures generated by the iterative procedure. $M+1$ is therefore the total number considered.

Since n^{n-2} is the number of different possible topological tree structures (which can connect a network of n labeled nodes) the algorithm considers a fraction given by

$$\text{Max \%} = \frac{M+1}{n^{n-2}} \times 100 = \frac{(n-2)(n-1)^2 - \sum_{j=1}^{n-2} \sum_{i=1}^{n-j} N_j (X_i)^2 + 1}{n^{n-2}} \times 100$$

Since $M+1$ is always less than n^3 , $\text{Max \%} < 100 n^3 / n^{n-2}$. When $n \leq 5$, this is not a good upper bound since 100% is better. However, for the cases usually of most interest where $n > 5$, $\text{Max \%} < 100n^{5-n}$.

It is generally acknowledged that algebraic growth where the computational requirements are of the order of n^k is acceptable when $k \leq 3$, where k is a constant independent of the size of the network.¹ Clearly, since $M+1 < n^3$ one can conclude that the growth characteristics of this procedure are not likely to be excessive as long as n^3 computations could be performed on a computer within a reasonable length of time.

All the efforts throughout Section 6.2 have been confined to the case where a single link capacity value is available. At this juncture we turn to incorporating the link capacity variable into the design problem.

6.3 The Approach to be Taken in the General Network Design Problem

At this point, we remove the single link capacity restriction from the network design problem and consider the situation where all of the key parameters - topological structure, type of data concentrator, and link capacity - are treated as variables. The fundamental approach to be taken will involve the generation of a multidimensional design parameter space from which the minimum cost feasible configuration in the space is obtained using an efficient search procedure. The solutions obtained by this method will, in general, be suboptimal, but they are shown to be significantly better than those of existing design procedures and also more realistic in the sense that all of the important design variables have been considered to some meaningful extent.

¹ See, for example, Reference [82].

One can interpret the strategy of the single link capacity design procedure of Section 6.2 as the selection of a minimal cost network from a set of configurations which comprise a multidimensional design parameter space. In this space shown in Figure 6.1, for example, as topological structure is varied, the number of central links per configuration decreases from $n-1$ down to possibly as few as 1 for the $n-1$ remote terminal situation. As topological structure is varied in Figure 6.1, the line costs per configuration usually range from a maximum value in the point to point network down to a relatively small value in the last structure of the sequence. Similarly, from expression (6.1), we see that, as the topological structure is varied, the number of MSC concentrators used per configuration ranges from 0 in the point to point network up to a maximum value in the network having the fewest central links. We now seek to add an additional variable to this design parameter space - namely, link capacity.

In Chapter 5, a considerable amount of attention was focused on the need for being able to use varying line speeds in different links of a network. Further motivation for designing networks which have different link capacity values comes from the economies of scale which exist in the cost-capacity function. A higher speed shared link has a smaller per unit transmission cost than multiple low speed links having an equivalent total transmission capacity. This fact provides a strong incentive to do as much sharing of lines as the response time constraint will permit. However, a network containing these high capacity links

is not the most efficient one if it uses more capacity in any link than the minimum necessary to satisfy the performance constraint. Thus the resolution of this delicate balance between average response time and network cost must be accomplished by a method which incorporates this tradeoff between the use of high and low speed links.

It is proposed to resolve this tradeoff by using the design procedure of Section 6.2 for each discrete capacity value available. Thus there exists a spectrum of values for the link capacity used in the procedure. At one extreme is the case where the configurations of Figure 6.1 are created using only the smallest capacity value for every link of every structure; the total amount of excess or unused capacity in each configuration of the sequence is continually kept at a minimum. At the other extreme, each configuration is designed using only the maximum capacity value for each link in every structure. By so doing the cost per unit of capacity is lower in every link. However, the total amount of excess or unused capacity in each structure of Figure 6.1 will usually be relatively large in this case. Usage of an intermediate link capacity value in forming each of the structures of Figure 6.1 will, in turn, lead to network configurations which have intermediate amounts of excess or unused capacity. We have proposed a design technique which will enable these tradeoffs to be systematically resolved by embedding both extremal strategies as well as intermediate approaches into the design procedure.

It should be evident that the decision making rules in the procedure of Section 6.2 depend on the particular link capacity value under consideration. This is to say that the sequence of topological structures generated when all links have capacity value Φ_x will, in general, not be identical to the sequence of structures generated when all links have capacity Φ_y , for $x \neq y$. Thus if any of N different values in the set $\Phi = \{ \Phi_1, \Phi_2, \dots, \Phi_N \}$ can be used, then N topological sequences would be generated if the procedure is used for all of the N values.

Furthermore, in Chapter 5 we discussed detailed procedures for efficiently utilizing the capacity resources in a network whose structure and type of concentrators are known. Thus link capacity variations can be effected in the following twofold manner

- 1) By varying the particular capacity value which is used for the previously-described single-capacity network design algorithm. Thus as many sequences of structures are considered as there are link capacity values from which to choose.
- 2) By successively reducing the unneeded excess capacity in the links of a particular structure of one of the sequences in the manner described in Section 5.3.

This twofold approach is appealing in the sense that the design parameter space includes the different topological sequences which result from applying the single-capacity design algorithm for different capacity values. The tradeoffs arising from the economies of scale in the cost-capacity function are resolved in this way. As the link

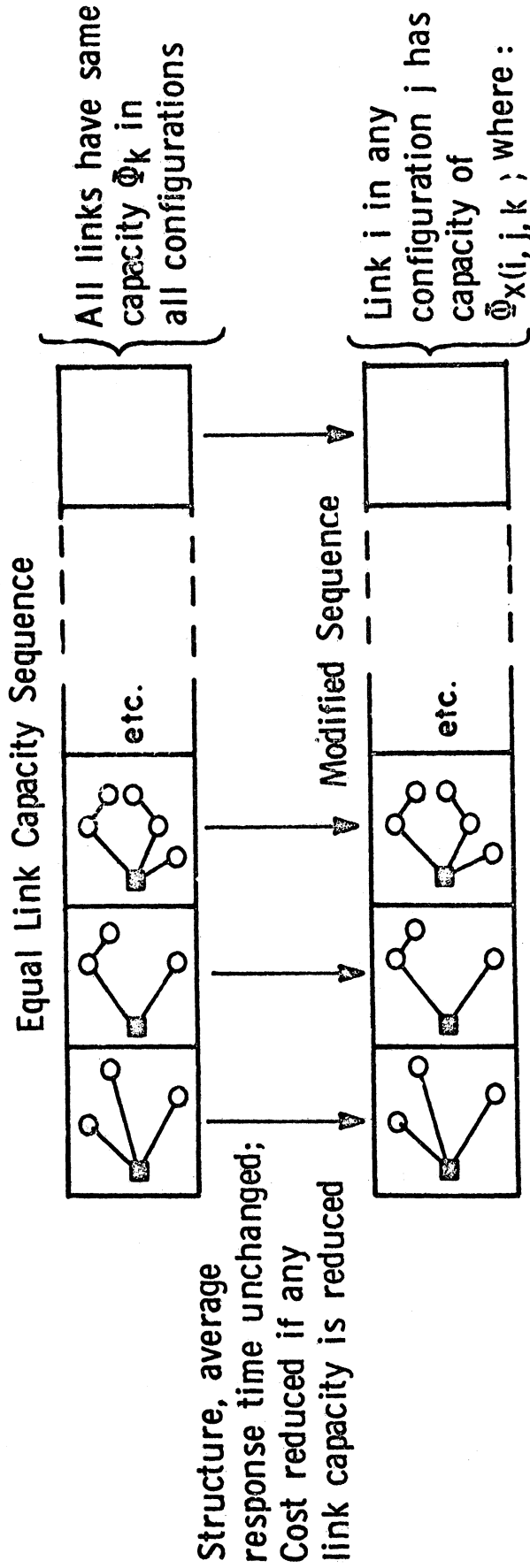
capacity values of the structures in the equal capacity sequence are systematically reduced using the technique of Section 5.3, network cost for a particular structure is reduced and the average response time increases. This behavior of key variables will be exploited in the formation of an efficient search procedure discussed in Section 6.5.

We now characterize this multidimensional design parameter space in a graphic manner.

6.3.1 The Multidimensional Design Parameter Space

If we take each structure of the equal link capacity sequence and discard all excess-capacity without changing the average response time, a new sequence of configurations is formed in which corresponding topological structures, types of data concentrators are identical to those of the original sequence. The link speeds, and hence the line costs are reduced, however, as long as some nonzero amount of excess capacity can be eliminated. The generation of this new sequence of structures is depicted in Figure 6.2. Starting with this sequence of structures in which all configurations are known to have acceptable average response times, we then consider further capacity reductions which do not necessarily preserve the average response time.

For each structure of the modified sequence of Figure 6.2, we vary the link capacity values in the manner illustrated in Figure 5.2.



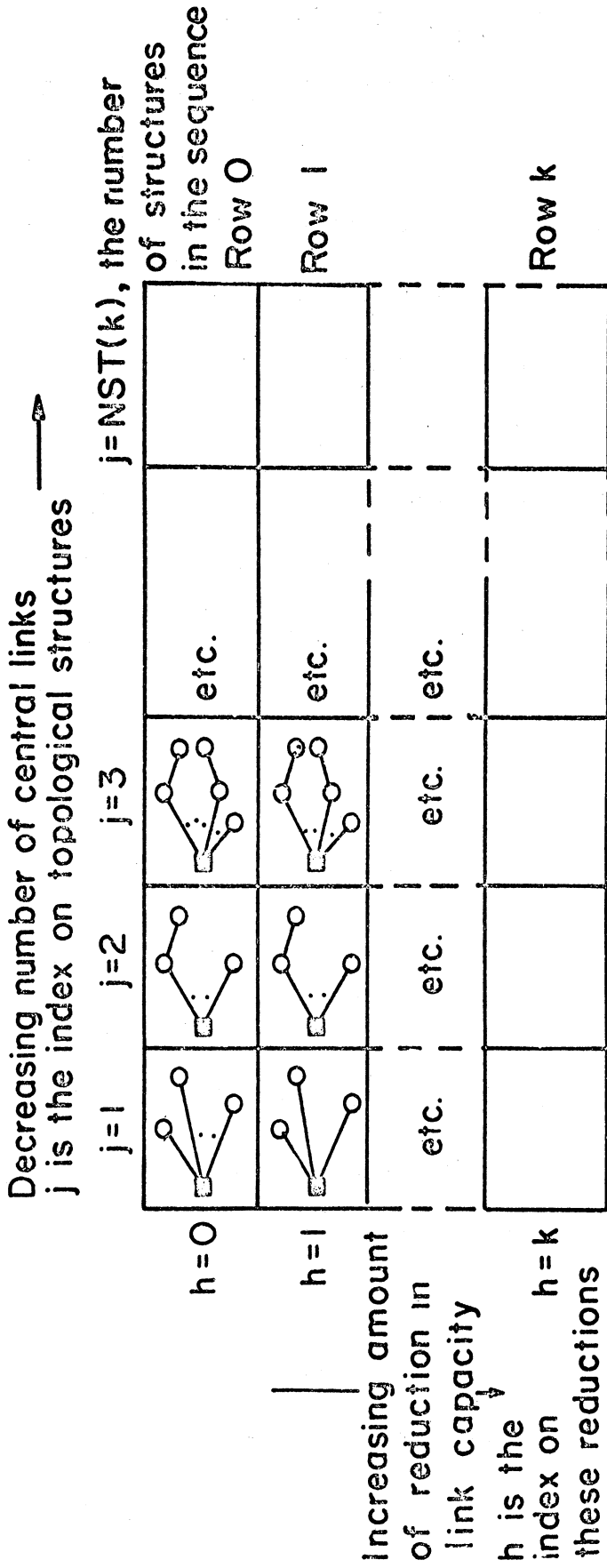
$\Phi_x(i, j, k) \triangleq$ The smallest capacity value from the set Φ which is not less than the sum of the optimal passage capacity assignments in link i of configuration j -- the original sequence having been obtained using the equal capacity - design procedure of Section 6.2.

Figure 6.2 Generation of New Sequence of Configurations by the Excess-Capacity Trimming Operation

Starting with the capacity value $\Phi_{x(i,j,k)}$, which is the value for the i th link in the j th configuration of the modified sequence generated using starting capacity value Φ_k , it is possible to depict this joint variation of structure and link capacity in Figure 6.3. In this figure, the Row 0 entries are identical to the configurations of the modified sequence of structures shown in Figure 6.2. In moving down any particular column of the space shown in Figure 6.3, the index h appearing in expression (5.1) increases, generating several different link capacity assignments for the topological structure under consideration in the column. It should be noted that it is not necessary to computationally generate all the columnar variations for the Row 0 entries in this space since the knowledge of row, column indices is sufficient to mathematically characterize the link capacity values directly from expression (5.1) for a given structure.

Each structure within one of the sequences generated by the single capacity design procedure is thus expanded into as many as k configurations which have different link capacity assignments. Thus Figure 6.3 represents a "subspace," $\mathcal{S}(\Phi_k)$, of the global design parameter space which was generated by starting with the nominal link capacity value Φ_k in all links.

In summarization, the total multidimensional design parameter space from which the minimal cost solution is obtained consists of a "family" of subspaces, $\mathcal{S}(\Phi_k)$, (Figure 6.4), one for each capacity value Φ_k in the set Φ , in which network topological structure, type of



For a given topological structure j in Row h, the capacity of link i is given by expression (5.1):

$$v_{i,h} = \Phi y(i, h) \text{ where } y(i, h) = \max \{b(i), x(i) - h\}$$

Figure 6.3 The Space of Design Parameter Variations Resulting from the Reduction of Link Capacity in the Structures of the Sequence of Figure 6.1 when these were Obtained by using the Capacity Value, Φ_k .

data concentrator, and link capacity are the parameters being varied. The dimensionality of the subspace $\mathcal{S}(\Phi_k)$ is given by $[k] [NST(k)]$ where $NST(k)$ denotes the number of structures generated by the equal link capacity network design procedures for the case when the capacity value under consideration is Φ_k .

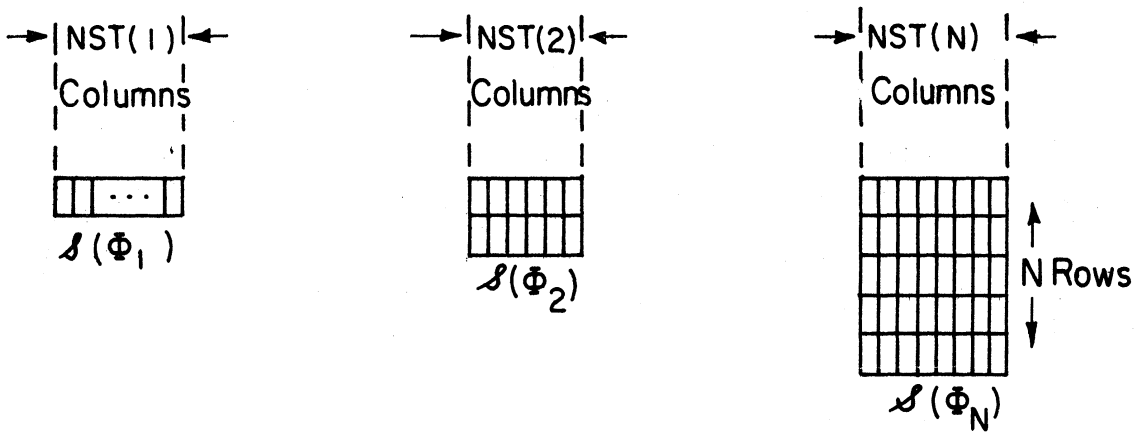


Figure 6.4 Graphical Representation of the Subspaces, $\mathcal{S}(\Phi_k) \quad k = 1, \dots, N$

The global design parameter space can be represented in terms of the collection \mathcal{S} of the individual subspaces as follows:

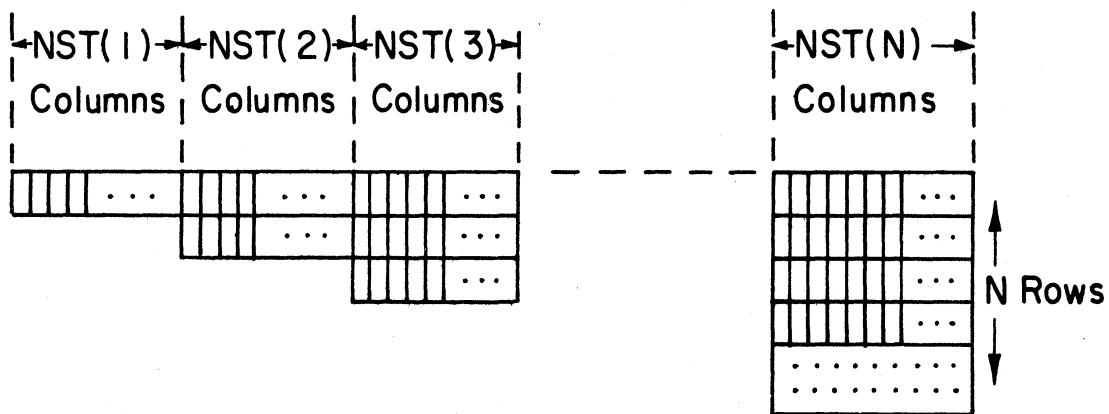


Figure 6.5 Graphical Representation of the Global Design Parameter Space \mathcal{S} where $\mathcal{S} = \bigcup_{k=1}^N \mathcal{S}(\Phi_k)$

6.3.2 Dimensionality of the Global Design Parameter Space

The maximum number of configurations, $|\mathcal{S}|$, considered in the entire space $\mathcal{S} = \bigcup_{k=1}^N \mathcal{S}(\Phi_k)$ is given by

$$|\mathcal{S}| = \left| \bigcup_{k=1}^N \mathcal{S}(\Phi_k) \right| = \sum_{k=1}^N \left(k \cdot \text{NST}(k) \right) \quad (6.3)$$

where $\text{NST}(k) \leq n-1$ in a network having a total of $n-1$ remote terminals.

A number of network configurations (points) in different subspaces may have equivalent mathematical models. Expression (6.3) for $|\mathcal{S}|$ represents an upper bound on the number of unique sets of design parameter values directly considered by the design procedure.

It must be reemphasized that while the design procedure can only consider points (configurations) in the restricted parameter space \mathcal{S} . However, a vastly larger number of definitely nonoptimal or infeasible configurations are automatically eliminated without direct consideration. For a given structure in one of the columns of the space \mathcal{S} , expression (5.2) denotes the number which are rejected because of nonoptimality. Similarly, for a given structure expression (5.3) gives the number which are rejected on the basis of infeasibility. Of course, since we consider only a subset of all possible design parameter combinations, the theoretically optimal solution may not even be contained in the space \mathcal{S} .

In preparation for the presentation of the design procedure, we consider now those computational modules which will perform off-repeated functions.

6.4 The Computational Modules

In this section we describe three computational modules which form the basis of the design procedure. One module called GNRATE implements the basic single capacity value algorithm described in Section 6.2. Another module called GTAVRT is an implementation of computations needed to assign network parameter values and then calculate the average response time in a network whose structure, capacity values, and types of concentrators are known. Network parameter values are determined from the calling arguments for the module. Finally, a module called GTCOST is described which obtains the cost of a configuration in the multidimensional design parameter space.

A. The Module GNRATE (Φ_k)

Execution of this module results in the generation of all topological structures and associated data concentrator types for the Row 0 entries in the subspace, $\mathcal{S}(\Phi_k)$ of the global design variable space illustrated in Figure 6.5. Φ_k denotes the particular link capacity value under consideration from the set Φ of choices available. This module performs exactly the same computations as those described in the algorithm of Section 6.2.

B. The Module GTAVRT (ROW, COL, Φ_k)

Execution of this module results in the computation of the average response time, T , for the network configuration of subspace $\mathcal{S}(\Phi_k)$

having row index ROW and column index COL. This module performs precisely the same computations as are described by the algorithm of Section 3.4, once the parameter values have been assigned. The topological structure and types of data concentrators are those of the COL-th structure in the sequence produced by the execution of the module GNRATE when its argument is Φ_k . The capacity value for the i th link is determined from expression (5.1) for $v_{i,h}$ when $h = \text{ROW}$.

C. The Module GTCOST (ROW, COL, Φ_k)

Execution of this module results in the computation of the total system cost for the network configuration of subspace $\mathcal{S}(\Phi_k)$ having row index ROW and column index COL. The topological structure and types of data concentrators are those of the COL-th structure in the sequence produced by the execution of the module GNRATE when its argument is Φ_k . The capacity value for the i th link is determined from expression (5.1) for $v_{i,h}$ when $h = \text{ROW}$. This module therefore performs precisely the computations indicated by the following expression:

$$\text{COST} = \sum_{i \in I} b(v_{i,h}) \ell_i + \sum_{y \in Y} \delta_y \quad (6.4)$$

- where
- $b(v_{i,h}) \triangleq$ the cost of a unit length link having capacity $v_{i,h}$ where $h = \text{ROW}$
 - $\ell_i \triangleq$ the length of link i .
 - $I \triangleq$ the set of all links in the network.
 - $\delta_y \triangleq$ the cost of the data concentrator at node y , as determined from the appropriate cost functions of Section 4.2.
 - $Y \triangleq$ the set of all data concentrator nodes in the network.

The costs for configurations in Row 0 of the subspaces $\mathcal{S}(\Phi_k)$ are produced as a by-product of the execution of the model GNRATE and actually don't have to be recalculated using the GTCOST module. In describing the algorithm, however, we have elected not to make use of this fact since the explanation is more readily comprehensible if not burdened down with a multitude of checks for special cases like this. Also, the computations performed by the GTCOST module are so straightforward that this recalculation of cost for Row 0 entries in each subspace $\mathcal{S}(\Phi_k)$ does not affect the computational efficiency of the overall procedure to any significant extent.

6.5 Network Design Procedure when Topology, Concentrator Type, and Link Capacity are Variables

In this section we present a basic algorithm for solving the problem of interest. Then in Section 6.6 a number of extensions to this basic procedure will be discussed. As will be seen, the particular extension which is most appropriate is not necessarily the same for any two applications. For this reason and since it is not difficult to modify the basic procedure for any of these possible extensions, we will present only the basic algorithm in this section. Hence, the algorithm has been tailored to obtain only the one feasible structural configuration in the space \mathcal{S} which has the lowest cost $\{v_{i,h}^{\wedge}\}$ capacity assignment.¹

¹ We refer to this configuration as the structure having the least costly $\{v_{i,h}^{\wedge}\}$ solution, where the $\{v_{i,h}^{\wedge}\}$ notation has the same meaning as it did when originally introduced in Section 5.3.

The design procedure consists of first generating the Row 0 entries in a given subspace $\mathcal{S}(\Phi_k)$ and then determining whether the minimum cost acceptable configuration within that particular subspace is a candidate for the global minimum cost configuration. Each of the subspaces $\mathcal{S}(\Phi_1), \mathcal{S}(\Phi_2)$, etc., are successively considered in this fashion. The minimum cost configuration is obtained by preserving relevant descriptive information as the search procedure evolves. Only a fraction of the actual configurations in the global design parameter space shown in Figure 6.5 are directly considered, since average response time and network cost both vary monotonically in every column of the space.

Let k be an index on the components of the set of capacity values $\{\Phi_1, \Phi_2, \dots, \Phi_N\}$. Let ROW be an index on the rows of the subspace $\mathcal{S}(\Phi_k)$ and let COL be an index on the structures of the topological sequence generated by the module GNRATE when its argument is Φ_k . For a network having $n-1$ remote terminals, when COL=1, the structure having $n-1$ central links is being considered. CMIN is the cost of the least expensive configuration with an acceptable average response time that has been encountered up to any given point in the search. After the procedure terminates, CMIN is the cost of the "best" solution from the global design parameter space $\mathcal{S} = \mathcal{S}(\Phi_1) \cup \mathcal{S}(\Phi_2) \cup \dots \cup \mathcal{S}(\Phi_N)$. T_{\max} again denotes the maximum acceptable average response time.

The descriptor information which characterizes the least costly feasible configuration considered up to any point is preserved in the variables IRWSV, ICOLSV, and KSV. These variables respectively,

save the row, column, and subspace indices for the best configuration considered up to the current point in the search. CSAV is simply a temporary storage variable which is needed at step 9 (of the algorithm to be presented) whenever a new feasible configuration is found whose cost is less than CMIN. We now present the algorithm.

6.5.1 The Basic Algorithm

- Step 0: Set $k=0$; Set CMIN to an arbitrarily large value.
- Step 1: Set $k = k + 1$; If k exceeds N , the number of discrete link capacity values, terminate the procedure.
- Step 2: Set $RWMX_k = k - 1$, where $RWMX_k$ is the largest row index in the subspace $\mathcal{S}(\Phi_k)$. Set $ROW = RWMX_k$, thus starting the search of this subspace in the bottom row. Set $COL = 1$. Execute the module GNRATE (Φ_k) .¹ This execution assigns a value to $NST(k)$, where NST is a vector whose k th component is the number of structures in the subspace $\mathcal{S}(\Phi_k)$.
- Step 3: Execute the modules GTAVRT (ROW, COL, Φ_k) and GTCOST (ROW, COL, Φ_k) .²

¹ The reader will recall from Section 6.2.1 that the last step of the algorithm for the GNRATE module sets the value of a variable, NST , equal to the number of structures generated. Since we are now dealing with several executions of this module for different argument values Φ_k , we have redefined NST as a vector so that the execution of the final step in the GNRATE module with the argument Φ_k will set the k th component of the vector NST .

² It is possible to perform step 3 in an alternate way, making use of lower bounds in an attempt to expedite the execution of the algorithm. This issue is discussed immediately after the presentation of the algorithm.

If $T > T_{\max}$ and $\text{COST} < \text{CMIN}$, go to step 5.

If $T \leq T_{\max}$ and $\text{COST} < \text{CMIN}$, go to step 9.

If $T \leq T_{\max}$ and $\text{COST} \geq \text{CMIN}$, go to step 13.

Step 4: No other configuration in the COLth column can possibly be both feasible and lower in cost than the configuration whose cost was CMIN. Hence, set $\text{COL} = \text{COL} + 1$. If $\text{COL} > \text{NST}(k)$, go to step 1. Otherwise, go to step 3.

Step 5: Control is transferred here whenever the most recently considered configuration has an unacceptable T but a cost which is less than CMIN. Some configuration in the same column with a smaller row index could possibly be a feasible solution with cost less than CMIN. Thus, set $\text{ROW} = \text{ROW} - 1$.

Step 6: Execute the module $\text{GTCOST}(\text{ROW}, \text{COL}, \Phi_k)$. If $\text{COST} \geq \text{CMIN}$, increment ROW by 1 and go to step 4.

Step 7: Execute the module $\text{GTAVRT}(\text{ROW}, \text{COL}, \Phi_k)$. If $T > T_{\max}$, go to step 5.

Step 8: Set $\text{CMIN} = \text{COST}$ and proceed to save the indices of this new configuration by transferring to step 12.

Step 9: Control is transferred here whenever the most recently considered configuration has both an acceptable T and a cost less than CMIN. There may, however, be another still better configuration in this COLth column. But in case there is not, the cost of the configuration already found must be saved.

Hence, save the present value of COST by setting $CSAV = COST$. Increment ROW by 1. If ROW exceeds $RWMX_k$, set $CMIN = COST$ and go to step 11.

- Step 10: Execute the module GTAVRT (ROW, COL, Φ_k). If $T \leq T_{max}$, execute the module GTCOST (ROW, COL, Φ_k) and go to step 9. Otherwise, set $CMIN = CSAV$.
- Step 11: Since incrementing ROW at step 9 produced an unacceptable T or an excessively large row index, the best configuration in the COLth column has been found. Set $ROW = ROW - 1$.
- Step 12: Save the current ROW, COL, and k values in IRWSV, ICOLSV, and KSV, respectively. Go to step 4. (Upon completion of the procedure, the final values of IRWSV, ICOLSV, and KSV indicate the minimum cost configuration in the space by row, column, and subspace, respectively).
- Step 13: Control is transferred here whenever the most recently considered configuration has an acceptable T but a cost not less than CMIN. If there exists a better solution in this COLth column than the best one found thus far, it must have a larger row index than the most recently considered configuration. Hence set $ROW = ROW + 1$. If $ROW > RWMX_k$, decrement ROW by 1 and go to step 4. Otherwise, go to step 3.

At steps 3, 7, and 10 in the algorithm presented it would certainly be possible and sometimes advantageous to calculate a lower bound (such as the one described in Section 5.3) instead of using the module

GTAVRT to obtain the precise value of average response time every time any of the steps is executed. The tradeoff here should be rather obvious. If the lower bound calculation (which is cheaper to perform than the precise calculation) produces a value T_{LB} which exceeds T_{max} , then the exit for $T > T_{max}$ could be taken without actually using the GTAVRT module.

This approach, however, would require more computation than the present method whenever $T_{LB} \leq T_{max}$ since in this case one would have to go still further and use the GTAVRT module to obtain a precise average response time. This is so because knowing $T_{LB} \leq T_{max}$ tells us nothing about whether the actual average response time satisfies the constraint value.

6. 5. 2 Concluding Discussion

The algorithm, as it has been presented in Section 6. 5. 1 obtains only one solution configuration from the entire space \mathcal{S} of Figure 6. 5 - namely the structure whose $\{v_i, \hat{h}\}$ solution has the lowest cost of any in the space. In the extensions to the basic algorithm which will be considered in the following section we will show that an improved solution can usually be obtained by applying branch and bound procedures to optimally assign link capacity values for each of the different network topologies of the space \mathcal{S} . The basic search algorithm would be modified in these situations to calculate and preserve appropriate descriptive information for the $\{v_i, \hat{h}\}$ solution for each structure in the space.

We now discuss extensions to the basic procedure of Section 6. 5. 1 and factors which influence the selection of these extensions for a given application.

6.6 A Critical Evaluation of the Procedure; Extensions to the Basic Algorithm

In the method presented in Section 6. 5. 1 the different columns (topological structures) of the space \mathcal{S} are searched and the solution is taken to be structure whose $\{v_i, \hat{h}\}$ capacity assignment is the least costly of any in the space. We have continually noted that by so doing a sub-optimal solution is obtained. It is of interest to precisely summarize the sources of deviation from true optimization methods which are inherent in this algorithm and to discuss how and if they may be eliminated.

There are two primary sources of nonoptimality in the design procedure proposed in Section 6. 5. 1:

- (1) For each column (topological structure) in the space \mathcal{S} , the search decisions are based on the cost of an approximate solution. By using this $\{v_i, \hat{h}\}$ assignment of link capacity, we assume that no less costly feasible assignment exists. This represents a sub-optimal approach.
- (2) As was discussed in Section 6. 2, the set of topological structures which are considered is restricted to those in sequences produced by the GNRATE module for different values of its capacity argument, Φ_k . This represents a suboptimal approach since some structure

not even considered by the procedure could be better than the best one in the space \mathcal{S} .

Although it is difficult to evaluate how seriously each of these sources of nonoptimality affects the solutions obtained by the procedure, previously discussed qualitative arguments and computational experience tend to indicate that the first factor is far more significant than the second. This first source of deviation can be eliminated whenever it is possible to completely execute an additional computation following the basic design procedure of Section 6.5.1. This final step consists of using branch and bound methods for every structure in the space \mathcal{S} and choosing the structure having the least costly optimal link capacity assignment. The results of the representative case studies considered in Chapter 7 will illustrate that significant improvements in the cost of the solution configuration can be obtained by applying this extra step.

In order to accomplish the elimination of this first source of nonoptimality in the most efficient manner, the algorithm of Section 6.5 would be modified to first obtain the $\{v_{i,h}^{\wedge}\}$ capacity assignments for every column (structure) in the space \mathcal{S} . Then for the structure whose $\{v_{i,h}^{\wedge}\}$ solution is the least costly, the branch and bound optimal capacity assignment procedure discussed in Section 5.3.1 would be used to obtain the true minimum cost feasible solution. Other structures of the space \mathcal{S} would then be successively subjected to the same optimal capacity assignment procedure, and considered in the order of increasing cost for the $\{v_{i,h}^{\wedge}\}$ solution.

In this case where branch and bound problems are being considered for several different columns (structures) of the space \mathcal{S} , the CMIN parameter of the branch and bound algorithm should be redefined as the cost of the cheapest configuration found up to any point in the procedure, including all topological structures examined thus far. To put it another way, the value of CMIN should be carried over from one column (structure) to the next. By starting with the configuration having the least costly $\{v_i, \hat{h}\}$ solution the branch and bound procedures will generally find the true minimum cost assignment over all columns (structures) in the space \mathcal{S} with the least amount of computational effort in comparison to other alternative approaches.

It may happen that the same topological structure appears more than once in the space \mathcal{S} . Obviously, the branch and bound optimization procedure need not be applied for the same structure more than once. Of the identical topological structures, the configuration having the least costly $\{v_i, \hat{h}\}$ solution should be considered; all other replicas of the same structure may be ignored.

This approach can be applied effectively in problems where several link capacity optimization problems such as those discussed in Section 5.3 can be successfully undertaken and solved within a reasonable length of time. Experience suggests that in relatively small problems (say those with 10 or fewer remote terminals) this step can be used effectively since it is not likely that very many of the structures in the space will need to be considered in much depth. For example, in every problem

considered for the eight node example of Section 7.3, these methods quickly produced the least costly structure in the space \mathcal{S} having optimally assigned link capacity values. In all of these problems, this structure turned out also to be the one having the least costly $\{v_i, \hat{h}\}$ solution. The usage of bounds for cost and response time resulted in a rapid rejection of all other topological structures in the space \mathcal{S} when the branch and bound procedure was used.

On the other hand, when this same technique of sequentially optimizing capacity assignments for every structure in \mathcal{S} was used in the 23 node example considered in Chapter 7, the cheapest configuration with optimally assigned link capacities was found not to be the structure having the least expensive $\{v_i, \hat{h}\}$ solution of all those in space \mathcal{S} . Thus, by considering the capacity assignment optimization procedure for other structures in the space \mathcal{S} , a significant reduction in the cost of the cheapest feasible network was accomplished. Further discussion of this example will be given in Section 7.5.

If computational limitations are not a factor, this aforementioned refinement to the basic procedure should always be considered. However, it has been noted in Section 5.3.2 that there are certain types of problems for which the branch and bound optimization procedures will be of limited utility, at best. Attempts can be made to improve the $\{v_i, \hat{h}\}$ capacity assignments for as many of the different structures in the space \mathcal{S} as it is possible to consider with a reasonable amount of computer execution time. It may, however, turn out that none of the structures can

be totally optimized within a reasonable length of time. In this case, one must be willing to live with the least costly $\{v_i, \hat{h}\}$ solution, any improvements found (even though it is not known whether they are optimum), or any obvious manual improvements which the optimization procedures might fail to find in the maximum quantum of computation time which one is willing to expend for such purposes.

There appears to be little which can be done to completely eliminate the second source of nonoptimality in the general case. Myriad heuristic possibilities exist for attempting structural improvements to the solutions obtained. However, there is no assurance that any reductions of this type can be effected, in general. It will be seen in the case studies which now follow that this second source of suboptimality is usually not as significant as that arising from the approximate capacity assignments.

Chapter VII

APPLICATIONS - SEVERAL REPRESENTATIVE CASE STUDIES

7.1 Introduction

In this chapter several illustrative network design examples are considered. For each of the examples, the results of the design procedure proposed in this thesis have been compared with the solutions obtained by using variants of other design algorithms such as the minimum spanning tree and the previously discussed Esau-Williams procedures.

It must be emphasized that in order to make any of these comparisons whatever, the analysis procedures developed in Chapter 3 of this work must be called upon, since neither the minimum spanning tree or the Esau-Williams procedures address the issue of analytic performance measures. Furthermore, it has been shown that these other design approaches are limited to the case of networks in which link capacity is not treated as a design variable. It was also previously demonstrated that significant improvements in cost-effectiveness can be obtained by treating link capacity as a design variable.

The particular examples to be considered in this chapter have been selected for the following reasons:

1. To illustrate the use of the design algorithm in detail.
2. To demonstrate the usefulness and limitations of the procedure.

3. To form the bases for comparative evaluations of the several design procedures in realistic application environments.

Remote terminal distributions which are relatively homogeneous in the geographic sense have been considered. On the other hand, a highly clustered example has also been treated. In this highly clustered case, the optimal structure is intuitively evident for certain response time ranges. Hence it serves to calibrate in some small way the accuracy of the various alternative strategies. It is significant to note that for this particular example the design procedure of this thesis obtained a solution which differed negligibly from the known optimum. This fact inspires confidence in the quality of the solutions that are obtained by the procedure in more complex networks. Qualitatively, this is so because highly complex networks are not treated any differently by the design procedure than are those which may have readily apparent solutions. To put it another way, the design program is incapable of intuition - hence all problems look the same to it regardless of how easy or difficult they might appear to an intelligent, well trained professional network designer. This characteristic represents one of the most noteworthy features of the design model developed in this work.

It is also significant to note that in none of the examples considered do any of the other network design procedures obtain a less costly feasible configuration than the one obtained by the algorithm of this thesis. In two of the three examples considered it will be shown that manual improvements could be made to the solution structure obtained by the

design procedure of this work.¹ However the resulting reduction in cost will be so small as to be negligible.

Before considering the specific examples, a discussion of the various alternative design strategies used in the comparisons is presented.

7.2 Alternative Design Procedures to be Considered

In addition to the Esau-Williams and minimum spanning tree algorithms discussed in Chapter 4, several variants of these algorithms are shown to consistently obtain better solutions than the unmodified algorithms. We have considered these improvements in formulating a variety of alternative design strategies to be used for comparative purposes.

7.2.1 Minimum Spanning Tree Algorithm with Excess Capacity Eliminated

It was shown in Chapter 5 that a significant amount of excess capacity can frequently be discarded from links of certain topological structures without changing the average response time. One intuitive variant of the minimum spanning tree algorithm is to discard the excess capacity from the links of the solution configuration in accordance with the method described in Section 5.2. It has been assumed that only multiplexors are used for data concentration in this particular variant, and that a single line speed is used in obtaining the solution configuration

¹ We include in this procedure the extensions discussed in Section 6.6.

before any excess capacity trimming if performed.

7.2.2 Esau-Williams Algorithm with Excess Capacity Eliminated

This variant is identical in nature to the one described in the previous section, only the topological structure is obtained using the basic Esau-Williams procedure for some particular link capacity value. After the equal link capacity solution has been obtained, the excess capacity is trimmed in the manner described in Section 5.2. Multiplexors are again used exclusively. It is quite interesting to note that in numerous particular examples (but not always), the Esau-Williams and minimum spanning tree algorithms produce identical structures; hence in such situations these two techniques will lead to network configurations having identical cost-performance characteristics.

7.2.3 Other Strategies

An additional pair of strategies was considered, even though neither ever yielded any solutions which were favorable in comparison to those of the previously discussed procedures. For the solutions obtained by the basic Esau-Williams and minimum spanning tree algorithms, the strategies consisted of placing a MSC concentrator at the node where the greatest reduction in average response time could be obtained, assuming equal link capacities. Excess capacity was then trimmed in accordance with the method of Section 5.2.

In all examples considered, the differential cost of upgrading a MX concentrator to a MSC was large enough to make the resulting

configurations significantly more costly than other feasible solutions.

We have summarized in Table 7-A the alternative strategies which were used in the comparison studies which follow. The acronyms are introduced for ease in referencing the strategies in ensuing discussion. It should be noted that the last strategy in Table 7-A refers to the solutions which were obtained using the basic procedure described in Section 6.5. Extensions of the type discussed in Section 6.6 are also considered but are discussed separately for each example since the most meaningful type of modification depends on the particular problem.

We now turn to the first case study.¹

7.3 An Eight Node Network Design Example

Consider the seven eastern U. S. cities listed in Table 7-B. It is desired to connect the terminals in each city to a central facility located in Washington, D. C. . Assume that each active source of message traffic has a mean Poisson arrival rate of $1/30$ mess/sec. Let all negative exponential message length distributions have a mean of 120 bits/mess. The mileage table used for this example is presented in Table 7-C. We confine the link capacity values to the seven speeds shown in Table 7-D; appropriate costs per unit length are also indicated therein.

¹ In all case studies, we are concerned only with the design of the forward network.

Strategy	Acronym
1. Basic Esau-Williams algorithm for the capacity value Φ_k	BEW(Φ_k)
2. Basic minimum spanning tree for the capacity value Φ_k	BMST(Φ_k)
3. Esau-Williams for the capacity value Φ_k , followed by discarding of excess capacity	EWED(Φ_k)
4. Minimum Spanning Tree for the capacity value Φ_k , followed by discarding of excess capacity	MSTED(Φ_k)
5. Esau-Williams for the capacity value Φ_k ; one MSC added at "best" node; then the excess capacity discarded	EWMSC(Φ_k)
6. Minimum Spanning Tree for the capacity value Φ_k ; one MSC added at the "best" node; then the excess capacity discarded	MSTMSC(Φ_k)
7. Algorithm described in Section 6.5 of this work; no extensions	BDOLL

Table 7-A

The Alternative Design Strategies Used in the Comparative Studies

Location of Remote Terminal	Maximum Number of Active Terminals During Busy Period	Total Arrival Rate of Messages
1. Chicago, Ill.	10	.3333 m/s
2. Detroit, Mich.	9	.3000 m/s
3. Charlotte, N. C.	4	.1333 m/s
4. Miami, Fla.	6	.20 m/s
5. New Orleans, La.	6	.20 m/s
6. New York, N. Y.	12	.40 m/s
7. Tallahassee, Fla.	4	.1333 m/s

Table 7-B

The Arrival Statistics for the Example of Section 7.3

		A	B	C	D	E	F	G	H
Chicago	(A)	—	235	601	1202	841	740	832	589
Detroit	(B)	235	—	519	1151	981	508	807	383
Charlotte	(C)	601	519	—	653	654	542	378	321
Miami	(D)	1202	1151	653	—	675	1093	381	920
New Orleans	(E)	841	981	654	675	—	1184	347	1493
New York	(F)	740	508	542	1093	1184	—	910	228
Tallahassee	(G)	832	807	378	381	347	910	—	677
Washington	(H)	589	383	321	920	1493	228	667	—

Table 7-C

Mileage Matrix for the Example of Section 7.3

Capacity Value	Monthly Cost per Unit Length in Dollars
450	1.4
900	1.7
1350	1.875
1800	2.00
2700	2.20
3600	2.375
5400	2.55

Table 7-D

Cost-Capacity Relationship for a Unit Length Link

In Table 7-E, the results obtained by using the various design strategies discussed in Section 7.2 are discussed. Table 7-F summarizes the results of improvements which were made to the solutions of the BDOLL strategy. These improvements resulted from using backtrack programming procedures to optimally assign link capacity in all structures of the design parameter space which are obtained by the algorithm of Section 6.5. The strategies behind these improvements were discussed at length in Section 6.6. The same data which is tabularized in Tables 7-E and 7-F is plotted on the graph of Figure 7.1 for illustratory purposes.

Design Strategy	Cost of SOLN Config, \$/Mo.	Av. Response Time, Sec/Mes
BDOLL, $T_{\max}=4.0$	A 3507	.8127
$T_{\max}=2.0$	A 3507	.8127
$T_{\max}=1.0$	A 3507	.8127
$T_{\max}=.65$	B 3825	.6209
$T_{\max}=.5$	C 3900	.369
$T_{\max}=.25$	D 4200	.2385
BEW(450), BMST(450)	E 3507	.8127
BEW(900), BMST(900)	F 4189	.369
BEW(1350), BMST(1350)	G 4587	.2385
BEW(1800), BMST(1800)	H 4871	.1763
BEW(2700), BMST(2700)	5325	.1158
BEW(3600), BMST(3600)	5723	.0862
BEW(5400), BMST(5400)	6121	.057
EWED(450), MSTED(450)	E 3507	.8127
EWED(900), MSTED(900)	I 3970	.369
EWED(1350), MSTED(1350)	J 4200	.2385
EWED(1800), MSTED(1800)	K 4623	.1763
EWED(2700), MSTED(2700)	L 4914	.1158
EWED(3600), MSTED(3600)	5318	.0862
EWED(5400), MSTED(5400)	5680	.057
EWMSC(450), MSTMSC(450)	4407	.604
EWMSC(900), MSTMSC(900)	5089	.277
EWMSC(1350), MSTMSC(1350)	5446	.1798

Table 7- E

Partial Experimental Results for the Example of Section 7.3.
Points plotted on graph of Figure 7.1 are denoted
by an alphanumeric preceding the cost.

Design Improvement Strategy	Index on Graph of Fig. 7.1	Monthly Cost of Optimal Capacity Assignment	Monthly Cost of Suboptimal Capacity Assignment	Capacity Changes from Suboptimal to Optimal Assignment
Optimize the capacity assignments in all structures of the design parameter space, choosing the structure having the least cost-ly feasible assignment, $T_{\max} = .5$	M	\$3718.4	\$3900.	In Figure 7.2(b), reduce the capacities of links 6 and 7 from 900 b/s to 450 b/s.
Same strategy, $T_{\max} = .25$	N	\$4103.	\$4200.	In Figure 7.2(c), change capacity of Link 3 from 1350 b/s to 1800 b/s and of Link 2 from 1350 b/s to 900 b/s.

- Notes: 1. Solutions were obtained in less than 30 sec. of CPU execution time on the 360/67 using the backtracking algorithm of Section 5.3.
2. For values of $T_{\max} \geq 1.$, the suboptimal assignments are also the optimal assignments for the structure in question.

Table 7-F

The Balance of the Experimental Results for the Case Study of Section 7.3

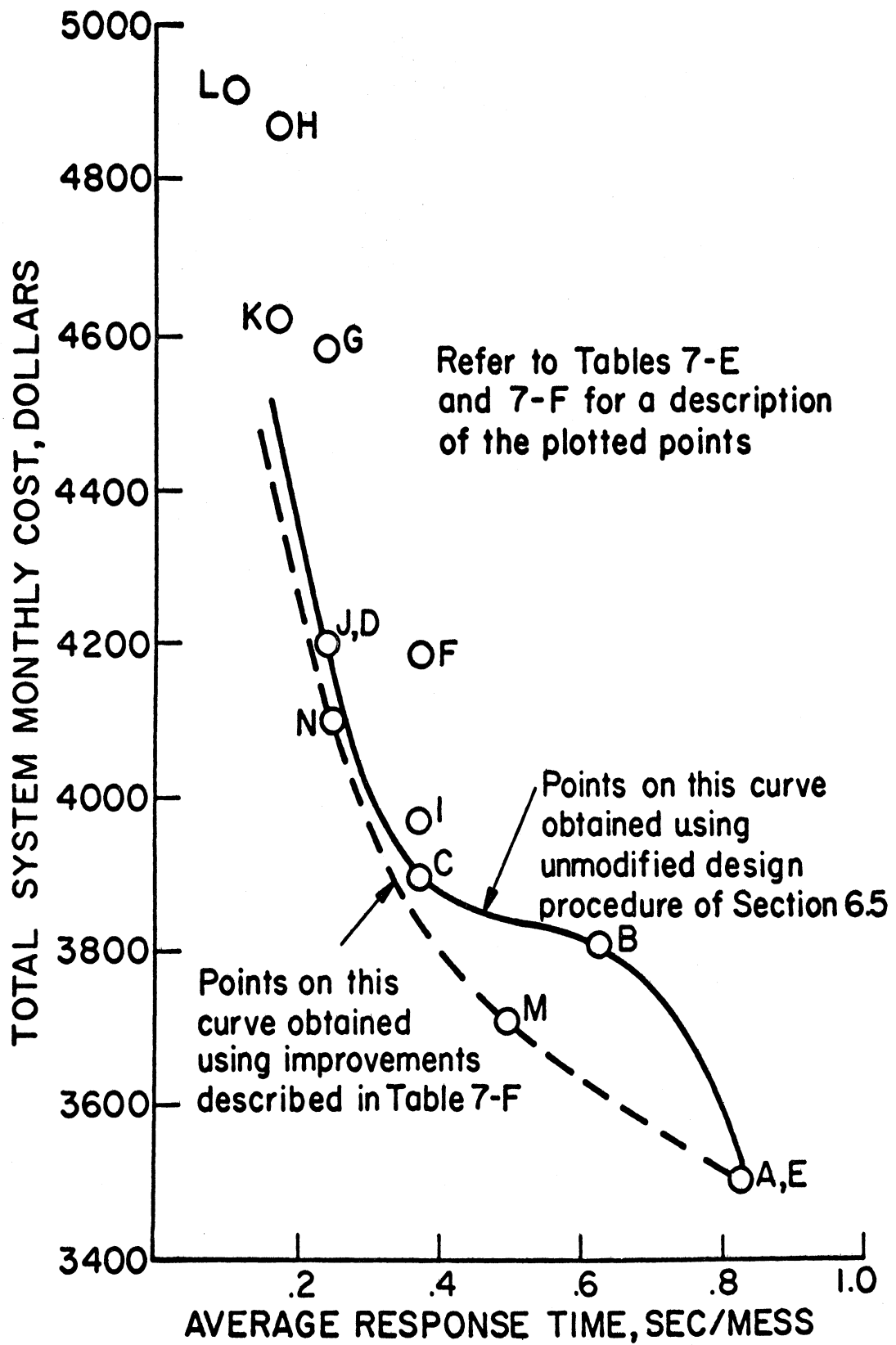


Figure 7.1 The Experimental Results for the Case Study of Section 7.3

7.3.1 Discussion of Results

Figure 7.1 represents a plot of the costs of the best solution obtainable with a given design strategy as a function of the average response time for that particular configuration. For the BDOLL strategy, by varying T_{\max} it is possible to generate new points in addition to the ones shown, and in essence, fill out the curve in more detail. This curve represents a feasibility boundary since by using the BDOLL strategy it is possible to realize any network with a cost-performance characteristic corresponding to a point in the area to the upper-right side of the curve in Figure 7.1.

Of course, if the theoretically optimum network configuration could be determined for each T_{\max} , the locus of points corresponding to the cost of the theoretically optimal network for a given T_{\max} would form a curve having a shape similar to the dashed line curve of Figure 7.1. However, it would be positioned so that, for a particular value of average response time, no point determined by any of the procedures of this work would fall below the point on this locus of optimal solution costs.

It should be emphasized that there is no way to effect a systematic variation in the cost and average response time values of structures obtained using alternative design procedures. These strategies by definition produce one and only one point in the cost - average response time space for a given starting link capacity value, Φ_k . Only by varying Φ_k can different points be generated. Hence the reader is cautioned to

resist all temptations to connect such points by continuous curves of any kind.

Figure 7.2 illustrates the solutions obtained using the unmodified design procedure of this work for five different values of T_{\max} , the maximum acceptable average response time. It is interesting to note that structure is invariant for the entire range of T_{\max} considered in this example, even though the cost of the solutions does change for the smaller values of T_{\max} considered. For large values of T_{\max} , in particular, those not less than 1.0 sec/mess, both the structures and the costs of the solutions are identical. Thus no economy is even realized by tolerating average response times above this value. One strongly suspects that the "suboptimal" solutions obtained by the design procedure may be optimal in this particular case, even though no verification of this claim can be made, in general. The first (and likely the most significant) source of nonoptimality discussed in Section 6.6 has however been eliminated for those points which form the lower curve of Figure 7.1. Nonetheless, it is difficult to comment further on the second source of nonoptimality in the context of this particular example.

7.3.1.1 Illustration of How the Structures are Generated

In order to illustrate the usage of the design algorithm presented in Section 6.2, we will monitor the steps taken by the design procedure when the Φ_k capacity value is 1350 b/s and $T_{\max} = .5$.

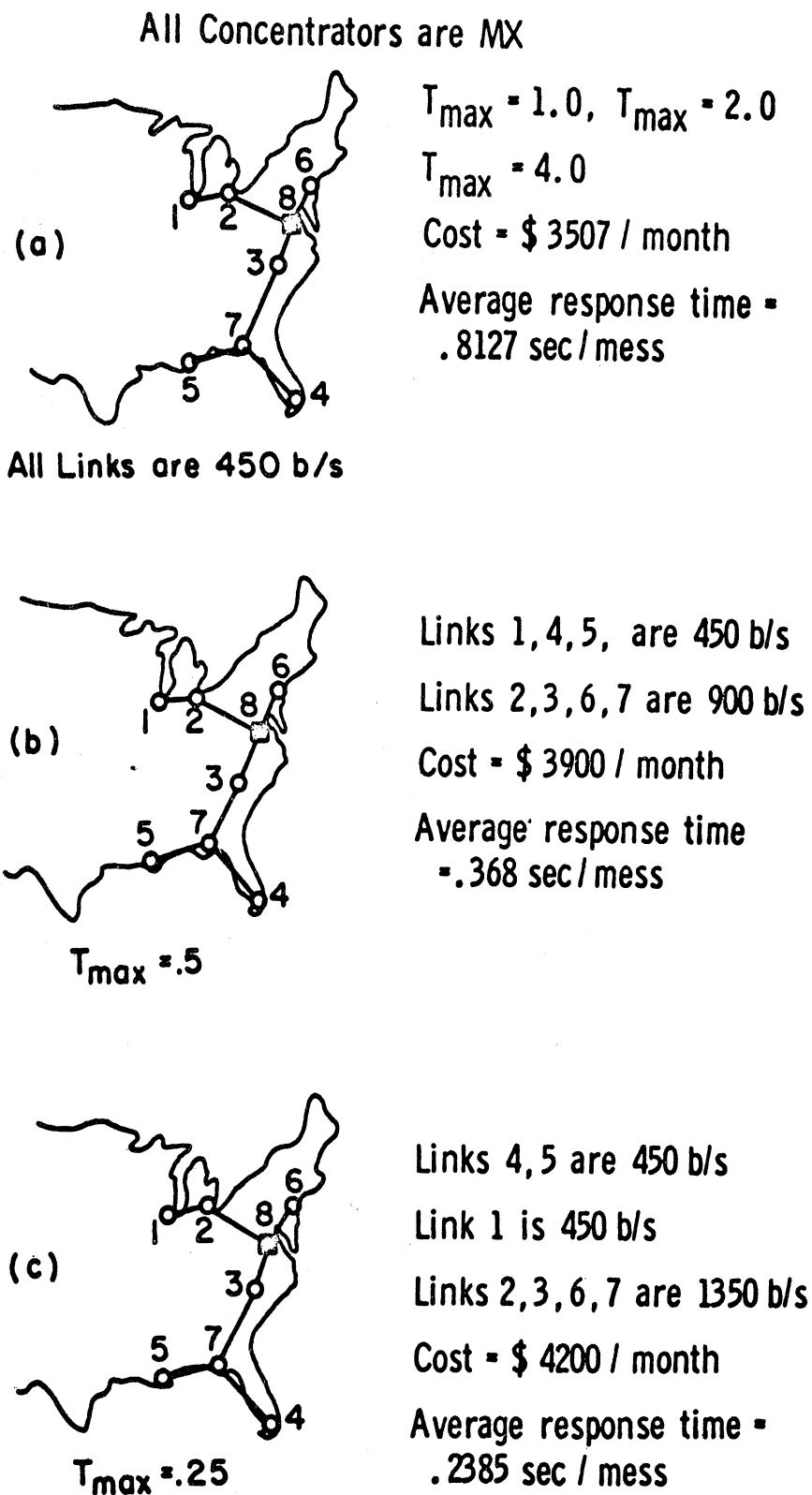


Figure 7.2 The Solutions Obtained by the Unmodified Design Procedures of this Thesis for the Case Study of Section 7.3

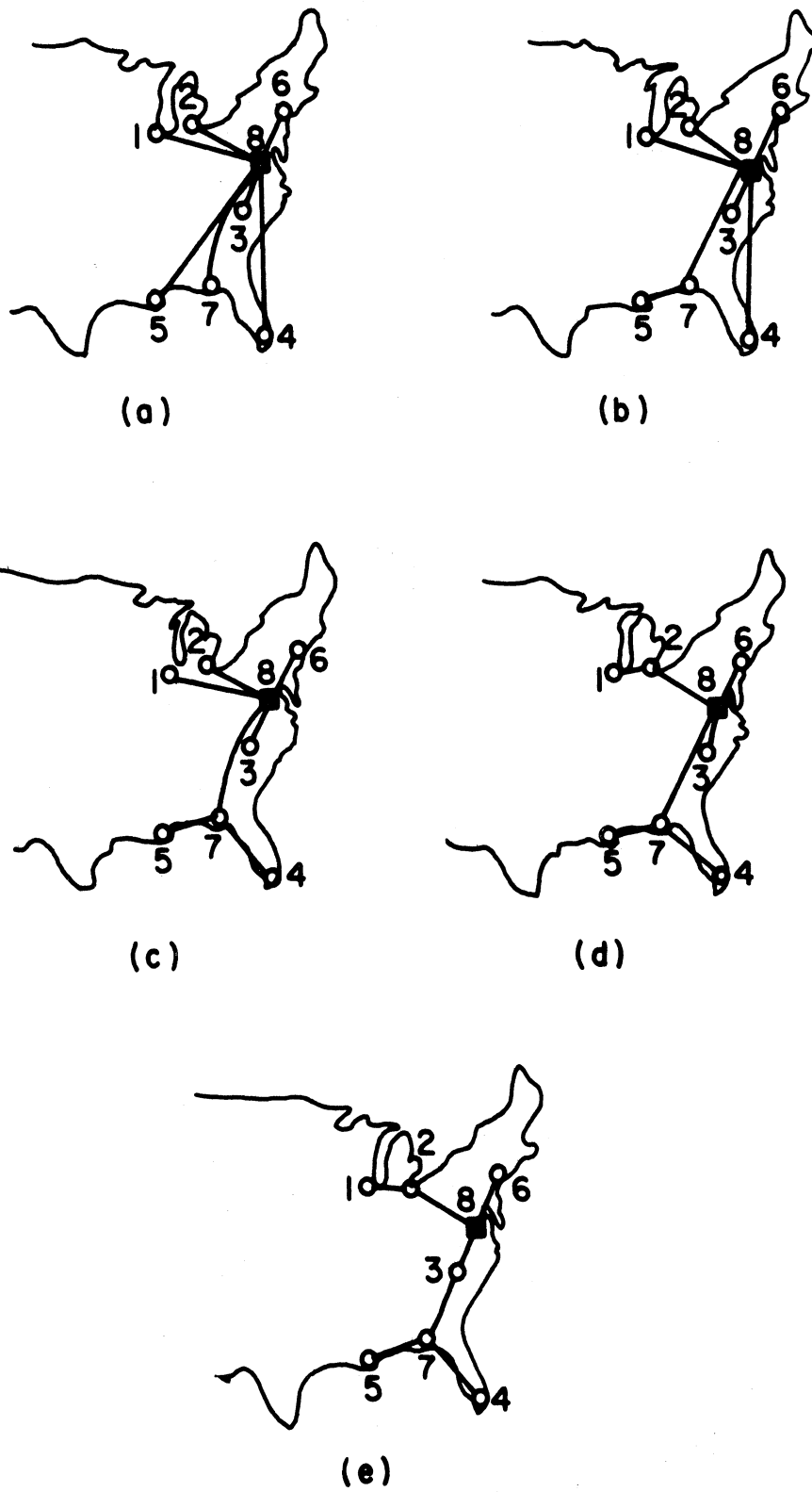


Figure 7.3 The Generation of Topological Structures in a Typical Situation

1. The point to point network of Figure 7. 3(a) is established as structure 1.
2. On examining the structural modifications which could be performed to the star network, it is determined that by deleting central link 5 and inserting link 5, 7 a net savings of \$2048. 75/month could be realized (\$2148. 75 of line savings less \$100. for the multiplexor at node 7.) This new structure has an average response time of .109 sec/mess so the change is made to obtain the configuration in Figure 7. 3(b).
3. Examining the alternative structural modifications which would be performed on the Figure 7. 3(b) network, it is determined that by deleting central link 4 and inserting link 4, 7 a net savings of \$985. 63/month could be realized (\$1010. 63 of line savings less \$25. for the additional multiplexor channel at node 7). This new structure has an average response time of .15 sec/mess, so the change is made to obtain the configuration of Figure 7. 3(c).
4. Examining the alternatives which exist for the structure of 7. 3(c), it is determined that by deleting central link 1 and inserting link 1, 2 a net savings of \$553. 75/month could be realized (\$663. 75 of line savings less \$100. 00 for the additional multiplexor at node 2.) This new structure has an average response time of .186 sec/mess so the change

is made to obtain the configuration of Figure 7.3(d).

5. The last modification which can yield any savings is now considered. By deleting central link 7 and inserting link 7,3 a savings of \$460.63/month could be realized. The resulting average response time is .238 so that the change is made, resulting in the structural configuration of Figure 7.3(e).
6. No further savings exist, so the last structure obtained.

As has been previously noted, the entire design operation consists of generating a sequence of structures for each of the capacity values which are being considered. Then the $\{v_{i,h}^{\wedge}\}$ solutions are obtained by discarding all unneeded capacity in the manner described in Section 5.2. The least costly of the $\{v_{i,h}^{\wedge}\}$ solutions for all structures in the design parameter space is the solution configuration for the unmodified design procedure of Section 6.5.

7.3.1.2 The Optimal Passage Capacity Assignments for a Typical Structure

We conclude our discussions of this example by presenting the optimal passage capacity assignments for a typical solution configuration. These values for the structure of Figure 7.2(a) when $T_{\max} \geq 1.0$ sec/mess are shown in Table 7-G. We have denoted the optimal assignment for the i -th terminal on the j -th link by $\alpha_{i,j}$ where these values are only defined for i,j pairs such that terminal i has a passage on link j .

Passages	Optimal Capacity Assignment
$\hat{a}_{1,1} = \hat{a}_{1,2}$	231.9 b/s
$\hat{a}_{2,2}$	218.1 b/s
$\hat{a}_{3,3}$	99.147 b/s
$\hat{a}_{4,4} = \hat{a}_{4,7} = \hat{a}_{4,3}$	125.85 b/s
$\hat{a}_{5,5} = \hat{a}_{5,7} = \hat{a}_{5,3}$	125.85 b/s
$\hat{a}_{6,6}$	450.00 b/s
$\hat{a}_{7,7} = \hat{a}_{7,3}$	99.147 b/s

Table 7-G

The Optimal Passage Capacity Assignments for the Links
of Structure in Figure 7.2(a)

7.4 A Twenty City Case Study

We now consider the much larger network problem associated with connecting the remote terminals located in the cities of Figure 7.4 to a central facility located in New York.¹

It is assumed that a single buffered remote terminal is active at each of the remote sites during the period of time considered. The same seven values of capacity will be used as for the previous example. Again it is assumed that messages enter the system from each remote terminal at the Poisson rate of $1/30$ mess/sec. The message length distributions are all assumed to be negative exponential with mean of 120 bits/mess.

Table 7-H gives the distances (in miles) which were used by the procedure in obtaining the solution configurations. Tables 7-I and 7-J portray the experimental results which were obtained using the alternative design strategies discussed in Section 7.2. The solutions structures and capacity assignments for selected values of T_{\max} are depicted in Figure 7.4. Some of these points have also been plotted on the graph of Figure 7.5 for illustrative purposes. The reader is again cautioned against interpolating between points on the graph which were obtained using procedures other than the one proposed herein. There is no way to form such curves since the existing procedures

¹ This example is similar to one considered in Chapter 2 of Desmonde [47] and is considered a representative example for this reason.

Buffalo	0.0	279	296	489	695	880	950	996	1058	1189	239	723	462	396	394	435	660	615	340	291
Phil., Pa.	0.0	0.0	120	340	652	742	802	832	909	1010	454	955	652	578	538	570	800	666	354	98
Wash., D. C.	0.0	0.0	0.0	228	520	625	701	749	800	900	398	907	589	508	460	485	705	542	245	228
Raleigh	0.0	0.0	0.0	0.0	350	413	488	513	587	692	507	983	632	565	470	485	664	444	240	440
Atlanta	0.0	0.0	0.0	0.0	0.0	292	375	401	406	598	597	898	580	546	440	432	485	213	351	750
Jacksonville	0.0	0.0	0.0	0.0	0.0	0.0	89	125	162	312	834	1180	855	813	712	709	762	590	558	834
Daytona	0.0	0.0	0.0	0.0	0.0	0.0	0.0	49	120	236	910	1275	938	904	798	800	850	614	640	900
Orlando	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	88	198	957	1300	992	947	825	830	883	618	678	930
Tampa	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	204	983	1298	1001	952	847	842	874	620	704	1009
Miami	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1151	1536	1180	1140	1030	1030	1079	811	860	1093
Detroit	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	521	240	178	198	232	450	468	285	492
Minneapolis	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	346	410	503	495	450	735	735	998
Chicago	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	81	165	157	248	403	403	720
South Bend	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	118	115	280	334	334	640
Anderson, Ind.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	37	264	270	245	605
Indianapolis	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	230	261	260	640
St. Louis	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	266	465	875
Nashville	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	312	755
Charleston, W. Va.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	454	0.0
New York, N. Y.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 7-H

Mileage Matrix Used for Example of Section 7.4

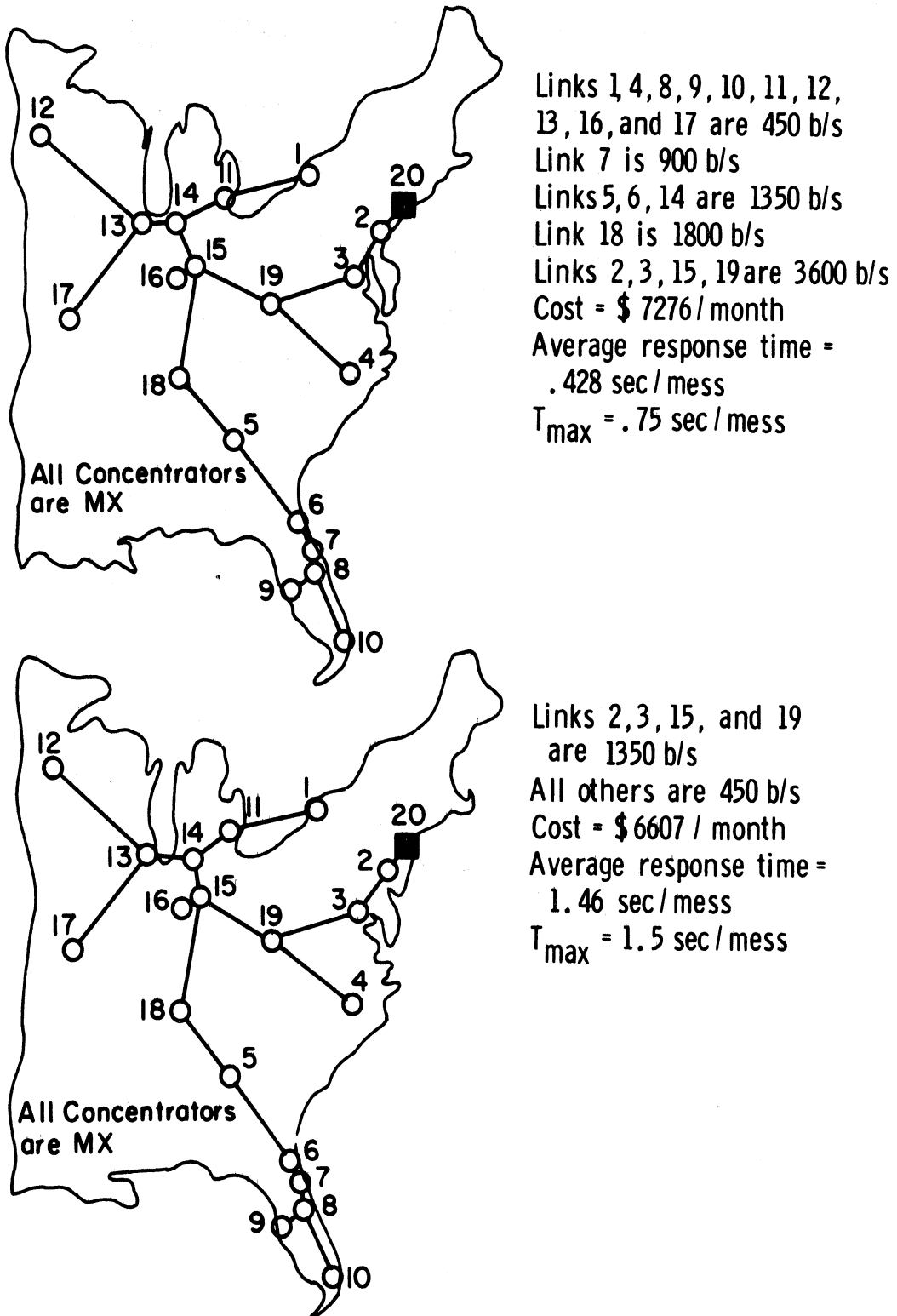


Figure 7.4 The Solution Configurations for the Case Study of Section 7.4

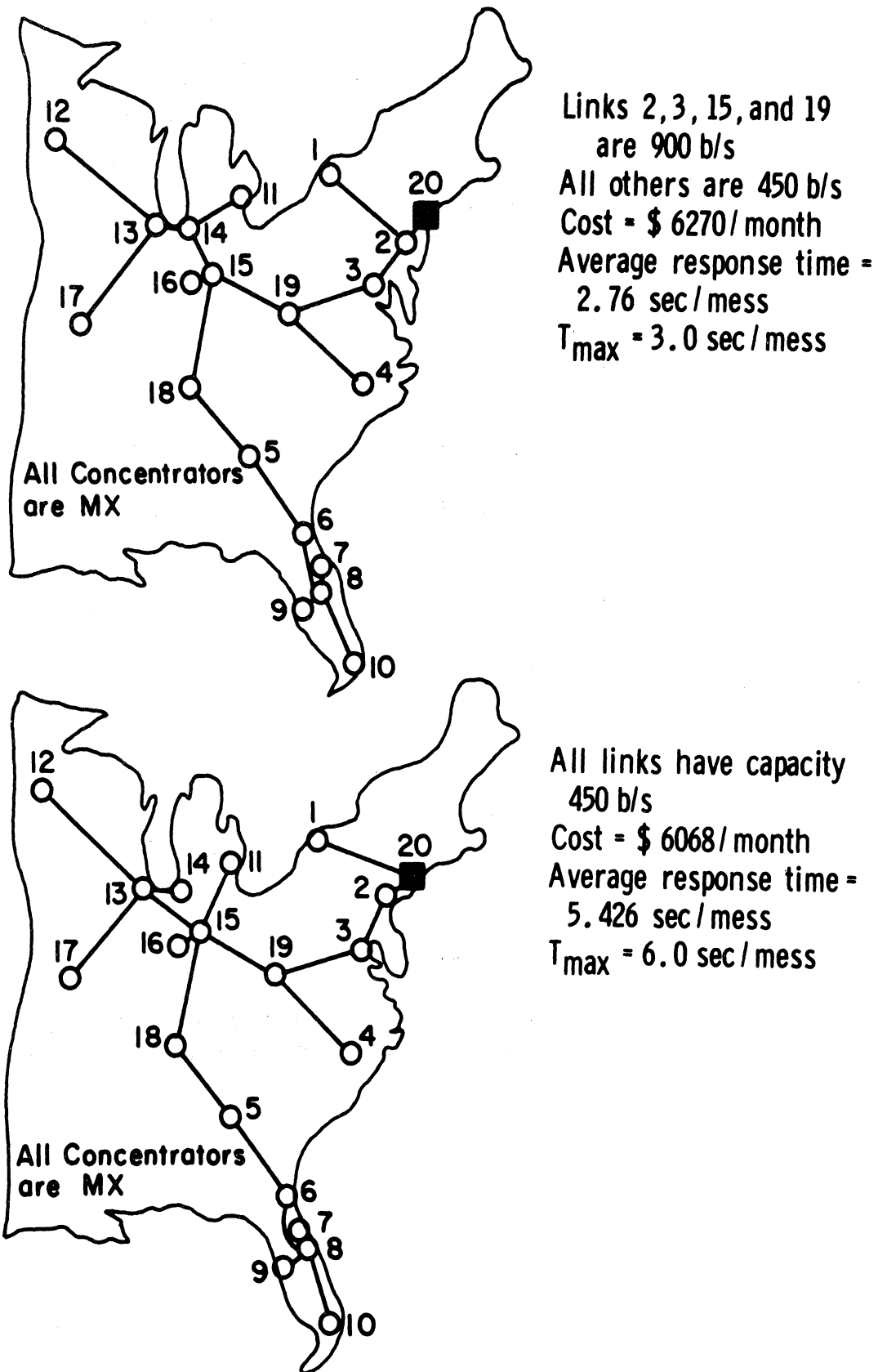


Figure 7.4 The Solution Configurations for the
Case Study of Section 7:4

Design Strategy	Cost of SOLN Config. \$/Mo.	Av. Response Time Sec/Mess
BDOLL, $T_{\max}=6.0$	A 6068	5.426
$T_{\max}=3.0$	B 6270	2.767
$T_{\max}=1.5$	C 6607	1.462
$T_{\max}=.875$	D 7009	.869
$T_{\max}=.75$	E 7276	.694
BEW(450)	F 6150	3.876
BEW(900)	G 7168	1.806
BEW(1350)	H 7762	1.177
BEW(1800)	I 8186	.873
BEW(2700)	J 8863	.869
BEW(3600)	K 9449	.647
BEW(5400)	10034	.428
BMST(450)	L 6185	6.095
BMST(900)	M 7189	2.767
BMST(1350)	N 7775	1.789
BMST(1800)	O 8194	1.322
BMST(2700)	J 8863	.869
BMST(3600)	K 9449	.647
BMST(5400)	10034	.428

Table 7-I

Experimental Results for the Case Study of Section 7.4. Points plotted on graph of Figure 7.5 are denoted by an alphanumeric preceding the cost.

Design Strategy	Cost of SOLN Config, \$/Mo.	Av. Resp. Time, Sec/Mess
EWED(900)	P 6462	1.806
EWED(1350)	Q 6835	1.177
EWED(1800)	R 7070	.873
EWED(2700)	S 7296	.869
EWED(3600)	T 7583	.647
EWED(5400)	8014	.428
MSTED(900)	U 6476	2.767
MSTED(1350)	V 6723	1.789
MSTED(1800)	W 7026	1.322
MSTED(2700)	S 7296	.869
MSTED(3600)	T 7583	.647
MSTED(5400)	8014	.428
EWMSC(450)	7050	2.073
EWMSC(900)	7456	1.028
EWMSC(1350)	8125	.645
MSTMSC(450)	X 7085	1.648
MSTMSC(900)	Y 7579	.801
MSTMSC(1350)	Z 7976	.529

Table 7- J

Experimental Results for the Case Study of Section 7.4. Points plotted on graph of Figure 7.5 are denoted by an alphanumeric preceding the cost.

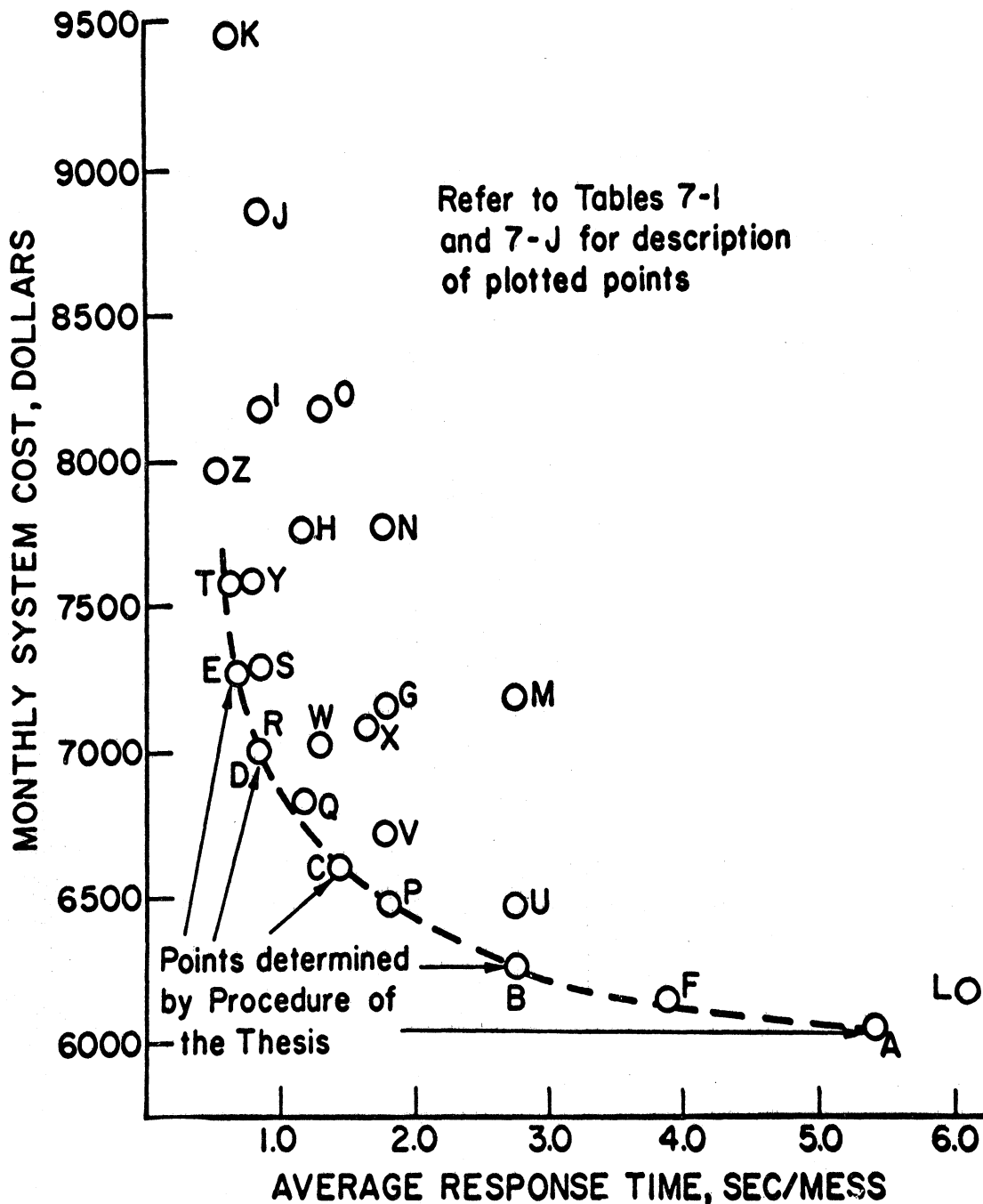


Figure 7.5 The Experimental Results for the Case Study of Section 7.4

and variants thereof do not consider the response time constraint as a variable in the problem specification.

7.4.1 Discussion of Results

This particular example illustrates a situation in which the solution configurations obtained for $T_{\max} = .75$ and for $T_{\max} = 1.5$ could be structurally altered to lower the cost (only slightly, however) without changing the average response time. Thus it is possible to manually improve upon the solutions obtained by the procedure, even though the cost reductions associated with such improvements turn out to be negligible with respect to the total cost.

Specifically, in each of the first two structures of Figure 7.4 consider only nodes 6, 7, 8, 9, and 10. The cost of this subnetwork in the first case is equal to \$620./month for lines + \$225./month for multiplexors, for a total of \$845./month. If link 6,7 were replaced by a link from node 6 to node 8 one of the multiplexors could be eliminated and the new cost would be \$681.50/month for lines + \$150. per month for the single multiplexor, for a total cost of \$831.50/month. The reduction in cost of \$13.50/month is negligible with respect to the total network cost of more than \$7250. per month.

Similarly, a negligibly small manual improvement of \$24.60/month could be obtained by making the same structural alteration to the network of Figure 7.4 which is obtained when $T_{\max} = 1.5$.

These anomalies usually arise only when terminals are clustered to the extent that incremental concentrator costs are similar in magnitude to line costs between terminals in a cluster. As the degree of clustering is reduced, the likelihood of finding such improvements in the suboptimal solutions of the design procedure will also be reduced. It does not appear that this limitation of the design procedure can be considered a significant one.

Attempts were made to use backtrack programming procedures to optimize the link capacity assignments for each solution configuration shown in Figure 7.4. When $T_{\max} > 3.0$, the $\{v_{i,h}^{\hat{}}\}$ solution was optimal since all capacities were 450 b/s. Five minutes of backtracking CPU time on the 360/67 failed to obtain a better solution than the $\{v_{i,h}^{\hat{}}\}$ for the cases where $T_{\max} \leq 3.0$. In examining the $\{v_{i,h}^{\hat{}}\}$ solutions in these cases, one concludes that

- a) The assignments do not appear to be obviously improveable, and
- b) Even if they were, any cost reductions would not likely be very significant.

We now conclude our experimental studies by considering a final design example which has been selected primarily to point out the limitations of the design procedure and to illustrate the importance of combining common sense with the available design tools. This next example will also illustrate a case in which a MSC is used in the solution configuration if the available capacity values are restricted to

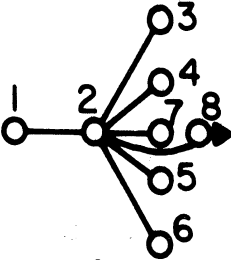
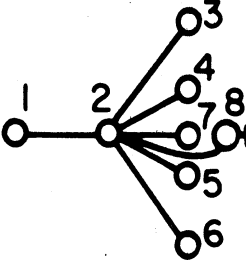
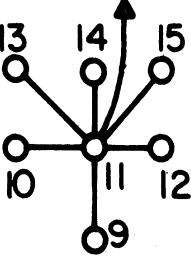
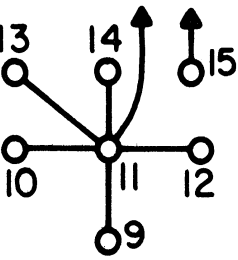
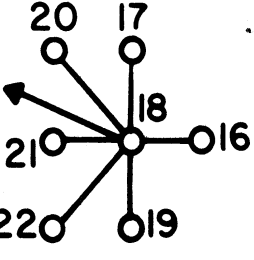
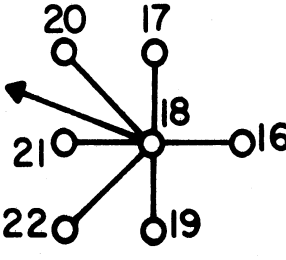
lower speed ranges.

7.5 A Highly Clustered Case Study

In this final example, we will illustrate the usage of the design procedure in an example where there are three clusters of remote terminals located relatively far from the central facility. We will also show that if only a restricted number of the lower speed capacity values from the set Φ can be used, then a MSC is used in the solution configuration.

Consider the network shown in Figure 7.6 where three clusters of remote terminals in New York, Houston, and San Francisco are to be connected to a central facility in Kansas City. The portion of the solution configuration which connects the clusters to the central facility is fairly easy to determine without the use of network analysis procedures. Since this problem has a solution which is fairly intuitive, a knowledgeable network designer would not likely spend much effort in determining the best topological structure. Nonetheless we have considered it since it sheds further light on how the procedure works and on its limitations.

Before proceeding further, one most important fact must be noted. Since the terminal distributions of Figure 7.6 are so highly clustered, it does not appear to be meaningful to average the message delays over the entire network. To put it another way, the problem should be treated as three independent subproblems since there is

$T_{max} = .5, 1.0, 4.$	$T_{max} = 2.0$	
 <p>Cluster A</p>	 <p>Cluster A</p>	<p>When $T_{max} = .5$ Links 2, 8 are 1800 b/s When $T_{max} = 1.0$ Links 2, 8 are 1350 b/s</p>
 <p>Cluster B</p>	 <p>Cluster B</p>	<p>When $T_{max} = .5$ Link 11 is 1800 b/s When $T_{max} = 1.0$ Link 11 is 1350 b/s</p>
 <p>Cluster C</p>	 <p>Cluster C</p>	<p>When $T_{max} = .5$ Link 18 is 1800 b/s When $T_{max} = 1.0$ Link 18 is 1350 b/s</p>

All capacities not otherwise specified are 450 b/s

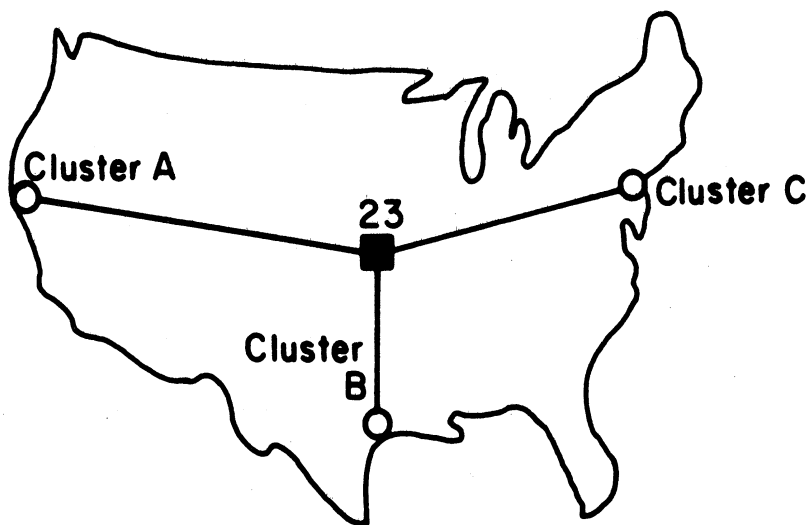


Figure 7.6 Solution Configurations for Clustered Terminal Case Study, Obtained Using Unmodified Procedure of Section 6.5

absolutely nothing to be gained by expanding the scope of the problem unnecessarily.

In order to illustrate what can happen if design procedures are used as a panacea, we have chosen, however, to pay no heed to the obvious fact that this problem should be analyzed as three separate subproblems. It will be shown that a fairly significant nonoptimality can arise as a consequence of the unintelligent usage of the design procedures. However, most of these anomalies would disappear, if the optimization procedures for assigning link capacity were used instead of the approximate $\{v_i, \hat{h}\}$ solutions for several of the structures in the sequence produced by the design procedure.

Again, it is assumed that a single buffered remote terminal is active at each of the remote terminal sites during the period of time considered. We will first consider the problem using the same seven discrete values of capacity as in the other case studies of this work. Then to illustrate an example of a solution configuration which uses a MSC, we treat the same design problem for a restricted number of lower speed capacity values - namely 450 b/s, 900 b/s, and 1350 b/s.

The arrivals are again assumed to be Poisson with mean rate of 1/30 mess/sec. The message length distributions are all assumed to be negative exponential with mean of 120 bits/mess. The distance matrix which was used is indicated in Table 7-K. Figure 7.6 depicts the solution configurations which were obtained by the unmodified design procedure of this work for various values of T_{\max} . In Tables 7-L

X \ Y	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	0.	33.4	70.7	52.7	52.7	70.7	66.7	100	1445	1454	1432	1410	1442	1420	1398	2725	2700	2700	2700	2675	2675	2675	2675	1250
2	0.	0.	52.7	23.5	23.5	52.7	33.3	66.6	1416	1425	1403	1382	1413	1391	1369	2691	2666	2666	2666	2641	2641	2641	2641	1216
3	0.	0.	0.	33.3	66.6	100	52.7	70.7	1376	1386	1364	1342	1375	1353	1330	2675	2650	2650	2650	2625	2625	2625	2625	1201
4	0.	0.	0.	0.	33.3	66.6	23.5	52.7	1393	1402	1380	1359	1390	1368	1346	2675	2650	2650	2650	2625	2625	2625	2625	1200
5	0.	0.	0.	0.	0.	33.3	23.5	52.7	1410	1419	1397	1376	1406	1385	1363	2675	2650	2650	2650	2625	2625	2625	2625	1201
6	0.	0.	0.	0.	0.	0.	52.7	70.7	1428	1436	1415	1393	1423	1402	1380	2675	2651	2650	2650	2625	2625	2625	2625	1201
7	0.	0.	0.	0.	0.	0.	0.	33.3	1387	1396	1374	1353	1384	1362	1340	2658	2633	2633	2633	2608	2608	2608	2608	1183
8	0.	0.	0.	0.	0.	0.	0.	0.	1359	1367	1346	1325	1355	1333	1311	2625	2600	2600	2600	2575	2575	2575	2575	1150
9	0.	0.	0.	0.	0.	0.	0.	0.	0.	35.3	25	35.3	55.9	50.	55.9	1643	1610	1621	1632	1587	1587	1598	1610	725
10	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	50	25	35.3	55.9	1610	1576	1587	1598	1554	1565	1576	1576	700
11	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	35.3	25	35.3	1632	1599	1610	1621	1576	1587	1598	1610	700
12	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	35.3	25	1655	1622	1632	1643	1599	1610	1621	1621	700
13	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	1599	1566	1576	1587	1543	1554	1565	1565	675
14	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	1622	1589	1599	1610	1566	1576	1587	1587	675
15	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1644	1611	1622	1632	1589	1599	1610	1610	675
16	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	35.3	25	35.3	55.9	50.0	55.9	1475	
17	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	35.3	25	35.3	1450
18	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	35.3	25	35.3	1450
19	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	35.3	25	1450
20	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	50	1425
21	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	1425
22	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	25	1425
23	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

Entries are

$$d_{X,Y} = d_{Y,X}$$

where $d_{X,Y}$ is the mileage from X to Y

Table 7 - K
Mileage Matrix for Example of Section 7. 5

Design Strategy	Cost of Solu. and Config.	Avg. R sponse Time Sec/Mess
BDOLL, $T_{\max} = 8.0$	A 6273	2.102
$T_{\max} = 4.0$	A 6273	2.102
$T_{\max} = 2.0$	B 7144	1.942
$T_{\max} = 1.4$	C 7283	1.015
$T_{\max} = 1.0$	D 7872	.670
$T_{\max} = .5$	E 8293	.499
BEW(450), BMST(450)	F 6705	2.102
BEW(900), BMST(900)	G 7831	1.015
BEW(1350), BMST(1350)	H 8488	.670
BEW(1800), BMST(1800)	I 8958	.499
BEW(2700), BMST(2700)	J 9708	.331
BEW(3600), BMST(3600)	10365	.247
BEW(5400), BMST(5400)	11022	.164
EWED(900), MSTED(900)	K 7712	1.015
EWED(1350), MSTED(1350)	L 8307	.670
EWED(1800), MSTED(1800)	M 8754	.499
EWED(2700), MSTED(2700)	N 9439	.331
EWED(3600), MSTED(3600)	10088	.247
EWED(5400), MSTED(5400)	10731	.164
EWMSC(450), MSTMSC(450)	P 7606	1.733
EWMSC(900), MSTMSC(900)	Q 8635	.838
EWMSC(1350), MSTMSC(1350)	R 9257	.553

Table 7- L

The Partial Experimental Results for the Clustered Example of Section 7. 5. Points plotted on the graph of Figure 7. 7 are denoted by an alphanumeric preceding the cost.

and 7-M, the experimental results for this case study are presented. For illustratory purposes, these points have also been plotted on the graph of Figure 7.7.

7.5.1 Discussion of Results

In cluster A, observe that a manual improvement to the solution structure could be made if the central link emanated from a multiplexor at node 7 and all other terminals were connected to node 7. The approximate savings associated with this operation would reduce the network cost by about .75% (a reasonably negligible amount) in the case where $T_{\max} = .5$.

Cluster B in the solution configuration when $T_{\max} = 2.0$ indicates that by using the basic procedure it is not possible to obtain the desired average response time with a single central link operating at 450 b/s. However, it can be seen that if other structures in the sequence of the multidimensional design parameter space were considered, it might be possible to improve the solution configuration appreciably.

This example points out the fact that the structure having the least costly $\{v_{i,h}\}$ solution is not always the most desirable. By applying the branch and bound technique to optimize the capacity assignments for several additional structures in the design parameter space, the improved solution of Figure 7.8 was obtained. The improvement associated with extending the basic procedure as suggested in Section 6.6 is quite significant in this particular case - a reduction from \$7283 per month to \$6483 per month.

Design Improvement Strategy	Index on Graph of Figure 7.7	Monthly Cost of Optimal Capacity Assignment in Improved Soln. Configuration .	Monthly Cost of Suboptimal Capacity Assign. (Diff. Structure)	Differences Between the Suboptimal and Improved Solution Configurations
<p>For the structure in the design parameter space S having the second least costly $\{v_i, h\}$ solution, use branch and bound to optimize the capacity assignments.</p>	<p>S</p>	<p>\$6483</p>	<p>\$7283</p>	<p>Contrast Figures 7.6 and 7.8 for the structural configurations associated with the $T_{\max} = 2$ constraint value.</p>

Note: The backtrack program found the optimal capacity assignment in the improved solution configuration with 13 sec of CPU execution time on the 360/67.

Table 7- M

The Improved Solution for the Design Problem of Section 7.5 when $T_{\max} = 2.0$

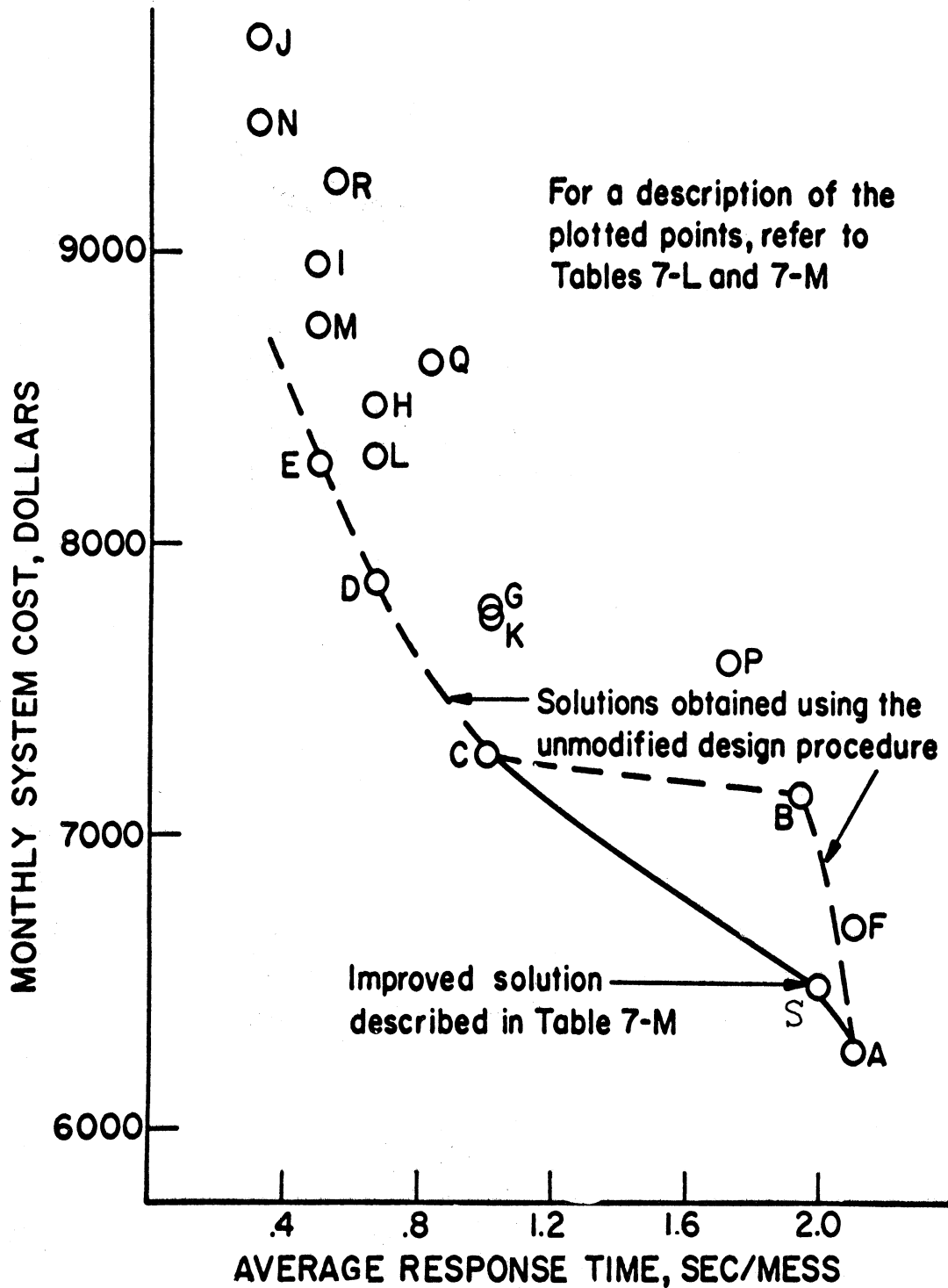


Figure 7.7 The Experimental Results for Clustered Terminal Case Study of Section 7.5

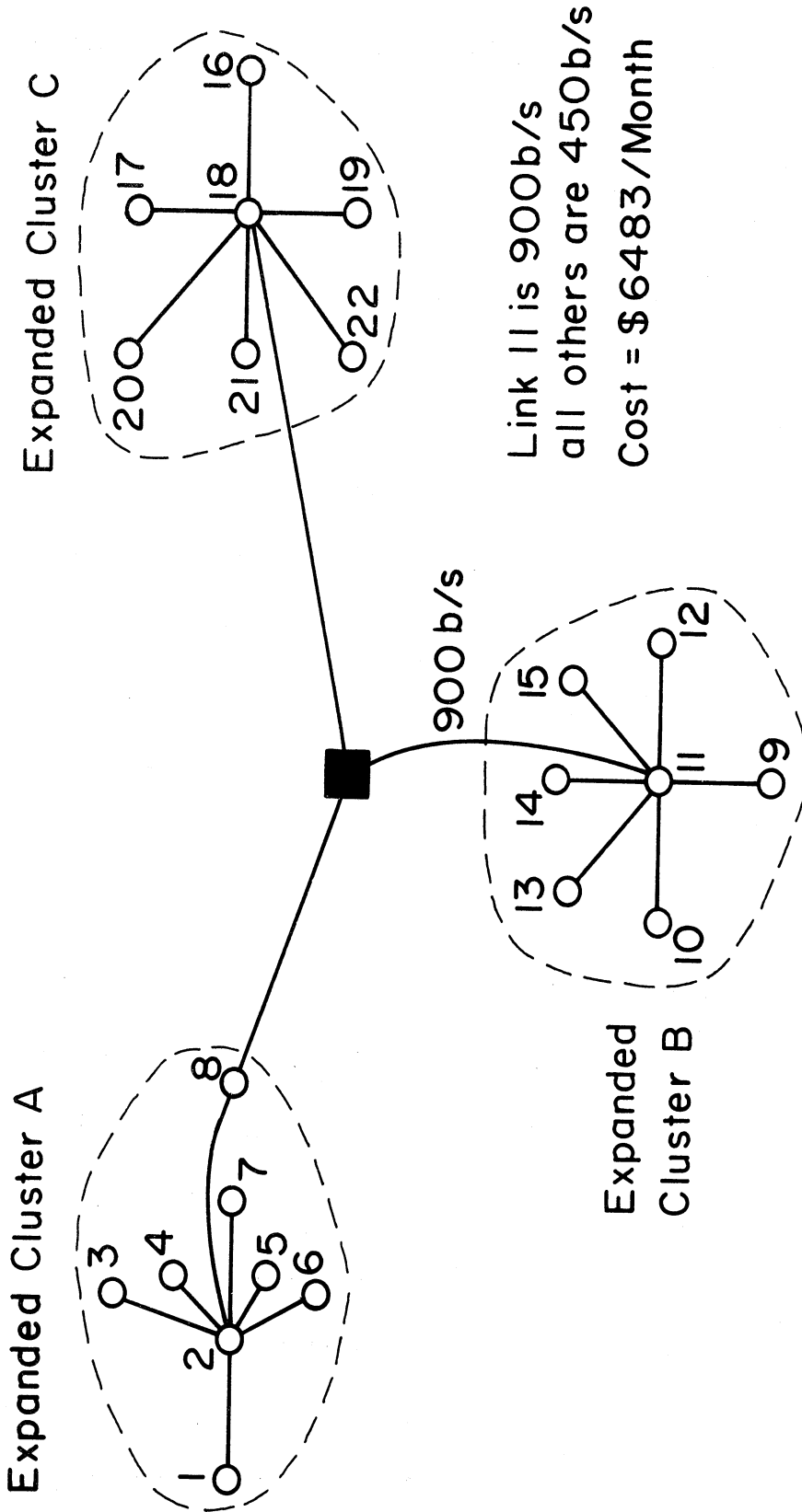


Figure 7.8 The improved Solution for the Design Problem of Section 7.5 when $T_{\max} = 2.0$

Attempts to improve the solutions obtained for other smaller values of T_{\max} failed in the allotted five minute execution time limit. No such obvious shortcomings of these solutions are apparent so the inability to improve upon them is not too surprising.

7.5.2 A Modification of This Problem

To further illustrate the use of the design procedure, we modify this problem slightly by restricting the set of available link capacity values to $\Phi = \{\Phi_1, \Phi_2, \Phi_3\}$ where $\Phi_i = 450 \times i$ for $i=1, 2, 3$.

The solution configuration obtained by the design procedure of this thesis when $T_{\max} = .5$ is shown in Figure 7.9, where a MSC is now used at node 2. Some manual improvements can be made to the link capacity assignments and topological structure of the solution obtained by the unmodified design procedure of this work. For example, the capacity values for links connecting terminals in the cluster A to the MSC at node 2 could be reduced from 1350 b/s to 450 b/s. It would also be possible to eliminate the multiplexor at node 8 and let the 1350 b/s central link from the "A" cluster emanate from node 2 instead. The total percentage cost reduction which would result from these manual improvements is approximately 1.5%.

Of course, the capacity allocation optimization procedures of Section 5.3 could be attempted in the previously discussed manner to possibly eliminate this particular source of nonoptimality altogether. Due to the artificial nature of this special example, however, no attempts were made to consider these contingencies further.

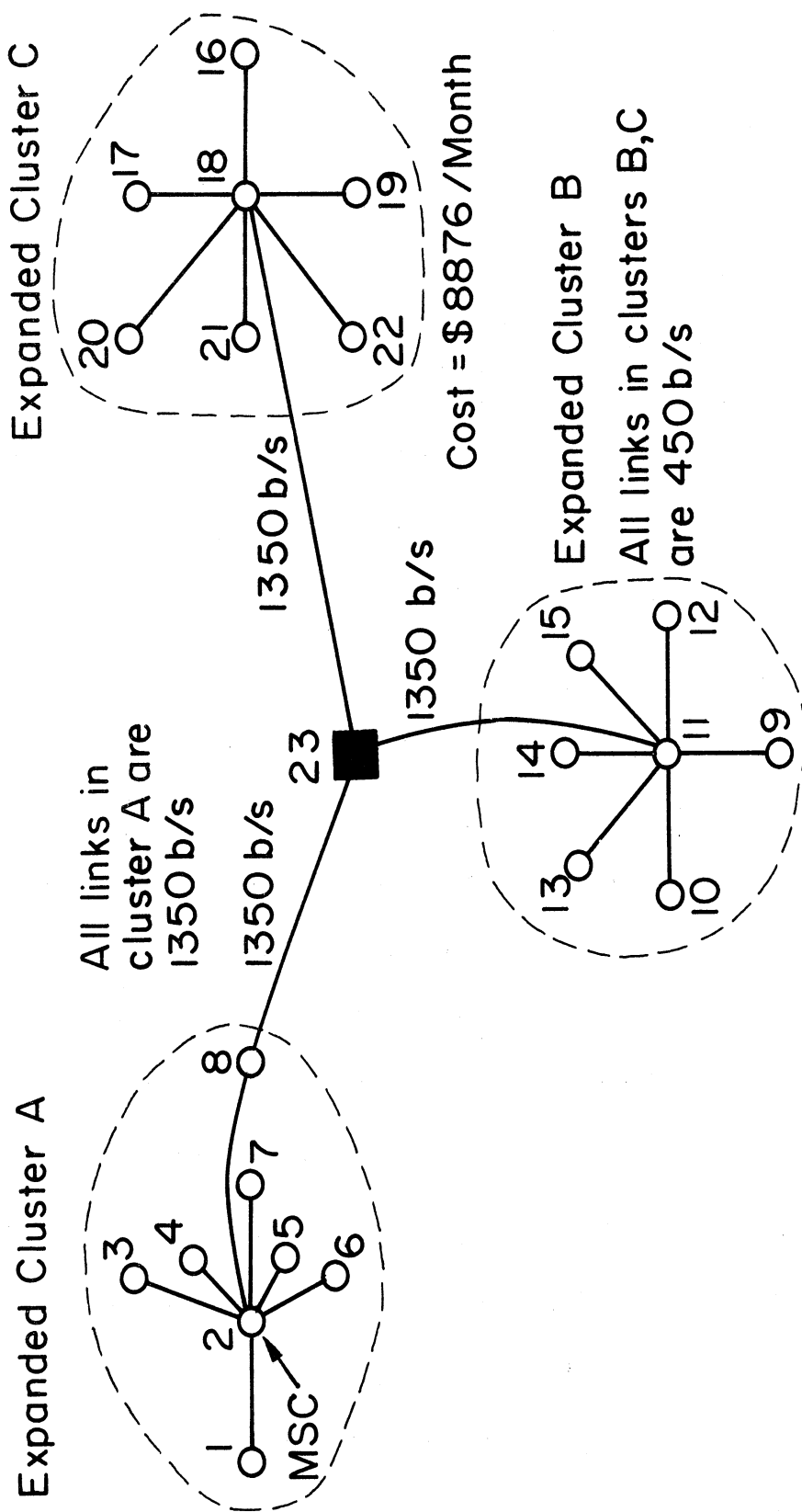


Figure 7.9 The Solution for the Modified Design Problem of Section 7.5.1 when $T_{max} = .5$

Chapter VIII

SUMMARY

8.1 Recapitulation

The objective of this work has been to formulate and solve a variety of resource allocation problems which arise in the design of centralized computer-communication networks. Since a large portion of the capital outlays in the teleprocessing systems are frequently devoted to the remote communications facilities, these problems are every bit as important as those concerned with efficiently utilizing the facilities at the central location. We have endeavored to develop realistic analytic models of the telecommunication network behavior which enable us to solve these various design problems with algorithms that are as computationally efficient as is possible. In some cases, it has been necessary to consider procedures which obtain solutions that are reasonable approximations to the true optima and which, at the same time, assure tractability.

In Chapter II, we commenced these efforts with a discussion of the important macroscopic level design variables and the design trade-offs which must be considered in centralized networks. A relatively simple-to-work-with, yet realistic performance measure, average response time, was precisely defined. The assumptions used to develop the problem solving models of later chapters were then presented, justified, and shown to generally provide a greater level of model detail than existing approaches.

Then in Chapter III we formulated queueing models for the various types of data concentrators. We developed an optimal passage capacity assignment procedure for determining the capacity in multiplexed communication links so as to obtain the minimum average message delay throughout the entire network. Simulation studies revealed the efficacy of making the infinite-buffer assumption for MSC concentrators.

As a prelude to solving the general network design problem, we devoted Chapter IV to rigorously analyzing existing design procedures and establishing realistic cost functions. Then in Chapter V, it was seen that usage of these approaches generally leads to an inefficient utilization of the link capacity resource since it is not treated as a variable. We presented methods for optimally allocating link capacity in a network so as to minimize line costs and satisfy a performance constraint.

All of the efforts of the first five chapters were then, in Chapter VI, integrated and used to develop a comprehensive, yet realistic model for designing centralized computer-communication networks. First, a basic algorithm was presented which can be effectively used in n terminal networks as long as n^3 sets of design calculations can be performed on the computer in a reasonable length of time. The solutions obtained using this procedure, even though consistently better than with existing design techniques, could sometimes, however, be improved using certain extensions which were then presented.

To cite the usefulness and limitations of the design procedures, and to compare them with other methods, a variety of representative case studies were considered in Chapter 7. It was interesting to note that in none of the examples considered did any of the other design procedures produce better solutions than those obtained with procedures proposed in this research.

All of these efforts have led to a number of useful conclusions. Most of these have been discussed as the particular problems were considered. A precise listing of the major contributions of this work was also presented in Section 1.7. Nonetheless, there are several important macroscopic level conclusions which should be noted at this point, in the interest of completeness.

1. The topological structure of the solution configurations obtained from the design procedure does not change significantly over the relatively wide range of T_{\max} considered. However, the link capacity values and costs for these configurations are quite sensitive to variations in T_{\max} .
2. The topological structures obtained by the procedure are consistently similar to those obtained using the minimum spanning tree algorithm, particularly when the constraints are loose, say for $T_{\max} \geq 3.0$. However in networks more highly congested than those examined in this work, this similarity is not likely to be so pronounced.
3. With present hardware technology, the MSC concentrators are not cost justifiable, except at the most heavily congested concentration

points and when the response time constraints are fairly tight, ($T_{\max} \leq 1.0$). As MSC costs come down, the design model will make greater usage of MSC's in the solution configurations.

4. To accomplish the most efficient network design, it is essential that different transmission capacity values be available for use in the different links of the network. Single link capacity network designs are simply not satisfactory.
5. The combinatorial complexities of the most general network design problem assuredly account for the paucity of previous work in this area. This thesis demonstrates the efficacy of using reasonable suboptimal methods to solve otherwise untractable problems. The resulting payoffs which have been shown justify this philosophy for resolving an apparent impasse between the conflicting forces of computational efficiency and mathematical rigor.

8.2 Future Research

We have chosen to design and analyze centralized communication networks with a model which is obviously an idealization of a real life system. It would be desirable to relax certain of the assumptions considered in Section 2.3, and still have a tractable problem. Unfortunately, however, most of the assumptions have been selected to guarantee computational feasibility so that their relaxation may lead to difficulties. Any efforts which enable the design problems of this work to be considered for less restrictive sets of assumptions would be contributory. Some of these possibilities are now presented.

1. It would be beneficial to be able to consider higher order moments of performance, or alternate measures altogether.
2. The non-idealities resulting from different statistical distributions for the traffic arrivals should be investigated.
3. The effects of alternate control procedures such as polling with short message lengths should be explored.
4. Other types of reliability should be investigated, such as the affects of total link failures, concentrator outages, etc..
5. The problem should be considered for other types of costing strategies (e. g. where the facilities are paid for on an as-used basis , or where a communication link cost is not a linear function of distance).

Other possibilities should include the investigation of alternate heuristic schemes and the development of bounds for estimating the accuracy of these heuristics.

Clearly these efforts are all concerned with the centralized network type of structure. An obvious area for further research is the distributed network design problem. Using superposition-like approaches to solve several constituent centralized problems is an area which should be investigated. However, additional degrees of computational complexity will again be encountered as multiple processing and/or data base locations are considered.

BIBLIOGRAPHY

1. J. J. Kucera, "Transfer Rate of Information Bits," Computer Design, Vol. 7, No. 6, June, 1968.
2. S. J. Kaplan, "The Advancing Communication Technology and Computer Communications Systems," AFIPS Conference Proceedings, Spring Joint Computer Conference, 1968.
3. J. B. Dennis, "A Position Paper on Computing and Communications," Communications of the ACM, Vol. 11, No. 5, May, 1968.
4. D. F. Parkhill, The Challenge of the Computer Utility, Addison Wesley Publishing Co., Reading, Mass., 1966.
5. E. Brockmeyer, H. L. Halstrom, and A. Jensen, "The Life and Works of A. K. Erlang," Trans. of the Danish Academy of Technical Sciences, No. 2, 1948.
6. E. C. Molina, "The Theory of Probabilities Applied to Telephone Trunking Problems," Bell System Tech. Journal, Vol. 1, No. 2, pp. 69-81, 1922.
7. G. F. O'Dell, "An Outline of the Trunking Aspect of Automatic Telephony," Journal of the Inst. Elec. Engrs., Vol. 65, pp. 185-222, 1927.
8. T. C. Fry, Probability and Its Engineering Uses, D. Van Nostrand Company, Inc., Princeton, N. J., 1928.
9. W. Feller, "Die Grundlagen der Volterraschen Theorie des Kampfes ums Dasein in Wahrscheinlichkeitstheoretischer Behandlung," Acta Biotheoret., Vol. 5, pp. 11-40, 1939.
10. W. Feller, An Introduction to Probability Theory and Its Applications, John Wiley and Sons, Inc., New York, 1957.
11. D. G. Kendall, "Some Problems in the Theory of Queues," J. Royal Statistical Society, Series B, Vol. 13, pp. 151-185, 1951.
12. F. Pollaczek, "Problemes stochastiques poses par le phenomene de formation d'une queue d'attente à un guichet et par des phenomenes apparentes," Mem. Sci. Math, Paris, No. 136, 1957.
13. P. M. Morse, "Stochastic Properties of Waiting Lines," Operations Research, Vol. 3, pp. 255-261, 1955.

14. A. J. Khinchine, Mathematical Methods in the Theory of Queueing, Charles Griffin and Co. , London, 1960.
15. R. Syski, Introduction to Congestion in Telephone Systems, Oliver and Boyd, London, 1960.
16. V. Benes, Mathematical Theory of Connecting Networks and Telephone Traffic, Academic Press, New York, 1965.
17. T. L. Saaty, Elements of Queueing Theory with Applications, McGraw Hill Book Co. , New York, 1961.
18. D. R. Cox and W. L. Smith, Queues, Methuen and Co. , London, and John Wiley and Sons, Inc. , New York, 1961.
19. P. J. Burke, "The Output of a Queueing System," Operations Research, Vol. 4, pp. 699-704, 1956.
20. J. Riordan, Stochastic Service Systems, John Wiley and Sons, Inc. , New York, 1961.
21. M. Pollack, "A Bibliography of Communication Network Studies," IEEE Trans. on Communication Technology, Vol. COM-13, No. 4, pp. 552-554, December, 1965.
22. L. R. Ford, Jr. and D. R. Fulkerson, Flows in Networks, Princeton U. Press, Princeton, N. J. , 1962.
23. R. Busacker and T. L. Saaty, Finite Graphs and Networks, An Introduction with Applications, McGraw Hill Book Co. , New York, 1965.
24. W. H. Kim and R. T. Chien, Topological Analysis and Synthesis of Communication Networks , Columbia U. Press, New York, 1962.
25. L. Kleinrock, Communication Nets Stochastic Message Flow and Delay, Lincoln Laboratory Publications, McGraw Hill Book Co. , New York, 1964.
26. R. E. Gomory and T. C. Hu, "Multi-terminal Network Flows," Journal of the Society of Industrial and Applied Mathematics, Vol. 9, No. 4, pp. 551-570, 1961.
27. R. E. Gomory and T. C. Hu, "Multi-commodity Network Flows," IBM research report RC-865, January, 1963.

28. W. Mayeda, "Terminal and Branch Capacity Matrices of a Communication Net," IRE Trans. on Circuit Theory, Vol. 7, pp. 251-269, 1960.
29. W. S. Jewell, "Optimal Flow Through a Network," M. I. T. doctorate thesis, Elect. Eng. Dept., 1958.
30. S. L. Hakimi, "On Simultaneous Flows in a Communication Network," IRE Trans. on Circuit Theory, Vol. CT-9, No. 2, pp. 169-175, 1962.
31. G. C. Hunt, "Sequential Arrays of Waiting Lines," Operations Research, Vol. 4, pp. 674-683, 1956.
32. J. R. Jackson, "Networks of Waiting Lines," Operations Research, Vol. 5, pp. 518-521, 1957.
33. P. Moran, The Theory of Storage, Methuen and Co., London, and John Wiley and Sons, Inc., New York, 1959.
34. K. Onaga, "Stochastic Flow of Messages and Its Efficient Transmission through Communication Nets," Report R-269, Coord. Science Lab., U. of Illinois, November, 1965.
35. R. T. Prosser, "Routing Procedures in Communications Networks, part I: Random Procedures," IRE Trans. on Communication Systems, Vol. CS-10, No. 4, pp. 322-329, 1962.
36. R. T. Prosser, "Routing Procedures in Communications Networks, part II: Directory Procedures," IRE Trans. on Communication Systems, Vol. CS-10, No. 4, pp. 329-335, 1962.
37. C. F. Haugh and M. A. Berk, "Matching Communications Facilities to Data Processors," IEEE Trans. on Comm. and Electronics, Vol. 82, No. 67, pp. 427-429, July, 1963.
38. G. Harrison, "Message Buffering in a Computer Switching Center," IEEE Trans. on Comm. and Electronics, Vol. 82, pp. 532-534, September, 1963.
39. E. U. Cohler and H. Rubinstein. "A Multicomputer Message Switching Data Processing System," IEEE Trans. on Com. Technology, Vol. COM-15, No. 3, June, 1967.
40. A. Weingarten, "Storage Requirements for a Message Switching Computer," IEEE Trans. on Com. Systems, Vol. CS-12, pp. 191-195, June, 1964.

41. N. Saber and H. Young, "The Relative Effectiveness of Assigned and Shared Storage in a Switching System," IEEE Trans. on Comm. Tech., Vol. COM-14, No. 6, pp. 756-762, Dec., 1966.
42. S. Stimler and K. A. Brons, "A Methodology for Calculating and Optimizing Real-Time System Performance," Comm. Assoc. Comp. Machinery, Vol. 11, No. 7, pp. 509-516, July, 1968.
43. J. T. Martin, Programming Real-Time Computer Systems, Prentice Hall, Inc. Englewood Cliffs, N. J., 1965.
44. J. T. Martin, Design of Real-Time Computer Systems, Prentice Hall, Inc., Englewood Cliffs, N.J., 1967.
45. W. J. Karplus, On-Line Computing, McGraw Hill Book Co., New York, 1967.
46. A. J. Weber, Record of the Eleventh National Communications Symposium, 23-26, October, 1965.
47. W. H. Desmonde, Real-Time Data Processing Systems, Introductory Concepts, Prentice Hall, Inc., Englewood Cliffs, N. J., 1964.
48. W. P. Margopoulos and R. J. Williams, "On Teleprocessing System Design, Part I, Characteristic Problems," IBM Systems Journal, Vol. 5, No. 3, 1966.
49. J. P. Bricault and I. Delgalvis, "On Teleprocessing System Design, Part III, An Analysis of a request-queued buffer pool," IBM Systems Journal, Vol. 5, No. 3, 1966.
50. T. W. Gay, Jr., "On Teleprocessing System Design, Part V, A technique for estimating channel interference," IBM Systems Journal, Vol. 5, No. 3, 1966.
51. W. Chang and D. J. Wong, "Computer Channel Interference Analysis," IBM Systems Journal, Vol. 4, No. 2, 1965.
52. P. H. Seaman, R. A. Lind, and T. L. Wilson, "On Teleprocessing System Design, Part IV, An analysis of auxiliary storage activity," IBM Systems Journal, Vol. 5, No. 3, 1966.
53. L. R. Esau and K. C. Williams, "On Teleprocessing System Design, Part II, A method for approximating the optimal network," IBM Systems Journal, Vol. 5, No. 3, 1966.

54. E. G. Coffman, "Stochastic Models of Multiple and Time-Shared Computer Operations," UCLA Dept. of Elect. Eng., Report No. 66-38, June, 1966.
55. B. W. Arden, "Time Sharing Systems, A Review," IEEE International Convention Record, Part 10, March, 1967.
56. V. L. Wallace and R. S. Rosenberg, "RQA-1, The Recursive Queue Analyzer," Report No. 07742-1-T, Systems Eng. Lab., University of Michigan, 1966.
57. N. R. Nielsen, "The Analysis of General Purpose Computer Time-Sharing Systems," Document No. 40-10-1, Stanford U. Computation Center, 1966.
58. T. B. Pinkerton, "Program Behavior and Control in Virtual Storage Computer Systems," Technical Report 4, CONCOMP, The University of Michigan, April, 1968.
59. J. Riordan, An Introduction to Combinatorial Analysis, John Wiley and Sons Inc., New York, 1958.
60. M. Held and R. M. Karp, "The construction of discrete dynamic programming algorithms," IBM Systems Journal, Vol. 4, No. 2, 1965.
61. R. A. Totschek, "An Empirical Investigation into the Behavior of the SDC Time-Sharing System," System Development Corp. Report SP-2191/000/00, August, 1965.
62. P. Morse, Queues, Inventory, and Maintenance, John Wiley and Sons Inc., New York, 1963.
63. A. L. Scherr, "An Analysis of Time Shared Computer Systems," Report MAC-TR-18, Mass. Inst. of Tech., Cambridge, Mass., 1965.
64. W. Erikson, "An Analytical Cost Comparison of Computer Operating Systems," Technical Memo. TM-3525, System Development Corp., Santa Monica, June, 1967.
65. L. E. Schrage, "Some Queueing Models for a Time-Shared Facility," Ph. D. Thesis, Cornell Univ., February, 1966.
66. J. L. Smith, "An Analysis of Time-Sharing Computer Systems Using Markov Models," AFIPS Conference Proceedings, Spring Joint Computer Conference, 1966.

67. D. W. Fife, "An Optimization Model for Time Sharing," AFIPS Conference Proceedings, Spring Joint Computer Conference, 1966.
68. G. E. Bryan, "JOSS: 20,000 Hours at the Console - A Statistical Summary," AFIPS Conference Proceedings, Fall Joint Computer Conference, 1967.
69. E. G. Coffman and R. C. Wood, "Interarrival Statistics for TSS," System Development Corp. Report No. SP-2161, August, 1965.
70. Analysis of Some Queueing Models in Real-Time Systems, IBM Manual No. F20-0007-0. Published by the IBM Corp., Poughkeepsie, New York, 1965.
71. Analysis of the 7740-7010 Teleprocessing System Workload, Auburn University, December, 1967.
72. R. L. Snyder, Comments on "The Relative Effectiveness of Assigned and Shared Storage in a Switching System" (Correspondence), IEEE Transactions on Communication Technology, Vol. CT-15, No. 3, June, 1967.
73. P. Bello, "Time-Frequency Duality," IEEE Trans. on Information Theory, Vol. IT-10, January, 1964.
74. F. B. Hildebrand, Methods of Applied Mathematics, Prentice Hall, Inc., Englewood Cliffs, N. J., 1958.
75. "A List of Rate Centers and Control Offices," Tariff FCC #255, filed by the American Telephone and Telegraph Co.
76. J. B. Kruskal, Jr., "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," Proc. Am. Math. Society, Vol. 7, pp. 48-50, 1956.
77. R. C. Prim, "Shortest Connection Networks and Some Generalizations," Bell System Technical Journal, Vol. 36, pp. 1389-1401, November, 1957.
78. "Synchronous Signaling Rates for Data Transmission," Comm. of the A. C. M., Vol. 11, No. 8, August, 1968.
79. "Introduction to Data Communications" a publication of the Digital Equipment Corp., Maynard, Mass., 1968.
80. E. L. Lawler and D. E. Wood, "Branch and Bound Methods, a Survey," Operations Research, Vol. 14, No. 4, July-August, 1966.

81. S. W. Golomb and L. D. Baumert, "Backtrack Programming," Jour. Assoc. Comp. Machinery, Vol. 12, No. 4, October, 1965.
82. L. J. White, "A Parametric Study of Matchings and Coverings in Weighted Graphs," Report No. 06920-11-T, Systems Eng. Laboratory, University of Michigan, May, 1967.

APPENDIX A

This appendix contains listings of the four GPSS/360 programs which were used to obtain the results shown in Tables 3-A and 3-B. The first program is the simulation run made for a system (See Figure 3.3) having a line speed of 450 b/s and mean arrival rate of $\lambda = 1$ mess/sec. The second program is for the same 450 b/s line speed but with $\lambda = 3$ mess/sec. The third and fourth programs are for a line speed of 900 b/s with values of $\lambda = 1$ mess/sec and $\lambda = 3$ mess/sec, respectively.


```

1 SIMULATE
2 1 FUNCTION RN1,C32
3 0.,0./0.048771,.05/.095163,.1/.139292,.15/.181269,.2/.221199,.25
4 .259182,.3/.295312,.35/.329680,.4/.393469,.5/.451188,.6/.503415,.7
5 .550671,.8/.593430,.9/.632121,1./698806,1.2/.753403,1.4/.798103,1.6
6 .834701,1.8/.864665,2./899741,2.3/.917915,2.5/.950213,3./969803,3.5
7 .981684,4.0/.988891,4.5/.993262,5./9975212,6./9990881,7./9996645,8.
8 .9998766,9./9999546,10.
9 2 FUNCTION RN2,C32
10 0.,0./0.048771,.05/.095163,.1/.139292,.15/.181269,.2/.221199,.25
11 .259182,.3/.295312,.35/.329680,.4/.393469,.5/.451188,.6/.503415,.7
12 .550671,.8/.593430,.9/.632121,1./698806,1.2/.753403,1.4/.798103,1.6
13 .834701,1.8/.864665,2./899741,2.3/.917915,2.5/.950213,3./969803,3.5
14 .981684,4.0/.988891,4.5/.993262,5./9975212,6./9990881,7./9996645,8.
15 .9998766,9./9999546,10.
16 ** FVARIABLE 1 CORRESPONDS TO MEAN MESSAGE LENGTH OF 120 BITS
17 1 FVARIABLE 120*FN2
18 2 FVARIABLE (1000/450)*P1
19 STORAGE S1,4000
20 GENERATE 1000, FN1,,,,F
21 ASSIGN 1,V1
22 ASSIGN 3,R1
23 TEST G P1,P3,INPUT
24 ASSIGN 2,P1
25 ASSIGN 2-,P3
26 *** P2 =AMOUNT OF STORAGE OVERFLOW
27 ENTER 2,P2
28 LEAVE 2,P2
29 TERMINATE
30 INPUT QUEUE 1
31 ENTER 1,P1 NODAL STORAGE
32 SEIZE 1
33 ADVANCE V2
34 RELEASE 1
35 LEAVE 1,P1 DEPART FROM STORAGE
36 DEPART 1
37 TERMINATE 1
38 START 16000
39 END

```

```

40 SIMULATE
41 1 FUNCTION RN1,C32
42 0.,0./0.048771,.05/.095163,.1/.139292,.15/.181269,.2/.221199,.25
43 .259182,.3/.295312,.35/.329680,.4/.393469,.5/.451188,.6/.503415,.7
44 .550671,.8/.593430,.9/.632121,1./698806,1.2/.753403,1.4/.798103,1.6
45 .834701,1.8/.864665,2./899741,2.3/.917915,2.5/.950213,3./969803,3.5
46 .981684,4.0/.988891,4.5/.993262,5./9975212,6./9990881,7./9996645,8.
47 .9998766,9./9999546,10.
48 2 FUNCTION RN2,C32
49 0.,0./0.048771,.05/.095163,.1/.139292,.15/.181269,.2/.221199,.25
50 .259182,.3/.295312,.35/.329680,.4/.393469,.5/.451188,.6/.503415,.7
51 .550671,.8/.593430,.9/.632121,1./698806,1.2/.753403,1.4/.798103,1.6
52 .834701,1.8/.864665,2./899741,2.3/.917915,2.5/.950213,3./969803,3.5
53 .981684,4.0/.988891,4.5/.993262,5./9975212,6./9990881,7./9996645,8.
54 .9998766,9./9999546,10.
55 ** FVARIABLE 1 CORRESPONDS TO MEAN MESSAGE LENGTH OF 120 BITS
56 1 FVARIABLE 120*FN2
57 2 FVARIABLE (1000/45)*PI
58 STORAGE S1,4000
59 GENERATE 3333,FN1,,,,,F
60 ASSIGN 1,V1
61 ASSIGN 3,R1
62 TEST G P1,P3,INPUT
63 ASSIGN 2,P1
64 ASSIGN 2-,P3
65 *** P2 =AMOUNT OF STORAGE OVERFLOW
66 ENTER 2,P2
67 LEAVE 2,P2
68 TERMINATE
69 INPUT QUEUE 1
70 ENTER 1,P1 NODAL STORAGE
71 SEIZE 1
72 ADVANCE V2
73 RELEASE 1
74 LEAVE 1,P1 DEPART FROM STORAGE
75 DEPART 1
76 TERMINATE 1
77 START 16000
78 END

```

```

79 SIMULATE
80 1 FUNCTION RN1,C32
81 0.,0./0.048771,.05/.095163,.1/.139292,.15/.181269,.2/.221199,.25
82 .259182,.3/.295312,.35/.329680,.4/.393469,.5/.451188,.6/.503415,.7
83 .550671,.8/.593430,.9/.632121,1./0.698806,1.2/.753403,1.4/.798103,1.6
84 .834701,1.8/.864665,2./0.899741,2.3/.917915,2.5/.950213,3./0.969803,3.5
85 .981684,4.0/.988891,4.5/.993262,5./0.9975212,6./0.9990881,7./0.9996645,8.
86 .9998766,9./0.9999546,10.
87 2 FUNCTION RN2,C32
88 0.,0./0.048771,.05/.095163,.1/.139292,.15/.181269,.2/.221199,.25
89 .259182,.3/.295312,.35/.329680,.4/.393469,.5/.451188,.6/.503415,.7
90 .550671,.8/.593430,.9/.632121,1./0.698806,1.2/.753403,1.4/.798103,1.6
91 .834701,1.8/.864665,2./0.899741,2.3/.917915,2.5/.950213,3./0.969803,3.5
92 .981684,4.0/.988891,4.5/.993262,5./0.9975212,6./0.9990881,7./0.9996645,8.
93 .9998766,9./0.9999546,10.
94 ** FVARIABLE 1 CORRESPONDS TO MEAN MESSAGE LENGTH OF 120 BITS
95 1 FVARIABLE 120*FN2
96 2 FVARIABLE (1000/900)*P1
97 STORAGE S1,4000
98 GENERATE 1000,FN1,,,,F
99 ASSIGN 1,V1
100 ASSIGN 3,R1
101 TEST G P1,P3,INPUT
102 ASSIGN 2,P1
103 ASSIGN 2-,P3
104 *** P2 =AMOUNT OF STORAGE OVERFLOW
105 ENTER 2,P2
106 LEAVE 2,P2
107 TERMINATE
108 INPUT QUEUE 1
109 ENTER 1,P1 NODAL STORAGE
110 SEIZE 1
111 ADVANCE V2
112 RELEASE 1
113 LEAVE 1,P1 DEPART FROM STORAGE
114 DEPART 1
115 TERMINATE 1
116 START 16000
117 END

```

```

118 SIMULATE
119 FUNCTION RN1,C32
120 0.,0./,048771,,05/.095163,,1/.139292,,15/.181269,,2/.221199,,25
121 .259182,,3/.295312,,35/.329680,,4/.393469,,5/.451188,,6/.503415,,7
122 .550671,,8/.593430,,9/.632121,,1./,698806,1.2/.753403,1.4/.798103,1.6
123 .834701,1.8/.864665,2./,899741,2.3/.917915,2.5/.950213,3./,969803,3.5
124 .981684,4.0/.988891,4.5/.993262,5./,9975212,6./,9990881,7./,9996645,8.
125 .9998766,9./,9999546,10.
126 FUNCTION RN2,C32
127 0.,0./,048771,,05/.095163,,1/.139292,,15/.181269,,2/.221199,,25
128 .259182,,3/.295312,,35/.329680,,4/.393469,,5/.451188,,6/.503415,,7
129 .550671,,8/.593430,,9/.632121,1./,698806,1.2/.753403,1.4/.798103,1.6
130 .834701,1.8/.864665,2./,899741,2.3/.917915,2.5/.950213,3./,969803,3.5
131 .981684,4.0/.988891,4.5/.993262,5./,9975212,6./,9990881,7./,9996645,8.
132 .9998766,9./,9999546,10.
133 ** FVARIABLE I CORRESPONDS TO MEAN MESSAGE LENGTH OF 120 BITS
134 1 FVARIABLE 120*FN2
135 2 FVARIABLE (1000/90)*PI
136 STORAGE SI,4000
137 GENERATE 3333, FN1,,,,F
138 ASSIGN 1,VI
139 ASSIGN 3,RI
140 TEST G PI,P3,INPUT
141 ASSIGN 2,PI
142 ASSIGN 2-,P3
143 *** P2 =AMOUNT OF STORAGE OVERFLOW
144 ENTER 2,P2
145 LEAVE 2,P2
146 TERMINATE
147 INPUT QUEUE I
148 ENTER 1,P1 NODAL STORAGE
149 SEIZE 1
150 ADVANCE V2
151 RELEASE 1
152 LEAVE 1,P1 DEPART FROM STORAGE
153 DEPART 1
154 TERMINATE I
155 START 16000
156 END

```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D		
<i>(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)</i>		
1. ORIGINATING ACTIVITY (Corporate author) University of Michigan - Electrical Engineering Department, Systems Engineering Laboratory Ann Arbor, Michigan		2a. REPORT SECURITY CLASSIFICATION Unclassified
		2b. GROUP
3. REPORT TITLE EFFICIENT ALLOCATION OF RESOURCES IN CENTRALIZED COMPUTER-COMMUNICATION NETWORK DESIGN		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Interim Technical Report		
5. AUTHOR(S) (First name, middle initial, last name) Dixon R. Doll		
6. REPORT DATE November 1969	7a. TOTAL NO. OF PAGES 344	7b. NO. OF REFS 82
8a. CONTRACT OR GRANT NO. F30602-69-C-0214	9a. ORIGINATOR'S REPORT NUMBER(S) SEL-TR-36 02641-1-T	
b. PROJECT NO. 5581		
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.	RADC-TR-69-305	
10. DISTRIBUTION STATEMENT This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of RADC (EMBIH) GAFB, N.Y. 13440.		
11. SUPPLEMENTARY NOTES RADC Project Engineer Rocco F. Iuorno/EMBIH AC 315 330-7010		12. SPONSORING MILITARY ACTIVITY Rome Air Development Center (EMBIH) Griffiss Air Force Base, New York 13440
13. ABSTRACT A variety of resource allocation problems arising in the design of centralized computer-communication networks are considered, with primary emphasis being placed on the communication network portion of such tele-processing systems. Analytic queueing models for characterizing the systems of interest are developed and used to solve the various design problems with efficient computational algorithms. After a discussion of the important design variables and tradeoffs, a precise performance measure, average response time, is defined. Then, queueing models are presented for the various types of data concentrators. An optimal passage capacity assignment procedure is developed for determining the subchannel capacity assignments in multiplexed links which minimize the average response time of the network. Methods are also presented for optimally allocating link capacity in a network so as to minimize line costs and satisfy a performance constraint. Finally, all previous efforts are integrated and used to develop an algorithm for designing centralized computer-communication networks. The proposed procedure can be effectively used in n terminal networks as long as n ³ sets of design calculations can be performed on the computer in a reasonable length of time. To illustrate the proposed procedures, several representative case studies are presented.		

DD FORM 1473
1 NOV 65

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Queing Models Resource Allocation Optimization Multiplexing Lagrange Multipliers with Constraints Teleprocessing System Design Channel Capacity Data Concentration Link Capacity Network Structure Computer-Communication Network Design						

UNCLASSIFIED

Security Classification

UNIVERSITY OF MICHIGAN



3 9015 02656 7209