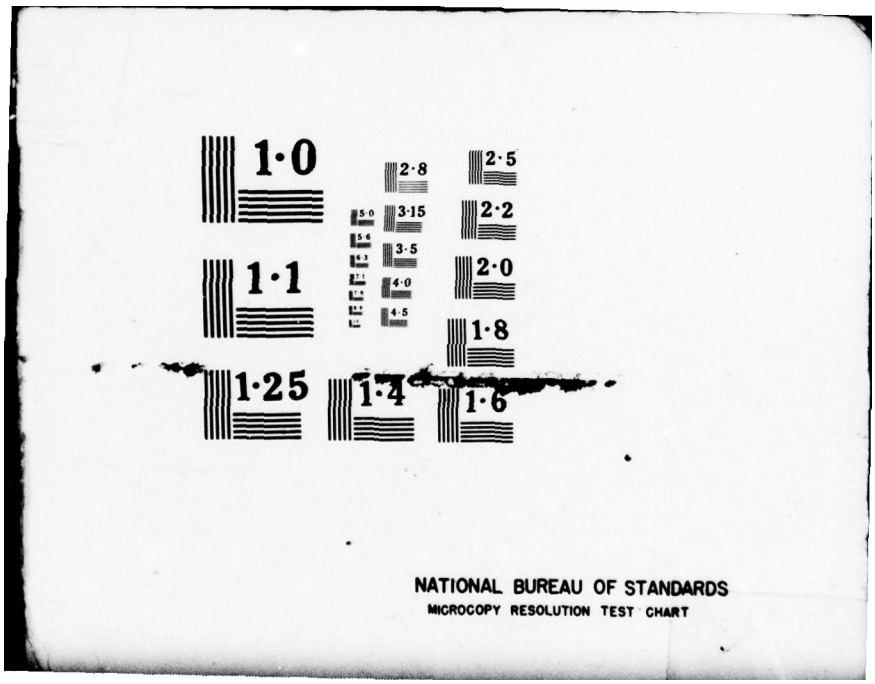
AD-A057 906 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 9/2
COMPUTER NETWORKS. ANALYSIS AND A CASE STUDY DESIGN.(U)
JUN 78 I D ROCHA

UNCLASSIFIED NL

1 OF 3
ADA
057906




1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

AD No. ADA057906
DDC FILE COPY

(2) LEVEL II

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DDC
RECEIVED
AUG 24 1978
B

(9) Master's THESIS,

(6) Computer Networks: Analysis and a Case Study Design.

by

(10) Ivano de Azevedo Rocha

(11) Jun[redacted]78

(12) 220 P.

Thesis Advisors: N. F. Schneidewind
D. A. Stentz

Approved for public release; distribution unlimited.

251450
78 08 23 039

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer Networks: Analysis and a Case Study Design		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ivano de Azevedo Rocha		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1978
		13. NUMBER OF PAGES 220
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Network, computer, computer network, design, time sharing, communication networks, data communications, computer systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Computer networks are the outcome of the combination of computer and data communications technologies. Part I of this thesis presents the background of each one of these technologies and exposes analytically the principles of computer networking in particular. In Part II, a preliminary design is developed for the Naval Postgraduate School (NPS) time sharing system. The study of Part II leads to a comparative analysis of four network architectures which are suitable for satisfying the needs of the NPS. Those architec-

next page

DD FORM 1 JAN 73 1473 (Page 1) EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601 |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

78 06 23 059

tures are compared in view of the desirable attributes of accessibility, evolvability, modularity, reliability and ease of operation. The communications requirements are derived and technologies for implementation of the communications links are recommended.

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

Approved for public release;
distribution unlimited.

Computer Networks: Analysis and a

Case Study Design

by

Ivano de Azevedo Rocha
Lieutenant Commander, Brazilian Navy
B.S., Pontificia Universidade Catolica do Rio de Janeiro

Submitted in partial fulfillemnt of the
requirements for the degrees of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

and

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1978

Author _____

Approved by _____
Thesis Advisor

_____
Thesis Advisor

_____
Chairman, Department of Electrical Engineering

_____
Chairman, Department of Computer Science

_____
Dean of Science and Engineering

_____
Dean of Information and Policy Sciences

ABSTRACT

Computer networks are the outcome of the combination of computer and data communications technologies. Part I of this thesis presents the background of each one of these technologies and exposes analytically the principles of computer networking in particular. In Part II, a preliminary design is developed for the Naval Postgraduate School (NPS) time sharing system. The study of Part II leads to a comparative analysis of four network architectures which are suitable for satisfying the needs of the NPS. Those architectures are compared in view of the desirable attributes of accessibility, evolvability, modularity, reliability and eas of operation. The communications requirements are derived and technologies for implementation of the communications links are recommended.

TABLE OF CONTENTS

I. INTRODUCTION - 11

PART 1 - COMPUTER NETWORKS - - - - - - - - - - - - - - - - 14

II. COMPUTER NETWORK TECHNOLOGY- - - - - - - - - - - 15

A. COMMUNICATION CHANNELS - - - - - - - - - - - 15

B. POINT-TO-POINT COMMUNICATIONS- - - - - - - 18

C. SWITCHING- - - - - - - - - - - - - - - - - - 22

D. MODEMS - - - - - - - - - - - - - - - - - - - 27

E. MULTIPLEXING - - - - - - - - - - - - - - - - 30

F. CONCENTRATION- - - - - - - - - - - - - - - - 40

G. COMMUNICATIONS PROCESSORS- - - - - - - - - - 44

H. BROADCAST COMMUNICATIONS - - - - - - - - - - 50

I. MULTIPOINT LINES - - - - - - - - - - - - - - 57

J. VALUE-ADDED NETWORKS - - - - - - - - - - - - 59

III. FUNDAMENTALS OF GRAPH THEORY - - - - - - - - - - 62

A. INTRODUCTION - - - - - - - - - - - - - - - - 62

B. DIRECTED AND UNDIRECTED GRAPHS - - - - - - - 63

C. RELATIONS- - - - - - - - - - - - - - - - - - 66

D. WEIGHTED GRAPHS- - - - - - - - - - - - - - - 66

E. PATHS AND CYCLES - - - - - - - - - - - - - - 67

F. CUT-SETS, CUTS AND TREES - - - - - - - - - - 70

G. THE MAX-FLOW MIN-CUT THEOREM - - - - - - - - 77

IV. COMPUTER NETWORK RELIABILITY - - - - - - - - - - 80

A. INTRODUCTION - - - - - - - - - - - - - - - - 80

B. THE FORD-FULKERSON ALGORITHM - - - - - - - - 81

C. PROBABILITY OF DISCONNECTIONS BY LINK FAILURES - 86

D. NODE FAILURES- - - - - - - - - - - - - - - - 88

E. KLEITMAN'S METHOD- - - - - - - - - - - - - - 90

V. TOPOLOGY - 92

A. SYSTEM CHARACTERISTICS - - - - - - - - - - - 92

B. CLASSIFICATION - - - - - - - - - - - - - - - 94

C. CENTRALIZED NETWORKS - - - - - - - - - - - - 94

D. DISTRIBUTED NETWORKS - - - - - - - - - - - - 99

1. LOOP - - - - - - - - - - - - - - - - - - - 100

2. LOOP WITH CENTRAL SWITCH - - - - - - - - 102

3. GLOBAL BUS - - - - - - - - - - - - - - - - 103

4. BUS WITH CENTRAL SWITCH- - - - - - - - - 104

5. BUS WINDOW - - - - - - - - - - - - - - - - 106

6. COMPLETE INTERCONNECTION - - - - - - - - 107

7. STAR - - - - - - - - - - - - - - - - - - - 109

8. MULTIPROCESSOR - - - - - - - - - - - - - - 111

9. REGULAR INTERCONNECTION- - - - - - - - - 112

10. IRREGULAR INTERCONNECTION- - - - - - - - 113

E. GEOGRAPHIC DISPERSION OF DISTRIBUTED TOPOLOGIES - - - - - - - - - - - - - - - - - - 115

VI. NETWORK INTEGRATION- - - - - - - - - - - - - - - 118

A. INTRODUCTION - - - - - - - - - - - - - - - - 118

B. INTERFACE COMPONENTS - - - - - - - - - - - - 119

C. PROTOCOLS- - - - - - - - - - - - - - - - - - 120

D. CONTROL REQUIREMENTS - - - - - - - - - - - - 122

E. ROUTING- - - - - - - - - - - - - - - - - - - 124

1. DETERMINISTIC ALGORITHMS - - - - - - - - 125

2. STOCHASTIC ALGORITHMS- - - - - - - - - - 128

3. ROUTING PERFORMANCE MEASURES - - - - - 132

F. FLOW CONTROL ALGORITHMS- - - - - - - - - - 133

G. THE CCITT STANDARD HOST INTERFACE-X.25 - - 136

PART 2 - DESIGN OF A NETWORK FOR THE NPS TIME SHARING SYSTEM 141

VII. PROBLEM DEFINITION - - - - - - - - - - - - - - - 142

A. THE PRESENT NPS CENTRAL COMPUTER FACILITY - - - - - - - - - - - - - - - - - - 142

B. THE NEED FOR A NEW SYSTEM- - - - - - - - - 146

1. RELIABILITY- - - - - - - - - - - - - - 147

2. NEED FOR A POWERFUL BATCH COMPUTER - - 147

3. INADEQUACY OF THE PRESENT TIME SHARING SYSTEM - - - - - - - - - - - - - 149

C. SCOPE OF THE DESIGN- - - - - - - - - - - - 151

VIII. DESIGN BACKGROUND- - - - - - - - - - - - - - - 153

A. MAN-MACHINE INTERFACE- - - - - - - - - - - 153

B. CHARACTERIZATION OF THE TIME SHARING AT NPS - - - - - - - - - - - - - - - - - - - 157

1. CHARACTERISTICS OF THE WORK- - - - - - 157

2. CHARACTERISTICS OF THE SOFTWARE RESOURCES- - - - - - - - - - - - - - - 159

3. ANALYSIS OF DATA - - - - - - - - - - - - 160

C. DESIGN GOALS - - - - - - - - - - - - - - - - 175

1. ACCESSIBILITY- - - - - - - - - - - - - - 175

2. MODULARITY - - - - - - - - - - - - - - - 175

3. ADAPTIVENESS OR EVOLVABILITY - - - - - 176

4. RELIABILITY- - - - - - - - - - - - - - - - 176

5. PERFORMANCE OR EFFECTIVENESS - - - - - - 177

6. SIMPLICITY OF USE- - - - - - - - - - - - 178

7. EASE OF OPERATION- - - - - - - - - - - - 178

D. DESIGN CONSTRAINT- - - - - - - - - - - - - - - 178

E. REQUIREMENTS - - - - - - - - - - - - - - - - - 179

1. HIGH LEVEL LANGUAGES - - - - - - - - - - 179

2. DEBUGGING FACILITY - - - - - - - - - - - 179

3. LINE EDITOR- - - - - - - - - - - - - - - 179

4. TEXT EDITION - - - - - - - - - - - - - - 179

5. INTEGRATED BATCH-TIME SHARING- - - - - 180

6. SYSTEM LIBRARY - - - - - - - - - - - - - 180

7. USER LIBRARY - - - - - - - - - - - - - - 180

8. CALCULATOR MODE- - - - - - - - - - - - - 180

9. MAIL SYSTEM- - - - - - - - - - - - - - - 180

10. DATA BASE MANAGEMENT - - - - - - - - - - 180

11. GRAPHICS TERMINALS - - - - - - - - - - - 181

12. COMMAND LANGUAGE - - - - - - - - - - - - 181

13. SECURITY - - - - - - - - - - - - - - - - 181

IX. PROPOSED ARCHITECTURES - - - - - - - - - - - - - 184

A. INTRODUCTION - - - - - - - - - - - - - - - - 184

1. THE NPS CAMPUS - - - - - - - - - - - - - 185

2. CENTRALIZED AND DISTRIBUTED SOLUTIONS- - - - - - - - - - - - - - - - 187

B. ARCHITECTURE 1 - - - - - - - - - - - - - - - 189

C. ARCHITECTURE 2 - - - - - - - - - - - - - - - 195

D. ARCHITECTURE 3 - - - - - - - - - - - - - - - - 198

E. ARCHITECTURE 4 - - - - - - - - - - - - - - - - 203

F. COMMUNICATIONS FACILITIES- - - - - - - - - - 208

1. RJE COMMUNICATION DATA RATES - - - - - - 209

2. TERMINALS DATA RATES - - - - - - - - - - 210

3. APPROXIMATE NUMBER OF TERMINALS- - - - 211

4. TOTAL BANDWIDTH REQUIRED AT EACH REMOTE SITE - - - - - - - - - - - - 212

5. COMMUNICATIONS TECHNIQUES OPTIONS- - - - - - - - - - - - - - - - - - 213

X. CONCLUSION - 215

LIST OF REFERENCES - 216

INITIAL DISTRIBUTION LIST- - - - - - - - - - - - - - - - - 220

LIST OF TABLES

TABLE		
8.1	VOLUME OF TIME SHARING USE	161
8.2	TERMINAL USE	163
8.3	CPU UTILIZATION	167
8.4	AVERAGE NUMBER OF SESSIONS PER USER	169

I. INTRODUCTION

Computer systems can be viewed in two ways: as information systems or as computation systems. In an information system data manipulation, data storage and retrieval in databases predominates over the computation aspect. In a computation system the mathematical transformation of input data into output is the main purpose of the system. Both aspects coexist in any computer system, yet generally one is stronger than the other.

Modern society is dominated by the existence of big and complex organizations inside and outside of governments. The only way to achieve efficiency, coordination and control in these giants is through prompt and reliable information. On the other hand, technology and science has progressed toward a sophistication which requires more computer power.

The greatest motivation for computer networks is, regardless of the main purpose of the computer system, sharing of expensive computers. The resources that can be shared are databases, processors and software. The sharing of data bases is the more important aspect for information systems, while the sharing of expensive, powerful processors, specialized processors and software is the main goal of computation systems. By sharing those expensive resources, computer networks can achieve cost-effectiveness.

Other important motivations for computer networks are

accessibility, flexibility, generality, availability, reliability and efficiency.

Those points are best stressed by the following quotation of Dr. Ruth M. Davis[1]:

> "1. Computer networks are essential for all those real time geographically dispersed control activities vital to our individual and national well-being.
>
> 2. Computer networks are the only practical means available for the sharing of expensive information resources, computing resources, and information handling equipment.
>
> 3. Computer networks are the only practical means of providing equality of access to and an equality of quality in public services, independent of geographical location.
>
> 4. Maxicomputers available through computer networks are perhaps the only economically justifiable means for the large scientific calculations essential to the advancement of much needed basic research and engineering.
>
> 5. Centralized management, in a real-time sense, of geographically dispersed organization is impossible without computer networks."

This thesis is divided in two parts. Part 1, Computer Networks, comprises chapters II to VI and is an analytical exposition of the technology and principles of computer networks with emphasis on generality and breadth. The second part, Design of a Network for the NPS Time Sharing System, comprises chapters VII to X and is a study leading to the suggestion of network architectures which are advantageous

---

[1] Computing Networks: A Powerful National Force-Keynote Speech COMPCON 73, February 27, 1973.

for the replacement of the present time sharing system at the Naval Postgraduate School.

PART 1

Computer Networks

II. COMPUTER NETWORK TECHNOLOGY

A computer network, also called computer-communication network, is, in the broad sense, any system composed of one or more computers and terminals, communication transmission facilities, and specialized or general purpose hardware to facilitate the flow of data between terminals and/or processors. Its parts consist of communication devices, host processors, transmission lines and a set of rules, implemented in either hardware or software, to insure the orderly flow of traffic in the network [Ref. 9].

A. COMMUNICATION CHANNELS

A communication channel is a path along which signals or data can be sent.

From the point of view of the user a discrete channel, i.e., a channel used to transmit a finite set of symbols, is measured by its capacity or transmission rate and its probability of error. The capacity is the speed at which information can be transmitted by the channel in symbols/second and usually given in bit/sec (binary digits/second). The probability of error or error rate of a channel measures the degree of uncertainty associated with the transmission of information over the channel and is sometimes given in bits in error/bits transmitted (probability is a dimensionaless number); it is generally referred to as the bit error rate or, BER. The probability of error of a channel may be made

as small as desired; one of Shannon's theorems states that it is possible to transmit information through a channel up to its capacity C with an arbitrarily small frequency of errors and that it is not possible to transmit at a rate higher than C with an arbitrarily small frequency of errors. This result may be regarded as a theoretical possibility; it is, indeed, practically possible to reduce the probability of error of a channel as much as desired by the use of error correcting codes, but at the cost of diminished capacity in the transfer of information; so, capacity can be traded for error rate. Up to now, no method is known for constructing codes that can achieve data speeds near capacity with vanishingly small error rates.

A continuous channel, that is, a channel which can transport analog signals or continuous waveforms, can be characterized in a very simplified way by two main factors: its frequency band and signal-to-noise ratio. The frequency band measures the capability of the channel in transporting sinusoidal signals of different frequencies; it can be concisely (although not precisely) specified by the lowest and highest frequencies which are 3 dB (half-power) below the center of the band of the channel transfer characteristic. A more simplified measure is the bandwidth of the channel which is the difference between the maximum and minimum frequency of its frequency band. The signal-to-noise ratio is a measure of the purity of the signal which arrives at

the receiving point.

The capacity of a continuous channel in bps is related to its bandwidth W and signal-to-noise ratio S/N by another one of Shannon's Theorem [Ref. 42]:

$$C = W\log_2(1+\frac{S}{N})$$

Again, this is a theoretical limit. In practice, the capacity of a continuous channel given by the equation above is far from being achieved.

Many times the term "channel" is used implying one way communication, whereas the word "circuit" sometimes implies two-way communications. Usually a communication circuit is defined as the complete electrical path providing one or two way communication between two points comprising associated send and receive channels.

Whether or not a communication facility has the capability of transmitting in both directions, simultaneously or not, can be an important factor in system design. According to this capability the operation of a communication facility may be classified into [Ref. 1]:

1. Simplex operation - only one direction of transmission;
2. Half-duplex operation - two directions of transmission are used one at a time;
3. Full-duplex operation - both directions of transmission can be used simultaneously.

Some communication media have the ability of simultaneous multipoint connection. Broadcast systems are those which permits simultaneous transmission from a single source to a variety of recipients. The predominant operational characteristic is that only one source can be transmitting at a time, while many other components of the system may be receiving from that source at the same time.

A point-to-point communication media, on the other hand, permits transmission and reception only between clearly defined termination points.

B. POINT-TO-POINT COMMUNICATIONS

Establishment of point-to-point communications is one subject of the communication engineering field. The technology for implementation of point-to-point circuits varies according to technical, economical and systems factors and can be classified into two groups: transmission lines and radio links.

Transmission lines are physical paths which can carry or guide electrical or electromagnetic waves. Examples of transmission lines are twisted pairs of copper wires, coaxial cables, wave guides, open wire, parallel-wire lines and fiber optics.

When it is necessary to transport information through a communication system this information is first put into the form of an electrical signal. Usually this signal cannot

be transmitted in its original form; this is possible through some transmission lines, usually in short distances; it is almost never possible through radio links. A intermediate transformation has to be performed. This original signal that is to be transmitted is called the baseband or basic signal. A sinusoidal signal called the carrier has one or more of its attributes of phase, frequency and amplitude modified by the baseband signal through the process of modulation and then is fed in the transmission media. Thus, through modulation the baseband signal is superimposed on the carrier, which transports or "carries" it. Depending on what atribute was altered, the process is called amplitude modulation (AM), frequency modulation (FM) or phase modulation (PM). The converse process of extracting the baseband signal from the carrier is called demodulation. This process is accomplished by sensing the attribute of the carrier which was previously modified.

Radio links utilize the phenomenon of electromagnetic propagation through a media, usually the atmosphere, to transport information. The frequency of the carrier determines important characteristics of the system. The most suitable bands of the radio spectrum for data transmission in computer networks (and in general) are the high end of VHF band (30-300MHZ), UHF (300-3000MHZ) and SHF (3-30GHZ). Because of problems of congestion of the spectrum, the VHF band is seldom used in computer networks.

Due to the cost involved, rarely is a communication system built for the implementation of a geographically dispersed computer network. Instead, to minimize the initial investment, communications facilities are obtained from common-carriers. Common-carriers are public utility companies that are recognized by an appropriate regulatory agency as having the authority and responsibility to furnish communication services to the general public.

For example, for great distances and small capacities it is more economical to lease a point-to-point circuit from a common-carrier. For short distances and large capacities it may be advantageous to implement a point-to-point communication facility over the option of leasing, if the costs are computed over the life cycle of the system. Typically, transcontinental lines are leased and intra-facility lines are the user's property.

Data services offered by common-carriers can be divided into private (leased) lines and switched (public) lines [Ref. 17]. In either case there is a variety of bandwidths. For voice bandwidth facilities the switched data calls are routed through the ordinary DDD (direct distance dialed) telephone network and a different connection is usually obtained each time a call is dialed. On the other hand, a private line does not traverse the switching equipment and its transmission characteristics are fixed. Conditioning is the technique of compensating for undesirable transmission

characteristics on a leased line by the use of equalization filters. This provides greater bandwidth and transmission quality for data transmission than a switched connection. Depending on the length of the line and usage pattern, the cost of a private line can be more or less than the cost of the switched network. Typically, if more than several hours of traffic are to be carried each day between two points, the private line is the more economical of the two alternatives. On the other hand, voiceband switched lines are used where the delay in establishing the connection is not important and where a slightly higher probability of error can be tolerated. Switched lines are typically used in computer-terminal connections.

The bandwidth of a voiceband circuit is something less than 3KHZ. These circuits can support data rates of up to 4800 bps on DDD calls and up to 10kbps on private lines /Ref. 17/.

Narrowband circuits are those of less than voice bandwidth, obtained by division of a voiceband facility into smaller ones.

Wideband circuits are those of greater width than a single voice bandwidth. There is a hierarchy of wide-band channels, obtained by the aggregation of smaller ones. Above the voice bandwidth comes next the group, which is equivalent to 12 voice channels, has 48 KHZ of bandwidth and, with associated modulation and demodulation devices, gener-

ally transmits data at a 50 kbps rate, although higher speeds are possible [Ref. 17]. Nearly all wideband circuits are private line arrangements, but there are switched networks available for broadband transmission. Occasionally wideband circuits (50 Kbps) are leased from a common carrier and used in a private switched network, such as the Department of Defense ARPA network.

The common-carriers, in addition to the analog channels primarily designed for transmission of voice, also offer true digital transmission systems, as, for example, the T-1 carrier which has a capacity of 1.544 Mbps and may transport up to 24 voice channels in digital form, using pulse code modulation (PCM) [Ref. 17].

C. SWITCHING

The number of communication circuits required for directly connecting any pair of a network with N stations is equal to $\frac{N(N-1)}{2}$. Thus, as the number of points increases, the option of complete interconnection becomes economically infeasible. For this reason, large networks using point-to-point communication facilities are not completely connected. To share the existing circuits in a manner that allows communication between any pair of points, switching techniques are employed.

Circuit switching, as illustrated by the voice telephone network, is a system whereby a complete physical path is

established from sender to receiver that remains in effect for the duration of the conversation [Ref 10]. In most telephone exchanges, this route is actually established by mechanical means (such as relays), though the selection of the route may be made electronically. The process of selecting a route, or call establishment may take on the order of seconds for a complex network. This speed is usually deemed too slow to be effective for supporting dynamically established communication between host computers. However, the increasing speeds associated with improved switches have permitted utilization of circuit switching by computer communications utilities. On the other hand, once the route is established, data transfer is continuous through the network in the sense that there are no delays added to data by the switches. End to end transmission delay is determined by the propagation times of the various circuit media employed. Circuit switching has been the primary technology in support of computer-terminal connections in remote-access networks.

Effective utilization of circuit switching requires careful matching of circuit capacity against the transmission requirements, which usually are time-variant. Unused bandwidth cannot be shared with other pairs of network components and the maximum information transfer rate is limited by the capacity of the circuits in the route. For these reasons, circuit switch seems ill-suited to "bursty" traffic, as in

interactive use, because throughput requirements vary and result in line under utilization. Another factor is that circuit switching is subject to line contention delay when some circuits required for a particular path are busy.

A message is defined to be a logical unit of information from the viewpoint of the user. Telegrams, files, programs and queries are examples of messages. As a natural alternative to circuit switching, a system providing message delivery service can be devised. Thus, in a role analogous to the Post Office, a message is "dropped into" the system which assumes responsibility for its delivery. In message switching, an individual message is stored and forwarded at each switch, along its path from source to destination, on the basis of address information it carries with it. Message switching differs from circuit switching in that circuits are not dedicated for exclusive use of given sender-receiver pair. As a result, the opportunity exists for utilizing one circuit to carry traffic between several source destination pairs. This, in turn, permits increased circuit utilization [Ref. 29].

Two challenges in achieving effective operation of a message switched system must be resolved: (1) balancing resource utilization against delay; and (2) introducing some type of mechanism to give short messages reasonable service [Ref. 29].

Transmission delay between source and destination exists

in message switching as a result of the contention for resources one message encounters each time it reaches a switch and waits for transmission to the next step. The first issue is a typical management problem; the conflicting desires of users and management must be mediated. Users want minimum delay, which requires low utilization, whereas management demands high resource utilization, which incurs long delays. The second problem can be solved by having the quality of service proportionally distributed according to the amount of work requested i.e., manage the system in such a way that the delays are approximately proportional to the size of the messages. One method of accomplishing this objective is to establish a priority scheme based on the message length. But this can conflict with operational priorities. Moreover, it is natural to attempt to defeat this mechanism by subdividing a long message into shorter pieces.

Packet switching is a special type of message switching distinguished by a number of characteristics which define a particular type of communication system. With this technique, messages are typically broken into a series of short, fixed-length, addressed packets of data which are routed independently to their destination using store-and-forward procedures as in message-switching [Ref. 10]. At the destination message processor, the message is reassembled from the packets. Single packet messages can then be transmitted with a minimum of delay while throughput can be achieved for multi-

packet messages through simultaneous transmission of several packets through different paths. Through a complex evaluation of individual circuit and switch loads, packet switching computers dynamically alter routing of packets to the minimum delay circuits. Thus, packets of one message may traverse different routes and arrive at the destination out-of-order.

Packet-switching networks typically implement a "logical circuit" (in no way related to the physical paths traversed by packets) from source to destination with the following properties [Ref. 17]:

1. high availability-achieved by means of the routing algorithm with the capability of alternate paths;

2. fluctuating delay-due to the statistical nature of the switching process;

3. fluctuating data rate;

4. packet transmission speed advantages (relative to message switching) are achieved at the cost of control overhead on each data packet transmitted.

Packet switching networks reduce the delay associated with message-switching systems by eliminating secondary-storage buffering of data, needed in message-switching because of the space required for holding a queue of messages. In addition, data is transmitted as small units (packets) to reduce queuing on the output circuits and allow "pipelining" from source to destination as mentioned above.

Compared to circuit switching, packet switching is more effective if most of the messages are short as in data base query/response and interactive traffic. But circuit switch is better if all messages are "long", e.g., file transfer. In multi-modal traffic, i.e., mix of long and short messages, packet switching appears to be slightly more advantageous /Ref. 29/.

D. MODEMS

The utilization of the widespread telephone network is, from the point of view of the user, practical and economical for the transmission of data. However, the telephone network consists of voiceband channels. To transmit digital signals over these analog channels, which pass frequencies in the range 300-3400 HZ, it is necessary for a data transmitter to modulate a voice-frequency carrier signal and for a data receiver to demodulate this signal. A data transceiver is consequently known as a MODEM. Such a modem serves to interconnect data equipment with communication circuits. In addition to its basic function of translating between the binary digital signals of the data equipment and the modulated voice-frequency signals of the communication channel, it also performs a number of control functions which coordinate the flow of data between the data equipment.

The majority of modems are designed to accept and deliver a serial stream of binary data at the data equipment interface. There are, however, some types of modems which

handle a character at a time by accepting and delivering binary signals on several parallel interface leads and which modulate multiple voice-frequency carriers for parallel transmission over the voice channel /Ref 1/.

One common type of modem is the acoustic-coupled modem, in which a standard telephone handset is inserted into an acoustic interface connected to the modem. Acoustic-coupled modems operate reliably to speeds of about 300 bps, but are very much used because of their portability (no need for electrical connection to the telephone network).

Private-line modems are available for four-wire, full-duplex operation. In modems intended for two-wire operation a reverse channel for slow-speed signaling in the reverse direction is usually present. Full-duplex operation in two-wire channels is only possible at low data rates. In half-duplex operation the time required to turn the direction of transmission around becomes important. Also related to this turn around time is the initial acquisition period, i.e., the delay between the request for transmission and the reception of permission to transmit. This set-up time can amount to several hundred milliseconds. In polled systems this delay can become important.

In many high-speed modems a so-called automatic equalization is used to cope with amplitude and delay distortion and to reduce line conditioning requirements /Ref. 1/. Automatic or adaptive equalization is implemented using

transversal filters, which are tapped delay lines with adjustable gain at each tap feeding a summing amplifier. The gains are set by feedback circuitry. One drawback of automatic equalization is the necessary extension of the modem set-up time to approximately adjust the tap settings with special test signals. However, automatic equalization can double the data rates achievable on the telephone channel.

There are modems for synchronous and asynchronous operations. In asynchronous mode one character is sent at a time, preceeded by a start bit and terminated by a stop bit. In the synchronous mode characters are sent in a continuous stream. Transmission is synchronized by a clock internal to the modem with a high degree of stability and precision. This fixed rate of transmission does not require start and stop bits. Almost all high-speed modems are synchronous.

Binary frequency modulation in the form of frequency shift keying (FSK) is the usual choice when simplicity and economy are more important than bandwidth efficiency [Ref. 11]. Typically one frequency is selected to represent the zero and another frequency is selected to represent the one.

Phase modulation (PM) in the form of differential phase-shift keying (DPSK) is employed for higher speeds [Ref 11]. The term differential implies that symbol meaning is based on a change in phase from the previous state and not an absolute phase reference.

Amplitude modulation (AM) is used in transmitting multi-

level symbols, for achievement of high data rates, particularly in connection with a reduction of bandwidth technique such as vestigial side band (AM-VSB) [Ref. 11].

Typical data rates for switched lines are [Ref. 1]:

≤ 300 bps asynchronous

≤ 1200 bps asynchronous

2400, 3600, 4800 bps synchronous;

and for private voiceband circuits are:

≤ 1800 bps asynchronous

2400, 4800, 7200, 9600 bps synchronous.

E. MULTIPLEXING

Communication lines are a major component in the cost of a geographically dispersed computer network. Thus, minimization of communication lines cost is an important objective in network design. Multiplexing is one way of diminishing the cost of communications. Multiplexing refers to the technique of aggregating a number of "low" speed signals into a "high" speed signal. Then, instead of using a low capacity channel for each "low" speed signal, a high capacity channel carries the aggregated signal. At the other end of the channel the aggregated signal is demultiplexed, i.e., it is separated into its "low" speed components. The process is completely transparent to the user. Figure 2.1 illustrates the multiplex process.

The economical reason behind multiplexing is that the

relation cost/bandwidth (dollar per hertz) for leased lines decreases as the bandwidth (or capacity) of the line increases. Then, it is cheaper to lease one high capacity line than many low capacity lines with the same total capacity. Multiplexing is extensively used in the telephone

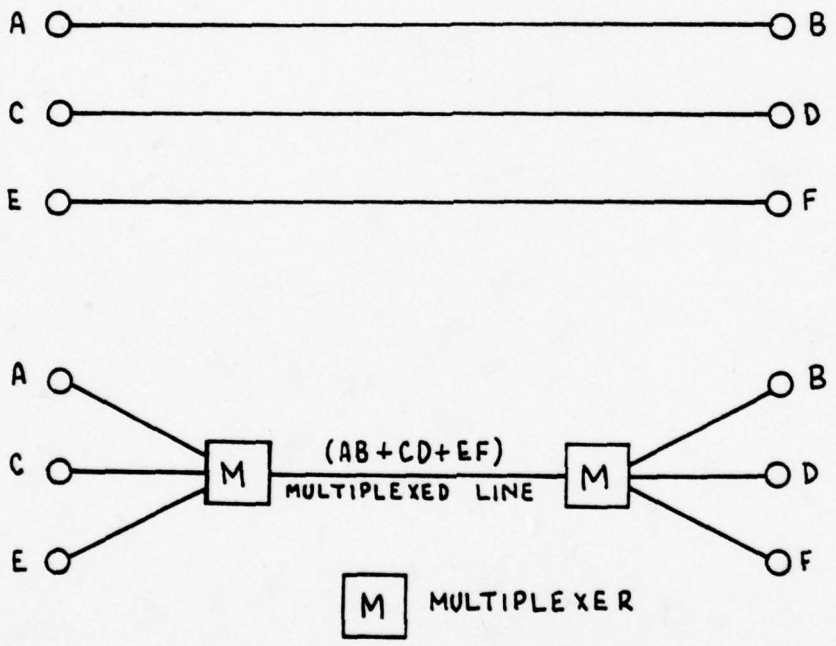


Figure 2.1 - Multiplexing

network for voice channels; even disregarding economics, without multiplexing it would not be possible to implement the multiplicity of the long-haul circuits in existence today over microwave links, because of the problems which a great number of radio frequency carriers would cause in the utilization of the frequency spectrum.

There are two techniques used for multiplexing: frequency division multiplexing (FDM) and time division multiplexing (TDM).

In frequency division multiplexing the frequency bandwidth of the composite channel is divided into smaller bands which are allocated to each component channel (subchannels). A typical FDM system for data transmission is shown in Figure 2.2 [Ref. 14]. In this example, each subchannel is assigned to a frequency band where the binary digits are transmitted in FSK, i.e., one tone corresponding to 1 and another corresponding to 0. Another modulation process could be used in the subchannels such as PSK, DPSK, etc., but FSK is generally used for reasons of simplicity and economy.

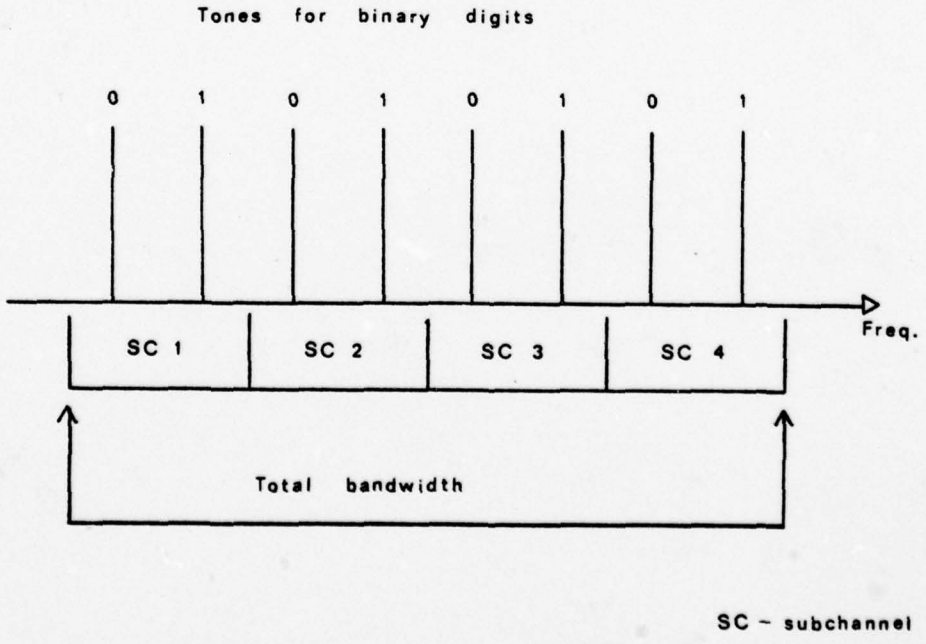


Figure 2.2 - Spectrum partitioning in FDM multiplex for data transmission

When FDM is used on a voice grade line, each subchannel may typically transmit data asynchronously at speeds up to 150 bits/s, although in special cases at faster speeds. One of the limitations of FDM stems from the need for guard bands or safety zones between adjacent subchannels to prevent mutual interference /Ref. 14/. These guard bands impose a practical limit on the efficiency of an FDM system. Also, the generally used FSK is not the better modulation process for bandwidth efficiency. For example, with an FDM equipment operating on a private voice grade line, the maximum composite or aggregate low speed bit rate achievable will typically range from 1800 to 2000 bits/s, although in some cases it can be slightly higher /Ref. 14/.

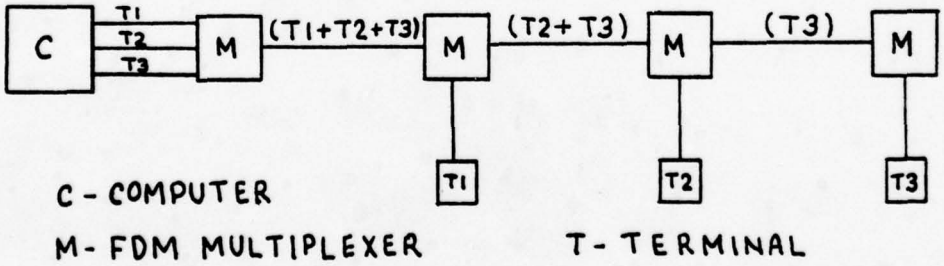


Figure 2.3 - Derivation of subchannels with FDM

FDM's primary advantage to end users is its relatively low cost in applications where FDM's aggreagte bit-rate limit is not constraining. Another advantage of FDM is the ease of making subchannels derivations at intermediate points in a route as exemplified in Figure 2.3 /Ref. 14/. Thus FDM is particularly cost-effective in multiplexing an unclustered terminal group whose aggregate bit rate does not exceed the above mentioned limit.

TDM can be performed in two ways, as synchronous time division multiplexing (STDM) or as asynchronous time division multiplex (ATDM).

STDM aggregates incoming "low" speed channels by allocating a time slot for each one in the composite frame of the "high" speed output. Thus, as shown in Figure 2.4 the relative position of the time slot inside the frame determines to what subchannel the information belongs. STDM samples each incoming subchannel at a constant rate and order, peels off bits or characters and interleaves them onto the high-speed data stream.

For data communications STDM is generally more efficient than FDM because it is capable of higher speeds over the same bandwidth. The example, while FDM goes up to 2000 bps over private voice grade lines, STDM can operate at speeds of 4800, 7200, and even 9600 bit/s in certain instances /Ref. 14/.

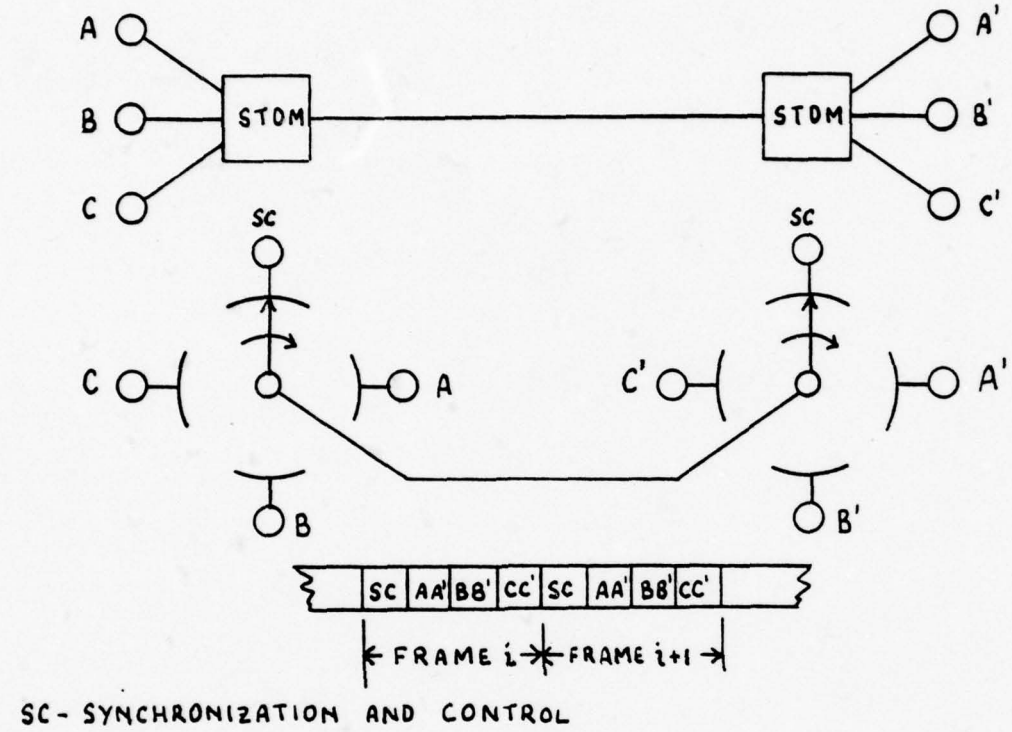


Figure 2.4 - STDM Principle

STDM equipments have to be cascaded to modems for connection to common-carrier lines because their outputs and inputs are digital signals. FDM's, in contrast, don't need modems because they transmit and receive analog signals. A typical use of STDM is shown in Figure 2.5.

STDM may perform bit or character interleaving on the shared line when serving start-stop (asynchronous) terminals exclusively. In these applications, character interleaving is usually more efficient since a modest amount of bandwidth compression is possible. The start and stop bits of each

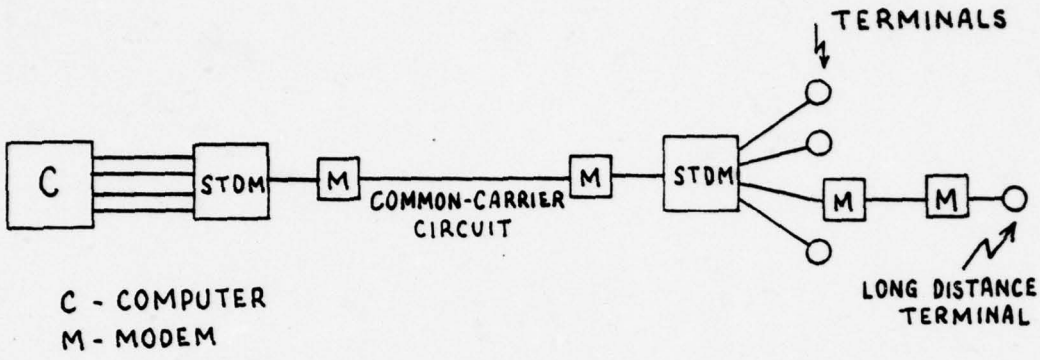


Figure 2.5 - Connection of terminals to a computer with STDM

character entering the STDM may be stripped off prior to the character's insertion into the frame of multiplexed data. Bits stripped from incoming data characters are reinserted by the demultiplexing unit prior to the distribution of the characters to their respective output lines. Thus, the incoming characters are effectively reencoded using fewer bits per character for transmission over the shared line, enabling a typical aggregate low-speed bit rate of 1.1 times the shared link's transmission rate [Ref. 1, 14].

When used in conjunction with synchronous lines, STDM generally uses bit interleaving. Whether bit or character interleaving is used, special predetermined code sequences are used between STDM's to define the beginning of each new frame of multiplexed data and to maintain synchronization. Demultiplexing is thus accomplished assuming an implicit relationship between the output lines and the relative position

of the time slot as shown in Figure 2.4.

In comparison to FDM's, STDM's are expensive for derivations because a relatively complete STDM unit and associated modem have to be used at any point where one or more subchannels are being picked up or dropped. Also, the problem of synchronization among the multiples STDM units has to be addressed. Fundamentally two options exist in this regard - to use a master clock from which all elements derive their timing or to use an independent synchronous clock at each line. This second process requires buffering to absorb a data build due to slight variations in the frequency of independent clocks. Most of the STDM networks use the master clock option due to economic reasons /Ref. 14/. Figure 2.6 displays an example of a synchronous TDM network with derivations.

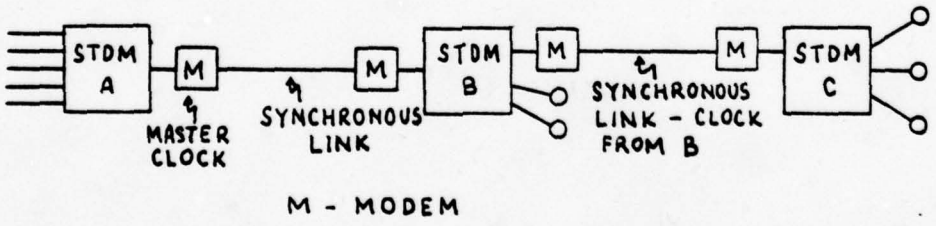


Figure 2.6 - STDM network with derivation

In summary, both FDM and STDM are widely used in data networks for economy in communication costs. The main usage has been in computer-remote terminals connections. When the aggregate low-speed bit rate for all terminals does not exceed 2000 bps, either FDM or STDM can be used, but FDM

will be more cost effective [Ref. 14]. Whenever a higher aggregate bit rate is required, or whenever any synchronous terminals are included in the sharing group, STDM will usually be required. However, in higher bit rate applications involving geographically diverse terminal locations, the use of an integrated blend of FDM and STDM is indicated. FDM's are used to span isolated terminal sites, creating traffic clusters which are then synchronously multiplexed to one or more distant computer sites.

Asynchronous time division multiplex (ATDM) differs from STDM in that a dedicated subchannel is not provided for each terminal in the sharing group. Since, under certain conditions of heavy loading, an ATDM may be incapable of accomodating all incoming lines in the shared channel, statistics and queuing becoming important considerations. Thus, it is a hybrid technique between multiplexing and concentration (that will be discussed later) and is often called statistical multiplexing.

The fundamental notion behind ATDM is to exploit the fact that in STDM systems, many of the time slots in the fixed-format frames are wasted since a typical remote terminal will actually be transmitting data less than 10 percent of the time it is on line. As shown in Figure 2.7, ATDM dynamically allocates the time slots in a frame of data to the currently active users, reducing the fraction of wasted time slots and thereby increasing overall line utilization and throughput

/Ref. 1, 12, 14/. Analytical studies indicate that approximately from 2 to 4 times as many users could be accomodated on a voice grade line as with STDM, assuming an application environment where either method could be used /Ref. 1, 12/. In certain situations where low-duty cycle terminals are serviced by an ATDM system over a broad-band link, the margin could be greater.

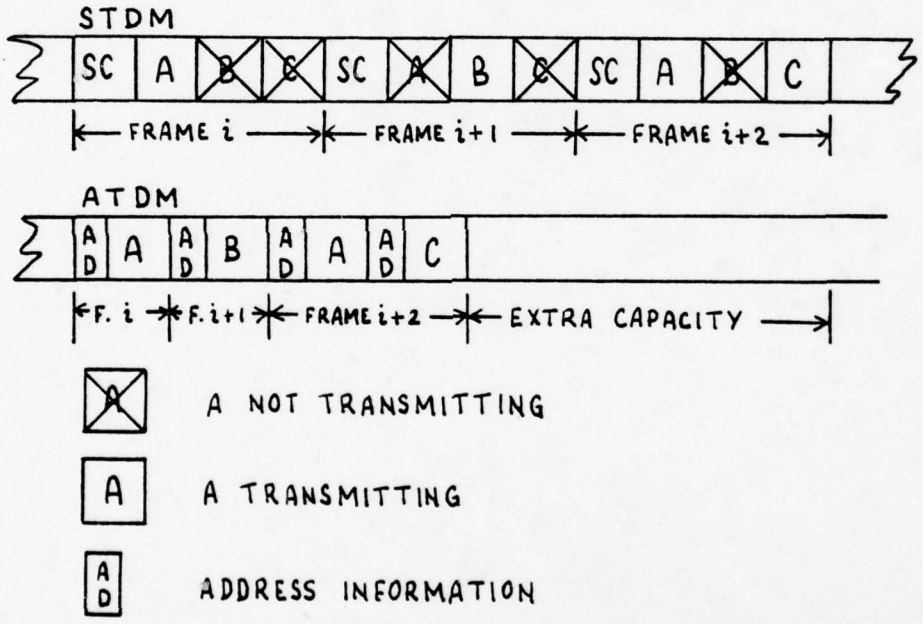


Figure 2.7 - Comparison between STDM and ATDM

The tradeoff disadvantages of ATDM are the costs of substantially more elaborate addressing and control circuitry, data buffers to hold incoming messages, and the possibility of blocking and queuing delays under heavily loaded conditions.

The complexity of control and the storage requirement for buffering lead to implementations of ATDM with the minicomputer /Ref. 9/. This minicomputer implementation implies a fundamental component cost of approximately ten times the cost of a FDM or STDM unit. Such a figure requires large economies to be achieved for cost effectiveness. Such economy can be achieved by handling a large number of channels which can go typically to 64 with increased cost /Ref. 9/. Also, the idle capacity of the minicomputer can be used to perform local operations such as polling, error checking, line control, etc., thus alleviating the central computer of those tasks.

F. CONCENTRATION

The word concentration appears to have a very broad meaning in data communications /Ref. 1, 9/.

Sometimes the word multiplexing is restricted to FDM and STDM. In this context, concentration is distinguished from multiplexing by the characteristic that the shared line or the high speed side of a device performing concentration has less capacity than the sum of all the capacities of the lines which share the high speed facility (low speed side)/Ref. 1/. In a multiplexer the sum of capacities at the low speed side "equals" the capacity of the high speed side. Thus, a multiplexer is transparent to the operation and a concentrator not. ATDM according to this interpretation is classed as concentration.

Sometimes concentration is used in a general sense encompassing multiplexing and message concentration. According to this interpretation, concentrators may be of three types /Ref. 1/:

1. Nonbuffered concentrators - those devices in which data are multiplexed by bit and don't buffer at a character level or more; FDM and bit interleaving STDM are in this class.

2. Buffered concentrators - those devices that buffer at a character level before forwarding; one or more characters are assembled as they are received, before forwarding at a higher rate; character interleaving STDM and ATDM are in this class.

3. Store-and-forward concentrators - those devices that are 'block'-or 'message'-oriented, rather than character oriented as above.

In another usage concentration is used for message switching concentration and line or circuit switching concentration /Ref. 14/.

Message switching concentration (MSC), called store-and-forward concentration, involves the "multiplexing" of entire messages or fixed-length portions of long messages. The MSC accumulates message blocks in it buffer until one is completely assembled and the high-speed line is available to transmit it. Thus the high-speed line transmits variable-length frames of data with appropriate addressing and control

information; all data characters in each frame are generally associated with the same source-destination pair. MSC is thus a special case of message switching.

The primary disadvantage of MSC, compared to FDM or STDM, are economic in nature, and relate to the cost of a stored program computer and buffer storage usually required [Ref. 9, 14]. Again, the use of a minicomputer has some supplementary benefits such as the capability of performing remote line control, code conversion, error checking, selective routing and error control of the multiplexed circuit. Some problems also arise when MSC's are required to handle messages with widely varying block lengths. As in message-switching, when one extremely long user data block is assembled in the concentrator, it may tie up the shared line for an inordinately long time, at the expense of other users. A final disadvantage of MSC relates to its reliability characteristics. Most individual component failures will put the system down, since minicomputers are used. The concentration duplication is often used at each concentration node [Ref. 9, 36]. The high costs generally make other line-sharing schemes more cost-effective in applications involving moderate traffic. But, in large, multiuser networks, message switching concentration affords flexibility and performance advantages over other line-sharing schemes [Ref. 9].

Circuit switching concentration involves a switching device which electrically bridges a group of n inputs to a

group of m output links on a demand basis (n is typically from 3 to 5 times the value of m in commercial applications) /Ref. 14/. Thus, circuit switching concentration is a special case of circuit switching. Ordinarily, the input lines and the output trunks to which they are switched have similar bandwidth and transmission properties. Private automatic branch exchanges (PABX) are examples of circuit switches. Although they have been used for conventional voice telephony, such devices may function equally well when used for computer communications as a line switching concentrator with the advantage that they may be built inexpensively.

No message queuing delays are introduced at the circuit switch once a connection is established. But there is contention for the use of the trunks and several possibilities exist for allocating the output trunk lines. The serving mechanism may be first-come-first-serve, with requests that arrive at the switch when all trunks are occupied being lost. Another more complex scheme may queue incoming requests that were not satisfied for future serving, when trunck become available.

The Figure 2.8 shows a typical application for a circuit switching concentrator.

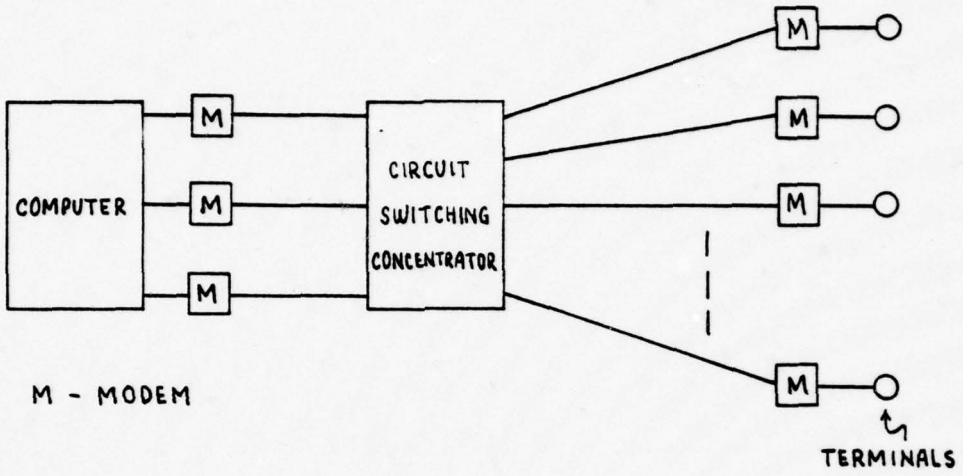


Figure 2.8 - Example of circuit switching concentration

G. COMMUNICATIONS PROCESSORS

Communication processors, also called network processors, are minicomputer based devices performing one or more of the following functions [Ref. 36]:

1. front-end processing;
2. message switching;
3. message concentration;
4. terminal multiplexing and control.

The front-end processor or FEP is a rather flexible concept so that, like many other computer terms, this one has no precise meaning. In general, a computer attached to a host computer and handling communications functions for this host is called a FEP [Ref. 12].

Many of the tasks performed in handling communications circuits are relatively simple, but highly repetitive, and can make considerable demands on the time of the host computer, which sometimes does not have appropriate capabilities for the real-time demands presented by data communications. It often proves more economic to have these functions being performed separately by the front-end processor. The main advantages of this approach are /Ref. 12/:

1. The cost of hardware to attach lines is often less with a minicomputer, because they are more flexible in accomodating a wide spectrum of communications network facilities, devices and line control procedures.

2. The processing load removed from the main computer will considerably increase the power available for computational purposes.

3. It becomes possible to separate the complete system cleanly into two parts: the main processor and the communications network. This gives increased flexibility and may allow one part to be enhanced or replaced without affecting the other. The communications network requirements are continuously changing due to the need to accomodate a large number of different terminals and communications facilities. Such changes should not be allowed to reflect back into the rather complex host computer operating system. With the use of an FEP the desired isolation can be attained and any configuration changes can be made in a smaller operating

environment.

Various general purpose communications functions are required by almost every application. These functions, which are normally handled by software, include [Ref. 36]: control of the communications network and its terminals, buffering and queuing of messages, checking message validity, authorization of the message senders and receivers, routing of messages between their different sources and sinks (as represented by different programs and/or terminals), message editing and formatting, etc. The partitioning of these functions between the FEP and the host computer may be done in various degress. More benefit of the use of the FEP will be achieved as the FEP assumes more communications functions thus alleviating the host of this load.

The connection of a FEP to a host may take different forms. In the simplest one the FEP replaces and emulates a communications controller. The FEP is thus attached to the host or to the host channel and acts as a device controller. Such a FEP is sometimes called transparent because it gives to the main computer a 'familiar' picture of the terminals. The advantage in this case can be its low cost compared with the stardard equipment, but it has the special advantage of flexibility, i.e., it can be made to accept 'foreign' terminals (not from the main computer manufacturer). The standard communications software of the host operating system is used and the host is not alleviated of the communications tasks [Ref. 12].

Another possibility of connecting a FEP to an already existent system, without the need of writing special software for the main computer operating system, is to use the standard software associated with different peripherals, such as disc drivers [Ref. 12]. The FEP then protects the main computer from many of the communication tasks, and leaves it with only those that are common to all peripherals.

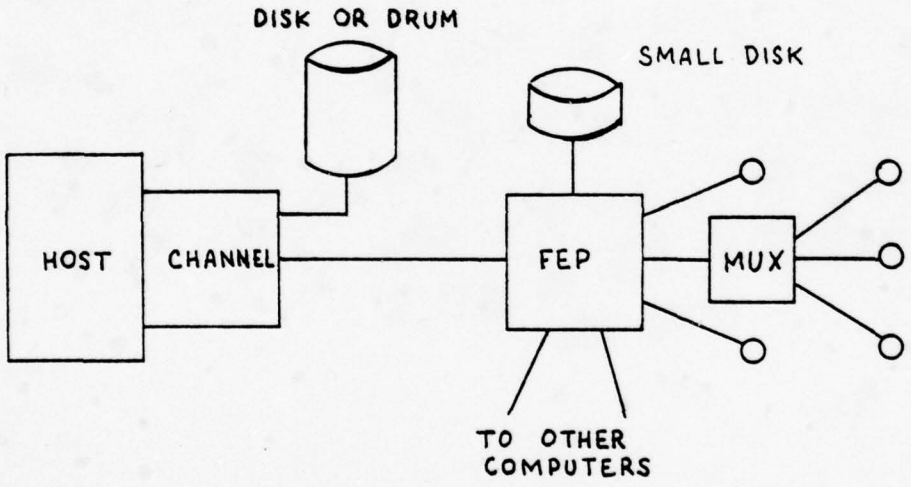


Figure 2.9 - FEP - host connection

In the configuration of Figure 2.9 the main computer handles the file system, using its large disc or drum store. The FEP has a small disc which can hold message queues and files for a few users so that editing can be carried out here. Transfers between the two processors can be mainly in large blocks thus protecting the host from all rapid interactions.

In some configurations there is no small disc attached to the FEP in Figure 2.9. Then, the only FEP storage is main memory and it does not handle some of the message control functions. The FEP in such cases is typically responsible for communications circuit control, line switching and terminal control, activities which sometimes are called network and device control [Ref. 36].

In the configuration of Figure 2.10, called a disk-coupled system, the FEP shares a disc with the main computer [Ref. 47]. The FEP handles most of the communication tasks and assembles complete messages on the disk for attention by the main computer. Only when it recognizes a command from the user that needs the service of the main computer, does it send that machine a task, which it does by a message stored on a special region of the disk. Conversely when the host processor wants to communicate, it notifies the front-end processor of the attributes of the output file that it has stored on the shared disk. The information exchange is thus done in a block basis.

Another organization, the direct coupled shared memory system, requires that both the FEP and central processors have essentially the same architecture and internal timing/control characteristics [Ref. 36, 47]. In this configuration, primary memory replaces the disk as the shared coupling but with information exchange interleaved on a word rather than block basis. The general advantage of the direct coupled

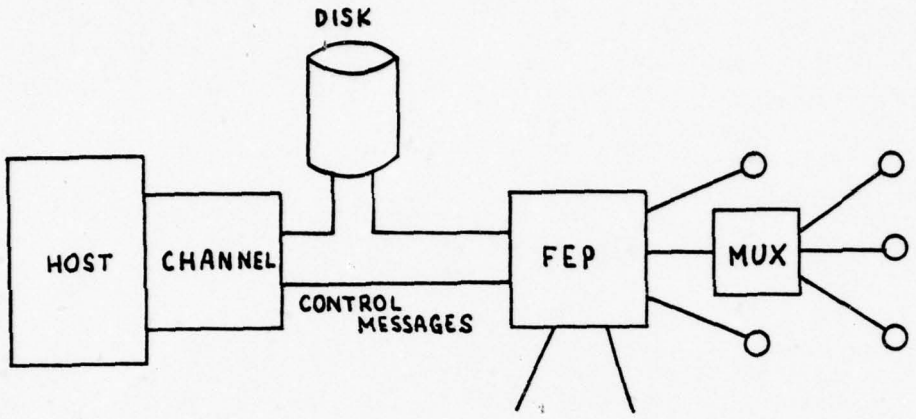


Figure 2.10 - Disk-coupled system

system over the shared disk system is the ability to pass data and programs without the need for intermediate disk storage. This configuration represents the most efficient integration of the FEP and host but it is usually only appropriate when the host and FEP are designed specifically for this application [Ref. 47].

The shared disk and direct coupled connections provide a way, after the architecture has been established, to configure a system that is essentially a dual processor. Larger architectures, on the other hand, are initially designed as multiprocessors with integrated input/output computers or FEP's [Ref. 47].

FEP's are used to remove data communications functions from the host, which can more efficiently perform its data

processing functions. In networks with multiple processors, for the same reason, the hosts are frequently interconnected by a communications subnetwork or subnet [Ref. 17]. The subnet is then an assemblage of communication circuits, FEP's, switches and communication processors. The Figure 2.11 shows schematically the subnet concept. Besides all the advantages of alleviating the hosts from the communications load, the subnet provides well-specified interfaces to facilitate interfacing multiple computers from a variety of vendors, which otherwise would be very difficult or even impossible to interconnect in a large network [Ref. 17]. In environments with a single mainframe or small, homogeneous, single-vendor, networks the subnet concept may not be employed.

H. BROADCAST COMMUNICATIONS

Most of the experience with computer networks has involved point-to-point circuits characteristic of common-carrier systems. Yet the broadcast multipoint feature of radio transmissions is of utility in many situations [Ref. 1, 30, 31].

A geographically widespread distribution of users may preclude joint usage of such a point-to-point communications facility. Depending upon the particular locations of the users, it can be more economical to provide each with a separate transmission path to the computer without the savings derived from the use of sharing techniques, such as

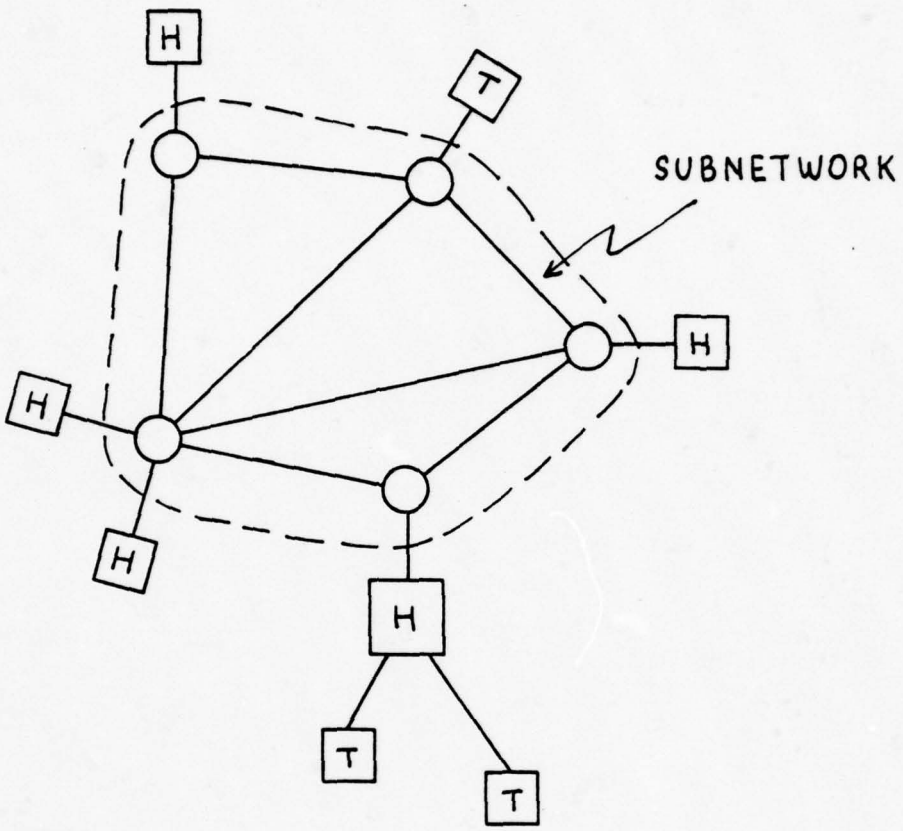


Figure 2.11 - Network with communications subnetwork

multiplexing, concentration and switching. Furthermore, commercially available transmission capability is often ill-suited for the data communications required in some remote computer access applications. Another requirement, mobility in the military environment, cannot be met by point to point circuits.

Under these conditions, netted operations with multipoint radio links can provide alternative means of communications. By appropriately choosing the operating frequency and multiple-access technique, multipoint links have the potential of simultaneously connecting many users and providing a reliable broadband digital communications capability. In general, the organization derived from the use of a multipoint broadcast facility is distinct from that of point-to-point communications [Ref. 17].

Multiple-access techniques are disciplines imposed on the utilization of a broadcast radio channel. Multiple access, which permits sharing of communications capacity, is distinguished from multiplexing in that the latter refers to joint usage by several baseband signals, whereas the former is concerned with joint usage by several radio frequency (RF) signals [Ref. 17]. Time division multiple access, TDMA, and frequency division multiple access TDMA are the RF analogs of STDM and FDM multiplexing, respectively. In TDMA a fixed time slot is cyclically allocated for each user of the channel over the entire bandwidth. In FDMA a fixed frequency band is allocated for each user all the time. Thus, they are called static reservation schemes [Ref. 1, 45].

Typically, a stationary satellite has a broadband repeater that simply amplifies all received signals and retransmits them in a broadcast mode back to earth. The propagation delay for a roundtrip transmission (up and down) to a

satellite transponder that is in a synchronous orbit is approximately 0.25s [Ref. 45]. If several terminals use the satellite at the same time, multiple-access techniques must be employed to ensure that links are not disabled by mutual interference, that the receiving terminals can identify the desired signals, and that efficient usage is made of the overall capacity of the satellite. Besides TDMA and FDMA there are other ways to use a given satellite channel for data communications. TDMA and FDMA divide the satellite channel in subchannels which are permanently assigned to users (ground terminals); this can be very wasteful in a bursty user environment [Ref. 31].

As alternative, 'random' access to the full capacity of the channel in a packet-switching mode can be employed. Again, a packet is defined merely as an addressed package of data that has been prepared by one user for transmission to some other user in the system. The satellite is characterized as a high-capacity channel with a fixed propagation delay that is large compared to the package transmission time. The transmission scheme to be considered is one wherein a particular transmitter assembles its packet and then, according to some discipline, transmits it rapidly on the channel at full speed. Many users operating in this fashion multiplex their messages on a demand basis [Ref. 1, 31]. The satellite acts simply as a relay accepting the packets and broadcasting them back to earth; this

broadcast transmission can be heard by every user in the system including the packet source, i.e., each user can hear his own transmission; this is called 'perfect feedback' or 'automatic acknowledgement' /Ref. 30, 31/. If one packet arrives at the satellite transponder when another packet is being relayed, both packets are destroyed.

One decentralized approach to the sharing of the satellite channel is the 'pure ALOHA' system. The name "ALOHA' comes from the pioneer computer network, built by the University of Hawaii, employing this scheme. In a 'pure ALOHA', users are allowed to transmit whenever they want. If, after receiving back their transmission, the packet is intact, they assume the transmission was successful. Otherwise, a 'collision' with another packet has occurred or the packet was contaminated by noise; in such cases, they retransmit the packet. If all users retransmit immediately upon hearing a conflict, then a collision will happen again. Thus, some scheme has to be used to introduce a retransmission delay to spread these conflicting packets over time. A random number generator is used to spread the time of retransmission in order to avoid a second collision.

Another uncoordinated approach for using the satellite channel is the 'slotted ALOHA', where time is divided into segments (slots) of the same duration (assuming packets of constant length). Users are allowed to transmit only at the beginning of a time-slot (time referenced to the satellite); then whenever a packet is ready, they will hold it

until the next time mark. The probability of interference is diminished, because collisions are restricted to a single slot duration.

The maximum throughput of a slotted ALOHA system is limited to 0.37 of the capacity of the channel, which is twice as much as a pure ALOHA system /Ref. 31/. However, in a bursty user environment with light traffic, the slotted ALOHA system yields the minimum average delay. TDMA and FDMA lead to a maximum channel utilization (throughput approaching the channel capacity) in heavy traffic; the disadvantage of these latter processes is the delay in low traffic situations, which is bigger than that of the random access techniques.

For ground broadcast radio nets the same techniques for multiple-access may be employed. Typically, FDMA is not used in ground systems because it would introduce stringent requirements for frequency stability and filtering at both user and computer terminals /Ref. 1/. Other methods emphasizing simplicity of terminal design are economically more attractive.

A fundamental difference of ground radio in relation to satellite communications is that the propagation delay is negligible compared to the size of a packet. Exploring this attribute, two more packet switching methods for multiple-access can be employed.

In the carrier sense multiple access mode, CSMA, the

terminals listen ('sense') the channel; if the carrier signal is present then the channel is considered in use by another terminal and transmissions are postponed until the channel is sensed as being idle. This information does not exist in satellite channels, because when the packet reaches the earth, the satellite transponder has already transmitted it. There are rules for deciding when a terminal user may transmit and what action he must take if his transmission collides with another transmission. CSMA disciplines can lead to channel utilizations which approach unity [Ref. 31]. This is a sharp contrast to the poor utilization of ALOHA channels.

The CSMA approach works on the assumption that all terminals are within line-of-sight of each other. This is not always the case; in general, terminals are within the range of a central station (computer center, gateway to a network, etc.), but out of range of each other or separated by some obstacle opaque to radio frequencies. This 'hidden terminal effect' leads to another packet switching channel sharing method, the busy tone multiple access mode (BTMA), because the existence of hidden terminals degrades the performance of CSMA. The operation of BTMA rests on the assumption that the central station is, by definition, within range and line-of-sight of all terminals. With BTMA, the total available bandwidth is divided into the information channel and the busy tone channel. Whenever the central

channel senses an incoming signal in the information channel, it transmits the BT signal on the BT channel. It is by sensing a carrier on the BT channel that terminals determine when the information channel is busy. The performance of BTMA is almost as good as that of CSMA. With the use of BTMA, the delay due to hidden terminals is quite small /Ref. 31/.

I. MULTIPOINT LINES

An important method of increasing line efficiency is to use multipoint or multidrop circuits, rather than point-to-point circuits. This method connects a number of geographically separated devices by a single circuit that passes from device to device.

The Figure 2.12 shows a number of multidrop lines connected to a computer and the Figure 2.13 shows two modes of connection for multipoint lines. Two wire connections are more suitable for short physical transmission lines such as for example the twisted pair of the local loop of a telephone cable. Four wire connections are used in common carrier trunk circuits, which are intrinsically two-way channels.

Two terminals cannot transmit simultaneously in multidrop lines. The line controller or modem at the master position maintains the discipline of the line in some fashion that allows only one transmission at a time. This intermittent nature of the use of a multidrop line may be dis-

guised by the use of buffers at the terminal. The buffer transmits into the line at high speed and communicates with the terminal at the user's rate, holding data until the proper time for transmission /Ref. 12/.

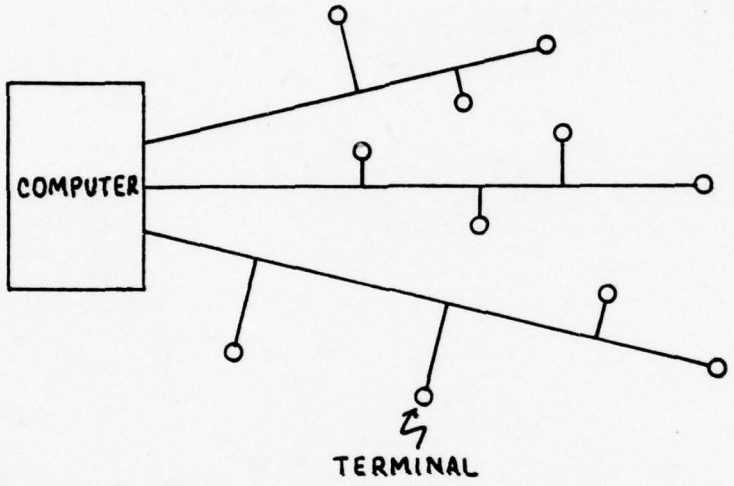


Figure 2.12 - Multipoint lines

A multipoint line is only possible when the times of use of terminals can be interleaved and the sum of the average data transmission rate of each terminal does not exceed the capacity of the circuit. There are many instances in which these conditions are satisfied as, for example, with interactive terminals /Ref. 12/. In addition to the savings in line costs, multipoint lines save in number of modems, since only one is necessary at the computer side, while with point-to-point lines the number of modems is equal to twice the number of terminals.

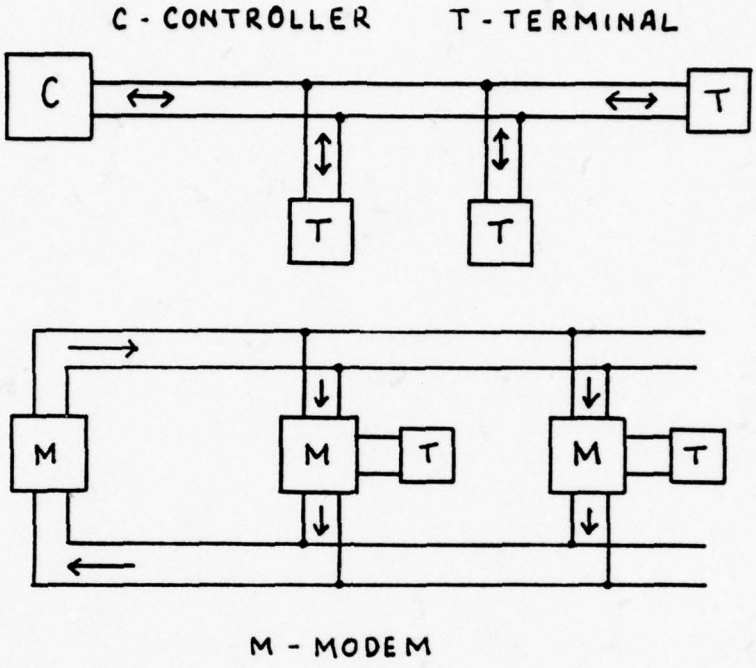


Figure 2.13 - Two (top) and four wire (bottom) multipoint connections

The advantage of multidrop lines may be offset by a reduced reliability and difficult of maintenance, when compared to point-to-point circuits. A failure in a single modem may make the line unusable and may be difficult to locate /Ref. 12/.

J. VALUE-ADDED NETWORKS

Value-added network carriers are private companies who lease services from common carriers to augment these basic services through the use of additional physical facilities

(such as intelligent switching processors) and to resell the "higher value" to the public [Ref. 10]. Thus, value-added networks (VANs) are built on the already existing communications facilities to provide a different kind of public network service. A VAN utilizes the existing common carrier network for data transmission while providing the additional services of interfacing the users' computers and terminals, routing messages to their destination while guaranteeing their integrity, and allowing small users to take advantage of the economies of scale inherent in large communications systems by pooling their demands for service. Thus, a customer need only connect to a VAN in order to gain access to other computers and terminals throughout the nation with no required communications development on his own part. Basically, VANs provide communication subnetwork services.

VANs improve the quality of the basic communications service in a number of ways. Through adaptive multiplexing, high speed lines can be shared by large numbers of users and more efficient use can be made of the available bandwidth. Alternate routing permits the network to maintain continuity of operations in the event of selective internal component failures. Powerful error detection techniques provides an end to end error rate to users that is many orders of magnitude better than a basic communications service could provide.

The intelligent components in a VAN can be employed to provide new services to customers not provided by other

types of communications services. Among these services are speed recognition and conversion, code conversion, and the imposition of a set of network standard interfaces [Ref. 12]. Speed and code conversion permit a wide variety of different terminals to communicate with each other. As an implementation of the subnetwork concept, the VANs provide a way of linking different computers by using standard interfaces.

Besides technical benefits, VANs are economically attractive for small users and users who have varying volumes of traffic.

III. FUNDAMENTALS OF GRAPH THEORY

A. INTRODUCTION

Graphs are used to model an imense variety of problems and physical systems. For many systems the modeling by graphs is quite natural.

Many aspects of computer networks can be easily represented and mathematically studied by the use of graphs. One of such aspects, for example, is the structural model of the network.

A graph may be considered as a set of points called nodes or vertices connected by lines called links or branches.

With each link and node of a graph, we can associate a number of parameters that represent the natural limitations and capabilities of the links and nodes. Those important parameters that are incorporated into the model are called weights. Weights are used in graphs to put non-structural information into the model.

A graph can be defined as an abstract mathematical system. Diagrammatical representations of graphs provide motivation for the terminology and also help to develop some intuitive feelings. Figure 3.1 gives a diagrammatical representation of a graph.

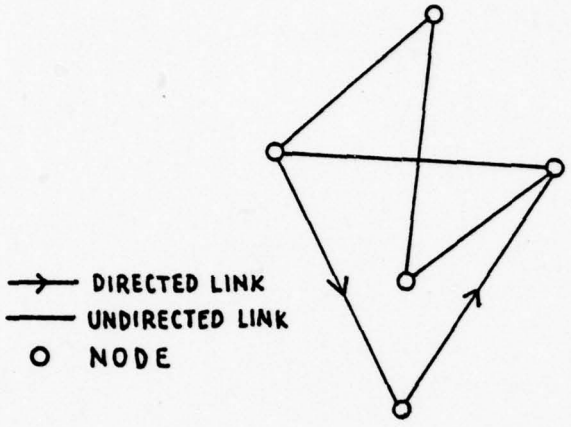


Figure 3.1 - Diagrammatical representation of a graph

B. DIRECTED AND UNDIRECTED GRAPHS

Some links in networks have the attribute of direction as, for example, when modeling a communication channel which allows transmission of information in one direction but does not in the reverse direction. Other links are two-way links allowing flow in both directions. Thus, it is necessary to distinguish between directed and undirected graphs.

A directed graph G=(V,B) consists of a set V of elements called nodes and a set B of ordered pairs of nodes called directed links [Ref. 22]. It is assumed that both sets, V and B, are finite.

The kth node in V is denoted by v_k, where k is any lowercase letter or numeral. Thus, if v has n elements, it can

be denoted as $V=(v_1, v_2, \ldots, v_n)$. Diagrammatically the elements of V are represented by circles or dots as in Figure 3.1. A link of B is represented by (i,j) when that link is directed from node v_i to node v_j. Diagrammatically the link (i,j) is represented by a line with an arrowhead pointing from v_i to v_j.

The node v_i is called the initial node of the link (i,j) while the node v_j is called the terminal node. Link (i,j) is called incident out of its initial node v_i and incident into its terminal node v_j or simply incident at v_i and v_j.

The nodes v_i and v_j joined by the link (i,j) are called adjacent.

The definition of graph does not permit more than one link from node v_i to node v_j, while it is possible to have the links (i,j) and (j,i) simultaneously since they are distinct.

If more than one link going from v_i to v_j is allowed but no more than S links from v_i to v_j the resulting structure is called a S-graph and the multiple links connecting v_i to v_j are designated by $(i,j)_1$, $(i,j)_2$, ... $(i,j)_r$ $(r \leq S)$. Figure 3.2 displays a 2-graph.

Quite commonly the existence of a link (i,j) implies the existence of a link (i,j) in the reverse direction. In such a case the graph is said to be symmetric. When that happens, it is possible to replace the directed branches (i,j) and (j,i) by an undirected branch $[i,j]$, which is

represented diagrammatically by a line without arrowhead, as seen in Figure 3.3.

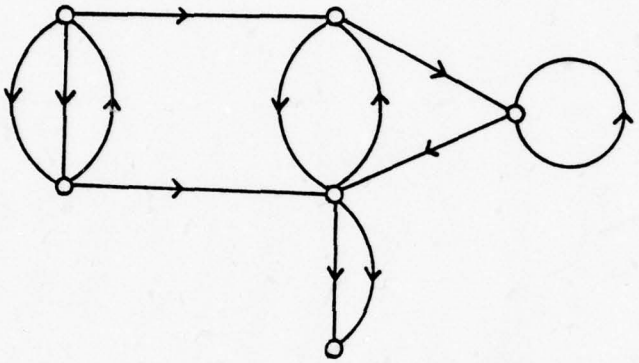


Figure 3.2 - A 2-graph



Figure 3.3 - Replacement of a directed graph (a) by an undirected graph (b)

An undirected graph G=(V,B) consists of a set of nodes V and a set B of unordered pairs of these nodes called links. An undirected link is represented by $[i,j]$ while a directed link is represented by (i,j).

The concept of an undirected graph gives a concise and neat representation of symmetric directed graphs, for example when representing a computer network structure where two-way communication links (duplex or half-duplex channels) are used throughout the network.

C. RELATIONS

Graphs can model not only the physical structure of systems but any abstract concept that can be formalized as a relation.

A relation R on a set S is any subset of the cartesian product SxS, or equivalently any set of ordered pairs (x,y) where $x \varepsilon S$ and $y \varepsilon S$.

The system that can be modeled by the relation R on a set S can be modeled by a graph G=(V,B) by making V=S and B=R.

Thus any abstract concept can be easily represented by a graph provided that it be modeled as a relation.

D. WEIGHTED GRAPHS

A weighted graph is a graph in which numbers are associated with links or nodes. Any number of weights can be associated with each node or link.

The set of weights associated with node v_k is denoted by $\{W_i(k)\}$. The set of weights associated with link (i,j) is denoted by $\{W_r(i,j)\}$.

A weighted graph is symmetric if $w_k(i,j)=w_k(j,i)$ for all

i,j and k. A weighted graph is said to be pseudo-symmetric if

$$\sum_{i \in v} w_k(i,j) = \sum_{i \in v} w_k(j,i)$$

for all j and k.

E. PATHS AND CYCLES

Let $G_1=(V_1, B_1)$ and $G_2=(V_2, B_2)$ be graphs. The graph $G_3=(V_3, B_3)$ is called the union of G_1 and G_2 if $V_3=V_1 \cup V_2$ and $B_3=B_1 \cup B_2$ and can be written as $G_3=G_1 \cup G_2$.

A graph $G_1=(V_1,B_1)$ is said to be a subgraph of the graph $G=(V,B)$, if $V_1 \subseteq V$ and $B_1 \subseteq B$. Figure 3.4 illustrates this concept, where $G=(\{v_1, v_2, v_3, v_4\}, \{(1,2), (2,3), (3,4), (4,1), (1,3)\})$ and its subgraphs showed are

$G_2=(\{v_1,v_3,v_4\}, \{(1,3), (3,4), (4,1)\})$ and

$G_2=(\{v_1,v_3\}, \{(1,3)\})$.

The degree d(j) of a node v_j is the total number of links that are incident at v_j. For a directed graph the inward demidegree $d^-(j)$ of a node v_j is the total number of links that are incident into v_j, i.e., v_j is the terminal node of those links. The outward demidegree $d^+(j)$ of a node v_j is the total number of links that have v_j as initial node.

Clearly,

$$d(j) = d^+(j) + d^-(j)$$

$$\tfrac{1}{2}\sum_{v_j \in V} d(j) = \sum_{V_j \in V} d^+(j) = \sum_{v_j \in V} d^-(j) = |B|$$

where |B| represents the number of elements of the set of links, i.e., the number of links of the graph.

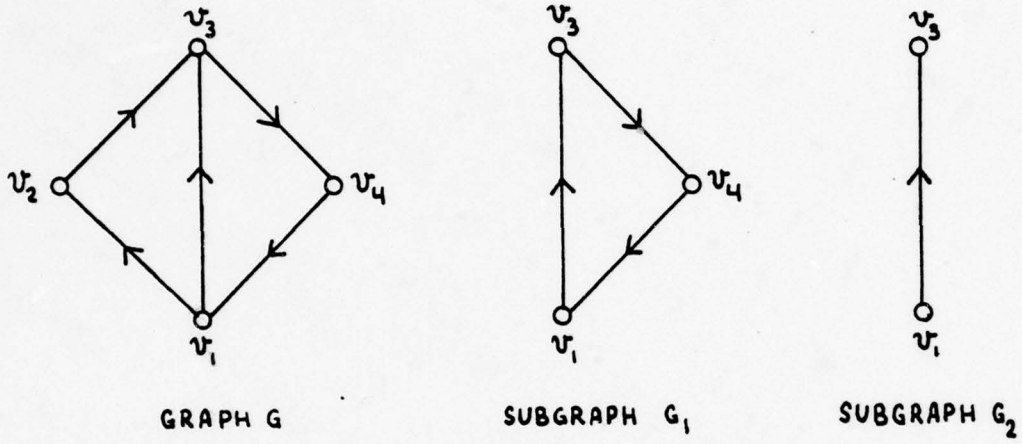


Figure 3.4 - Subgraphs of a graph

A self-loop is of the form (i,i) and contributes to both the inward demidegree $d^-(i)$ and the outward demidegree $d^+(i)$. Many authors don't consider self-loops in the treatment of graphs. Many times this is a matter of definition of the relation R which corresponds to the set of links of the graph. By considering that relation antireflexive, self-loops are excluded from the graphs. A relation R on a set S is said to be antireflexive if for every $x \varepsilon S$ $(x,x) \notin R$.

A graph is said to be homogeneous of degree x if for all $v_i \varepsilon V$, $d(i)=x$.

A directed path is a subgraph of G, specified by the sequence of nodes and links $v_{i_1}(i_1,i_2)v_{i_2}(i_2,i_3)\ldots\ldots$

$v_{k-1}(i_{k-1}),i_k$ such that in the subgraph, $d^-(i_1)=0$, $d^+(i_1)=1$, $d^-(i_k)=1$, $d^+(i_k)=0$ and for all i_j, $j=2,3,\ldots k-1$, $d^-(i_j)=d^+(i_j)=1$, i.e., all nodes and links are distinct.

An undirected path is a subgraph of G, specified by the sequence of nodes and undirected links $v_{i_1}\left[i_1,i_2\right] v_{i_2}\left[i_2,i_3\right]\ldots v_{i_{k-1}}\left[i_{k-1},i_k\right] v_{i_k}$ such that in the subgraph $d(i_1)=d(i_k)=1$ and for all i_j, $j=2,3,\ldots k-1$, $d(i_j)=2$. The definition of undirected graphs implies that $\left[i,j\right]=\left[j,i\right]$.

A path is a subgraph of G such that when link directions are removed, it is an undirected path.

A path is not necessarily a directed path when link directions are considered. Sometimes the sequence of nodes and links is represented simply by the sequence of nodes $v_{i_1}v_{i_2}\ldots v_{i_k}$ or by the sequency of links $(i_1,i_2)(i_2,i_3)\ldots(i_{k-1},i_k)$. Many times these sequences are referred simply as paths, the type of path being clear from the particular graph.

The graph G is said to be connected if for any pair of nodes there is a path in the graph. Otherwise the graph is disconnected.

A graph G is strongly connected if for any pair of nodes v_i and v_j there is a directed path from v_i to v_j and conversely for v_j to v_i.

A subgraph is maximal with respect to a property P if it loses P when more links or nodes are added to the graph. A maximal connected subgraph of G is called a component. The components of G define a unique partition of the nodes of G. When G is connected, it has only one component.

A directed cycle is a subgraph of G specified by the sequence of nodes and links $V_{i_1}(i_1,i_2)V_{i_2}(i_2,i_3)\ldots$ $V_{i_{k-1}}(i_{k-1},i_k)$ such that $d^-(i_j)=d^+(i_j)=1$ for all $j=1,2\ldots k$. In other words, a directed cycle is a directed path in which the first and last nodes are the same; it is a "closed" directed path.

An undirected cycle is a sequence of nodes and links with the same characteristics of an undirected path, except that the first and last nodes are the same.

A cycle is a sequence of nodes and links satisfying the requirements for a path, except that the first and last nodes are the same, or, in other words, a cycle is a subgraph of G such that when link directions are neglected it is an undirected cycle.

A graph that does not contain a cycle is said to be acyclic.

F. CUT-SETS, CUTS AND TREES

In dealing with problems of flows in networks and reliability, it is necessary to know what links and nodes contribute to the transmission of the commodity from one region of

the network to another. The total capacity of those links are to be considered in flow problems while in reliability problems the probability of failure of links and nodes must be additionally considered [Ref. 12, 22].

A set S is said to be minimal in relation to some property P if no proper subset of S has property P.

An undirected link cut-set of an undirected connected graph is a minimal set of links the removal of which yields a graph with two or more components. Figure 3.5 shows an example of cut-set. The removal of ([1,4], [2,3], [1,3], [2,4]) from the graph is 3.5 a) yields the graph in 3.5 b) with two components. No proper subset of this set yields a disconnected graph.

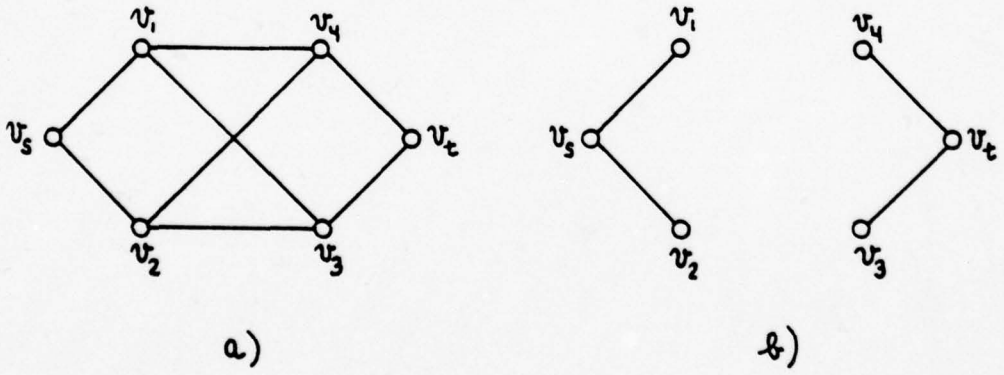


Figure 3.5 - Example of cut-set

When it is necessary to distinguish among the nodes that are separated by the cut-set, it is specified that an undirected cut-set of an undirected connected graph is an i-j

cut-set when the nodes v_i and v_j are in different components after the cut-set is removed from the graph. Thus, in the example of Figure 3.5 the cut-set ([1,4], [2,3], [1,3], [2,4]) is an S-4, S-t, S-3, 1-4, 1-t, 1-3, 2-4, 2-t and 2-3 cut-set.

A set of links of a directed graph is a directed link cut-set if its removal from the graph breaks all directed paths from at least one node of the graph and no proper sub-set breaks all directed paths between the same vertices. Figure 3.6 gives an example of directed link cut-set; the set {(1,4),(1,3),(2,3)} is a directed branch cut-set since

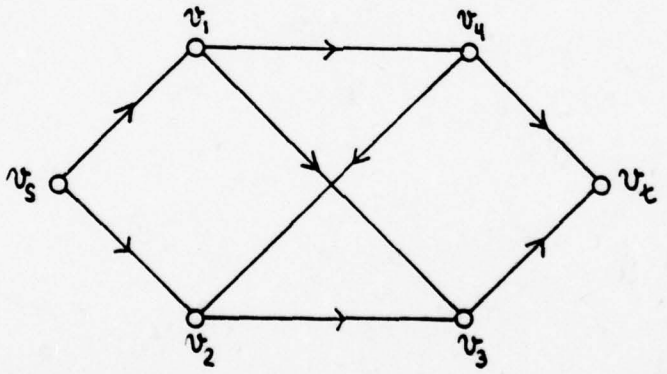


Figure 3.6 - Example of cut-set

its removal destroys all directed s-t paths and no proper sub-set does so.

Again, to specify the nodes which are affected by the cut-set removal, it is said that the cut-set is a directed s-t link cut-set if it is a minimal set of links of the

directed graph whose removal from G breaks all directed s-t paths.

The reliability of a network is not only affected by the removal of links; node removals must also be considered.

When a node v_s is removed from a graph, all links incident at v_s are also removed from the graph. An undirected node cut-set of an undirected connected graph G is a minimal set of nodes whose removal from G separates the graph into two or more components. In the example of Figure 3.5 $\{v_1,v_2\}$ and $\{v_3,v_4\}$ are both undirected node cut-sets since both break all paths between v_s and v_t; in this case it is said to be an undirected s-t node cut-set. The removal of an undirected s-t node cut-set from a graph breaks all undirected s-t paths. In Figure 3.7 the only undirected node cut-set is $\{v_1\}$.

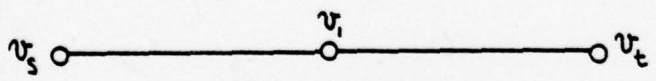


Figure 3.7 - Example of cut-set

A set of nodes of a directed graph G is a directed node cut-set if its removal from G destroys all directed paths from at least one of the remaining to at least one other remaining node and no proper subset breaks all directed paths between these vertices. A directed s-t node cut-set is a minimal set of nodes (not containing v_s or v_t) whose removal

from G breaks all directed paths from node v_s to mode v_t.

For a directed graph G, an s-t mixed cut-set is a minimal set of links and nodes, other than v_s and v_t. whose removal from G breaks all directed s-t paths.

For an undirected graph an s-t mixed cut-set is a minimal set of links and nodes not including v_s and v_t that, when removed, disconnects v_s from v_t.

Closely related to the concept of a link cut set is the concept of a cut. Loosely speaking a cut is a set of links that connects a set of nodes to the remaining nodes. This is illustrated in Figure 3.8 for an undirected graph.

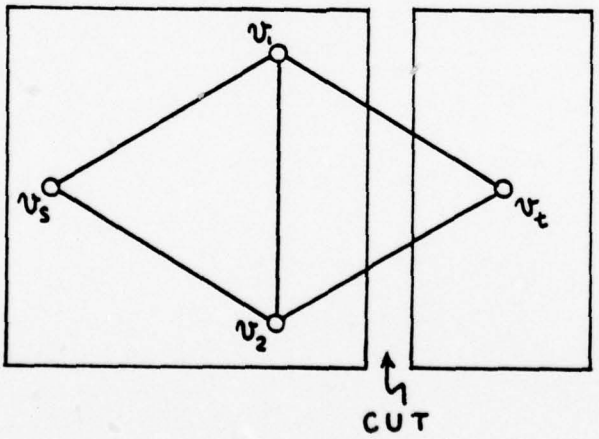


Figure 3.8 - A cut

More precisely let G=(V,B) be a directed graph. If $P \subseteq V$ and $Q \subseteq V$ (P and Q not necessarily disjoint), let (P,Q) denote the set of all links which are incident out of an element of P and incident into an element of Q, i.e.,

$(P,Q)=\{(i,j)\varepsilon B \mid v_i \varepsilon P$ and $v_j \varepsilon Q\}$. Given V_1, a subset of nodes of G, let $\bar{V}_1$ denote the set complement of V_1 in V. For any $V_1 \subseteq V$, the set of branches $(V_1,\bar{V}_1)$ is a cut. For example in Figure 3.6 if $V_1=\{v_s\}$ then $(\{v_s\}, \{v_1,v_2,v_3,v_4,v_t\})=\{(s,1),(s,2)\}$ is a cut, while if $V_1=\{v_s,v_1,v_2\}$ then the set of links $(\{v_s,v_1,v_2\}, \{v_3,v_4,v_t\})=\{(1,4), (1,3), (2,3)\}$ is another cut. Any of these cuts can be called s-t cuts since $v_s \varepsilon v$, and $v_t \varepsilon \bar{V}_1$. In general the notation $(X,\bar{X})_{s,t}$, where $X \subseteq V$ represents any cut in which $v_s \varepsilon X$ and $v_t \varepsilon \bar{X}$. The generalization for undirected graphs is simple: the set of branches $[P,Q]$ between two sets of nodes is defined as $[P,Q]=([i,j] \varepsilon B \mid v_i \varepsilon P$ and $v_j \varepsilon Q)$ and $[V_1,\bar{V}_1]$ is a cut.

The concept of cut is more general than that of directed cut-set as can be seen in the following theorem.

Theorem - Not every cut is a directed cut-set. Every directed s-t cut set is an s-t cut.

The same is valid for undirected graphics: every cut-set is a cut; not every cut is a cut-set.

Let G be an weighted graph in which the weight w(i,j) associated with the link (i,j) denotes the capacity of the link c_{ij}. The capacity of an arbitrary subset of links $B_1 \subseteq B$ is defined as the sum of all the capacities of the links of B_1. That is

$$C(B_1) = \sum_{(i,j)\varepsilon B_1} c_{ij}$$

where $C(B_1)$ denotes the capacity of B_1.

If the capacity of each link is unity, then $C(B_1)=|B_1|$ is the number of branches in B_1.

When an undirected link cut-set is removed, an undirected connected graph becomes disconnected, but if a proper subset of that cut-set is removed the graph remains connected. Different cut-sets have different numbers of links so the minimum number of links which can disconnect the graph when removed is not readily available, or the maximum number of links which don't disconnect the graph when removed is not available. Related to this last idea is the concept of a tree of a graph.

A tree T=(V,U) of a connected graph G=(V,B) is a connected subgraph of G which contains all nodes of G but no cycles [Ref. 22]. Figure 3.9 illustrates one possible tree for the connected graph of 3.9 a).

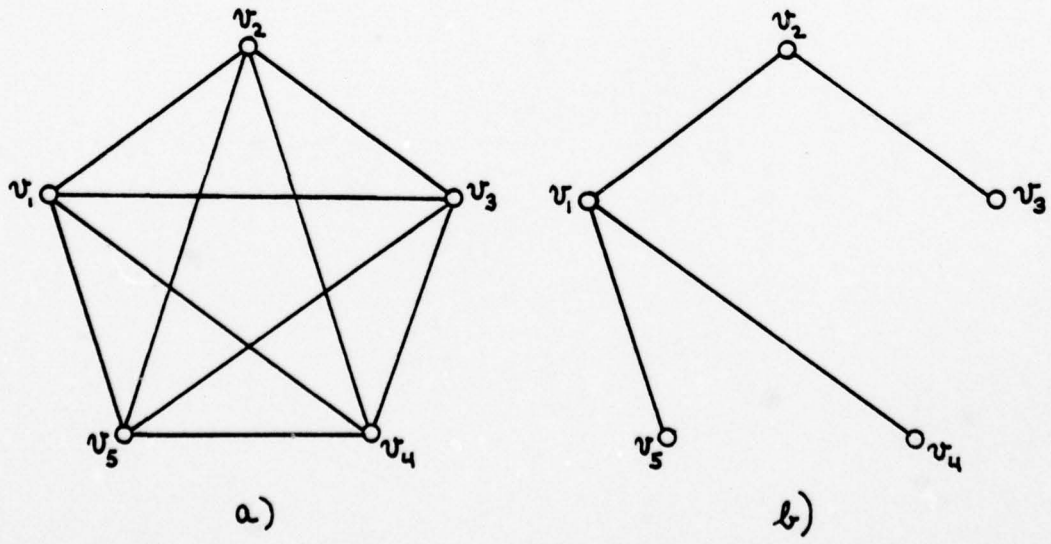


Figure 3.9 - Tree of a connected graph

A tree T=(V,U) of graph G=(V,B) with n nodes has the following properties:

1. U contains n-1 links.

2. T=(V,U) is connected but becomes disconnected if any link is removed.

3. Every pair of vertices v_i and v_j is joined by one and only one path.

4. Any subgraph of G which contains T as a proper subgraph, has at least one cycle.

5. Every undirected branch cut-set has at least one link in common with every tree.

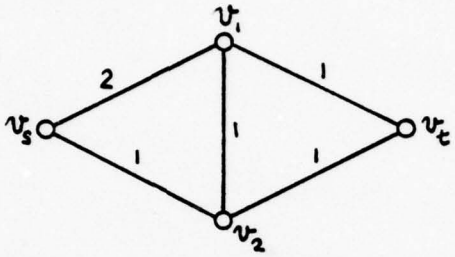
G. THE MAX-FLOW MIN-CUT THEOREM

Let us suppose it is necessary to transmit a commodity from a node v_s to a node v_t through a network which contains these nodes. We wish to determine the commodity flow which the network can accomodate. The max flow min-cut theorem allows us to solve this problem [Ref. 1, 12, 22].

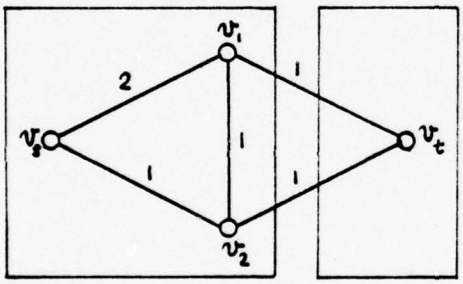
Theorem - The maximum flow from a node v_s to node a v_t is equal to the capacity of the s-t cut which has the minimum capacity among all s-t cuts.

Intuitively, if v_s and v_t are in different groups of nodes, then all flow from v_s to v_t must pass through the cut that connects them. Therefore, the maximum flow from v_s to v_t cannot exceed the capacity of the cut; in particular it cannot exceed the capacity of the cut of smallest capacity. Figure 2.10 illustrates the application of the

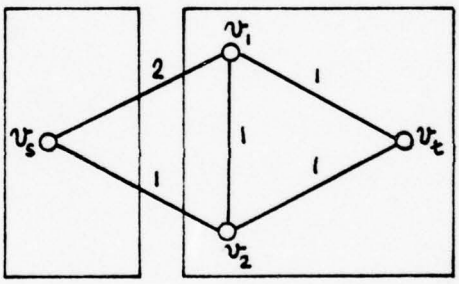
max-flow min-cut theorem to an undirected connected graph. The maximum flow from v_s to v_t is equal to two, the capacity of the s-t cut with smallest capacity.



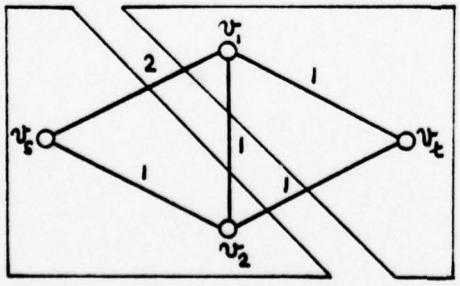
a) UNDIRECTED CONNECTED GRAPH



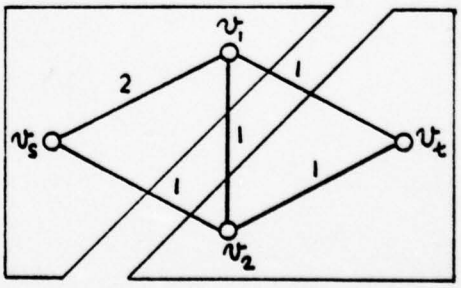
b) S-T CUT WITH CAPACITY 2



c) S-T CUT WITH CAPACITY 3



d) S-T CUT WITH CAPACITY 4



e) S-T CUT WITH CAPACITY 3

Figure 3.10- Illustration to the max-flow min-cut theorem

IV. COMPUTER NETWORK RELIABILITY

A. INTRODUCTION

A factor of major importance to the successful operation of computer networks is their reliability. This factor becomes increasingly significant as networks grow in size and their users become more dependent on their proper operation. Accordingly, one of the most important objectives in designing a network is to guarantee that it will function effectively even after some of its elements fail. Network reliability is strongly dependent on the topological layout of the communication links in addition to the reliability of the individual computer systems and communication facilities [Ref. 12, 22, 52].

The computer-communication network model that is generally used for reliability analysis is an undirected graph. It has the advantage of simplifying the analysis while it correctly represents most of the networks.

Failures may occur in both nodes and links of a network. Careful design and duplication of equipment can significantly reduce failure in the nodes at the expense of increased cost. Failures in links are not as controllable. To overcome link failures, networks can be designed with duplication of links, which may be very expensive, or may incorporate alternate routes between any pair of nodes, which may be more cost-effective.

The first step in studying the reliability of a system is to formulate precise criteria for its failure. These criteria will depend on the network function and its purpose. Reliability analysis of networks in concerned with the dependence of the network reliability on the reliability of its nodes and links. Reliability of a node or of a link may be easily defined as the probability of failure in time. The reliability of a network is much more difficult to define. Measures of the reliability of a network may be for example /Ref. 23/: a) the number of elements that must be removed before the network becomes disconnected, i.e., before one pair of nodes in the network cannot communicate; b) the probability that the network become disconnected; c) and the expected number of node pairs that can communicate through the network. Another class of measures arises when the importance of the nodes in the network is not the same; some nodes may be more important than others due to nodal functions or resources.

B. THE FORD-FULKERSON ALGORITHM

The Ford-Fulkerson algorithm is used for calculations of flow in the one commodity case. Given a network, such as the one in Figure 4.1-a), the algorithm can calculate the maximum flow that is possible from one node to another. The application of the algorithm to reliability calculations is for counting the minimum number of links in the smallest cut between a pair of nodes.

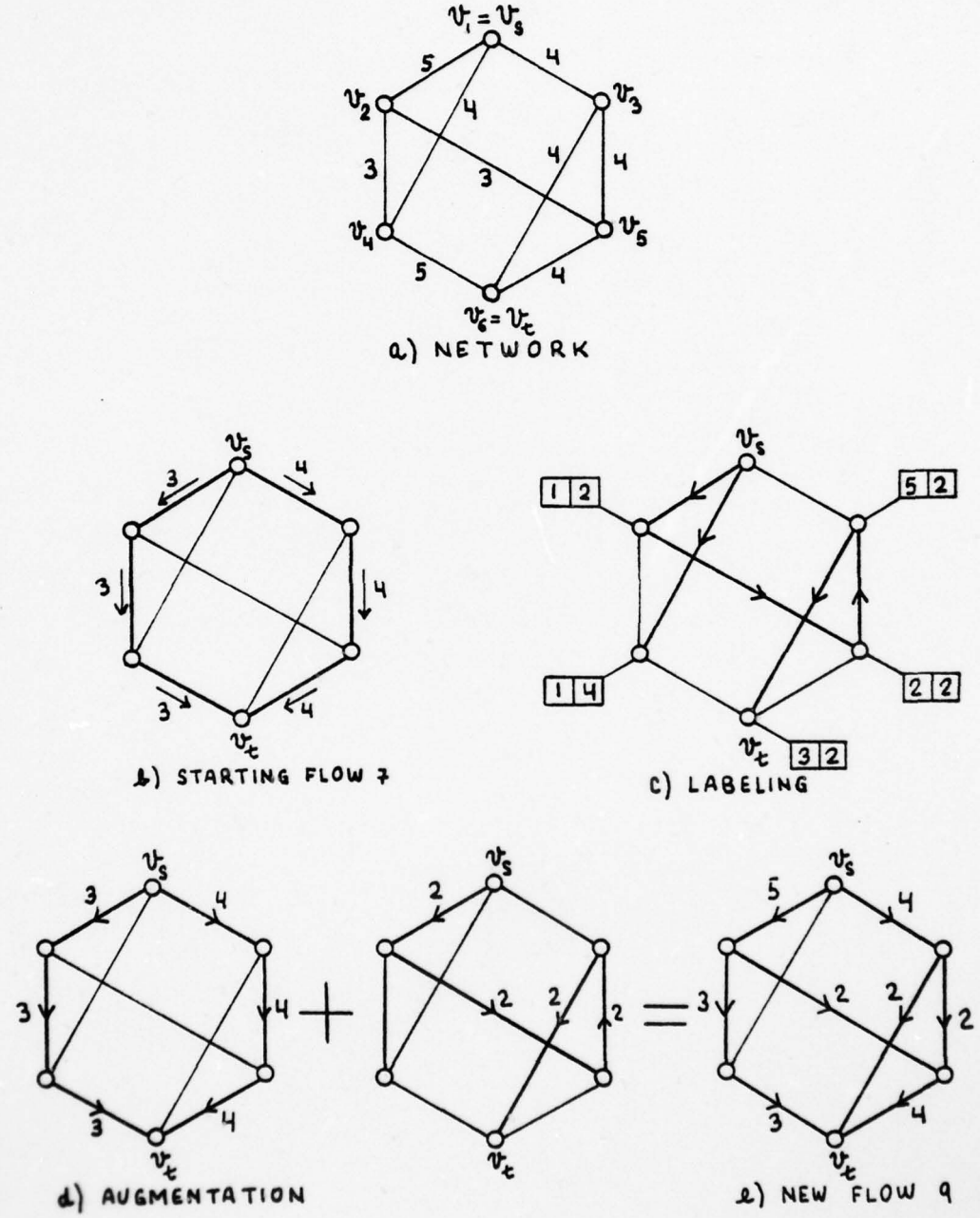


Figure 4.1 - The Ford-Fulkerson algorithm

The algorithm can be well illustrated by means of an example /Ref. 12/. Consider the weighted undirected graph of Figure 4.1a) where the weights represent the capacity of the links. In part b) a simple flow pattern is shown, one which clearly does not exceed the capacity of any link and carries 7 units of flow from v_s to v_t. The augmentation of this flow is shown in part d) in which a path from v_s to v_t has been chosen and an extra flow of 2 units added along that path. The result is a new flow pattern in e) which has nine units passing from v_s to v_t. The difficulty of the algorithm lies in finding the path along which the extra flow can be inserted. This part of the algorithm is based on labeling the nodes, and is shown in part c) of the figure.

The labeling process starts at v_s, each direction is tried in turn. Direction 1-2 would allow 2 more units of flow out of v_s so it is a possible start for the path. To note this step, a label (1,2) is attached at node v_2 with two items of information: the node v_1 from which the flow from v_s might be received and the extra flow 2. The next direction to be tried is 1-4, where an extra flow of four units out of v_i is possible and this is shown on label (1,4) at node 4. The third direction towards node 3 allows no extra flow and no label is attached to node 3 at this stage. Now node v is left out and it is marked as 'scanned' so that the labeling algorithm will not come back to it.

To proceed with labeling, labeled node v_2 is examined

and the two directions of exit from it explored. The direction of node v_4 is of no interest because that node is already labeled. A flow of three of 3 units towards node v_5 is possible, which results in labeling node v_5 with (2,2) to denote the flow of two which could be received via node v_2 from v_s. At this stage, node v_2 has been 'scanned' and is out of the running. At the next stage node v_5 is chosen to be examined. It is important to note that if node v_4 were chosen a different result would have been obtained. But the object is only to find a path from v_s to v_t to augment the flow and where there are alternatives either one will do. The next node to be labeled is v_3, then finally v_6, which is the terminal node. If there is a path for the flow to the terminal it will be found, though it may not be the most direct path.

To complete the path-finding phase the path is retraced from v_t backwards. The possible flow, which started out as 2, might have been reduced at any stage due to limitations on the capacity of the links. The final flow value of 2 is shown on the label (3,2) at v_t and this is the possible amount of the augmentation of this stage. The node references on the label allow the path to be retraced from v_t as v_t-v_3-v_5-v_2-v_s. Then this flow is added to the initial flow pattern as shown in 4.1d and the new flow is 9 as shown in 4.1e.

Now the whole labeling and augmentation process is repeated. In Figure 4.2 the next flow of 11 was found and then the maximum of 12 was reached. The approach to the maximum

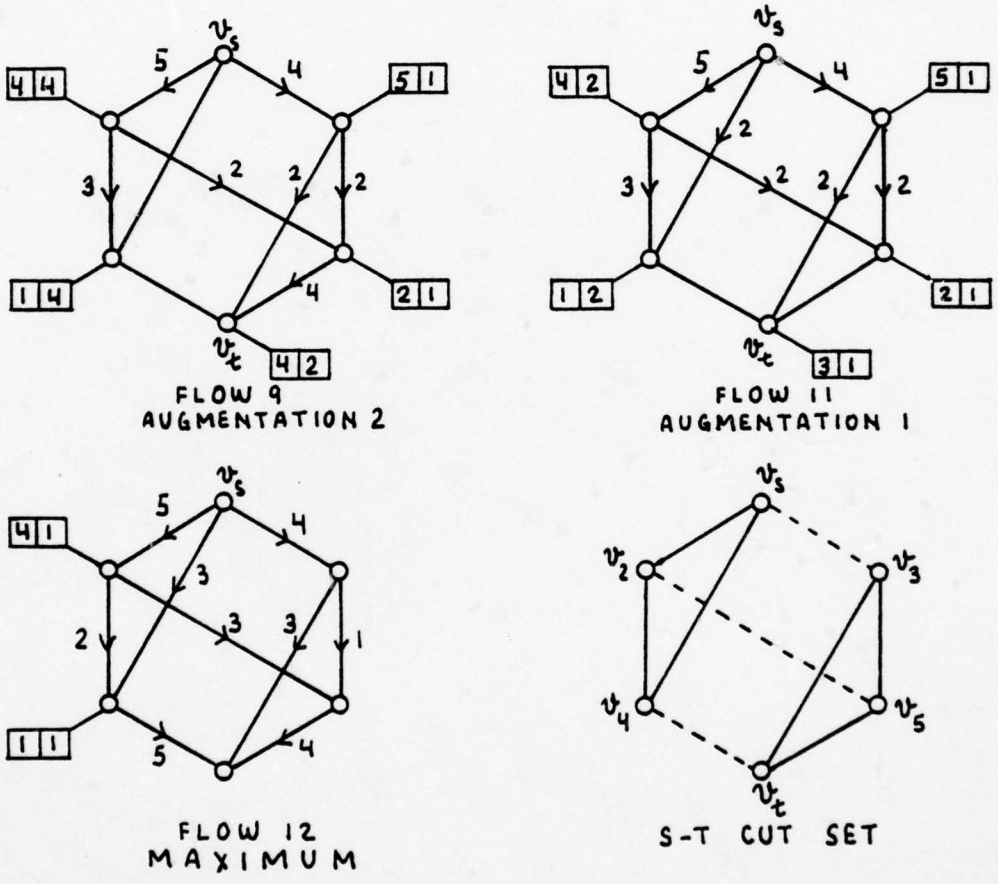


Figure 4.2 - The maximum s-t flow and cut-set

would be different if different augmentation paths were used, but the resultant maximum flow would be the same. At the final stage the process could not continue because in

the labeling process the tree extending from v_s could not reach v_t and this marks the end of the maximization of the flow. Two nodes have been labeled. Together with node v_1 they form the only part of the network accessible to an augmenting flow. From this set of nodes v_1, v_2 and v_4 all the outgoing links (1,3), (2,5), and (4,6) are saturated. If follows that the set in question divides the network and is minimal; thus it is a link cut-set. It is also an s-t cut-set (and consequently an s-t cut) since it breaks all paths between v_s and v_t. Then the augmentation process stops because v_s and v_t are separated by a cut of saturated links.

The Ford-Fulkerson algorithm determines the maximum flow that is possible between v_s and v_t and shows one pattern of flow which achieves this maximum. It is not always a unique pattern. The algorithm also determines an s-t cut which carries the full flow capacity in all links of the cut when the s-t flow is maximum.

C. PROBABILITY OF DISCONNECTION BY LINK FAILURES

Given the network of Figure 4.1a), let p be the probability of failure of a link. Assuming that the failures of links are independent, the probability that a set of m links be out of service is $p^m(1-p)^{9-m}$, since the network has 9 links. To know the probability that v_s and v_t are not connected, it is enough to sum expressions like that for all

failure patterns that disconnect v_s from v_t. This would involve the examination of 2^9=512 subsets of the set of nine links.

In general, if a network has n links then the number of sets which have i links is ${}_nC_i$. Out of this number only A_i sets disconnect v_s from v_t if removed from the network. It is important to note that A_i includes not only the s-t cut-sets but all those which have the s-t cut-sets as a sub-set. If P(s,t) represents the probability of disconnection of v_s from v_t due to link failures:

$$P(s,t) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-1}$$

If the probability of any disconnection is needed, then A_i must account for the number of sets with i links which cause any disconnection, instead of only those which disconnect v_s from v_t.

In real computer networks p is very small [Ref. 12, 23, 52]. Concentrating on very reliable networks, the calculation of an approximate result for the expression above is simplified because the first non-zero term is dominant. If the number of elements of the smallest s-t cut is denoted by $\Theta(s,t)$, the the first non-zero term in the equation has $i=\Theta(s,t)$.

For small networks $\Theta(s,t)$ can be derived easily by inspection, but this is not the case for large and heavily

connected networks. The Ford-Fulkerson algorithm can be used for this calculation. It is necessary then to assign the capacity 1 to all links in the network and derive the maximum flow from v_s to v_t. This value is equal to the number of links in the smallest s-t cut by the max-flow min-cut theorem.

The probability of v_s and v_t being disconnected by link failures is proportional to $p^{\theta(s,t)}$ if p is small. In this case the smallest value of $\theta(s,t)$ among all pairs of nodes, denoted by θ, is a measure of the vulnerability of the network to link failures. This quantity θ is the smallest number of links that have to be removed to cause some disconnection in the network.

It can be proved that the quantity $\theta(s,t)$ is the maximum number of link disjoint paths between v_s and v_t. Two paths are link disjoint if they have no link in common, though they could have common nodes.

D. NODE FAILURES

Node failures are analyzed in a slightly different way than that of link failures and it is convenient to study the effect of mixed failures, i.e., those involving links and nodes. The main concepts are those of node cut-set and mixed cut-set.

The number of elements in the smallest s-t node cut-set represents the minimum number of nodes that must be removed from the network to break all s-t paths. This number is

denoted by v(s,t). It is possible to prove that [Ref. 12, 22]

$$v(s,t) \leq \theta(s,t).$$

The node cut-sets and the related connectivity measures v(s,t) have the disadvantage of being unable to handle nodes that are adjacent, because v(s,t) is not defined. The measure of vulnerability of the whole network to node dropouts is the number v, the least of all v(s,t) in the network, which is the minimum number of nodes to be removed to make the network disconnected. When the network is fully connected no v(s,t) is defined but, by convention n=N-1 for a network with N nodes [Ref. 12, 52].

In the same way that $\theta(s,t)$ and v(s,t) were defined, it is possible to define u(s,t) for mixed cut-sets. It is the minimum number of links and nodes which disconnects v_s and v_t, when removed from the network. This number is more useful than v(s,t) because it is defined for any pair of nodes and it can be proved that [Ref. 12, 22]

$$v(s,t)=u(s,t)$$

wherever v(s,t) is defined.

In the same way, u is the minimum of the numbers u(s,t) of all node pairs in the network. Again it can be proved that $u = v$ including the case of fully connected network of N nodes where u=v=N-1.

Analagous to $\theta(s,t)$ and link disjoint paths, u(s,t) is the

minimum number of node disjoint paths between v_s and v_t. Two paths with v_s and v_t as extreme nodes are called node disjoint if they do not have nodes in common other than v_s and v_t. Of course, since two paths that are node disjoint are also link disjoint, the number of node disjoint paths is included in the number of link disjoint paths

$$u(s,t) \leq \theta(s,t)$$

The numbers u(s,t) are the best way of specifying the connectivity property for reliability analysis purposes. If only $\theta(s,t)$ is specified there is the possibility that a node or mixed failure could break the s-t paths. It is important to note that, because $u(s,t) \leq \theta(s,t)$, node failures are more likely to disconnect non-adjacent nodes, if the probability of failure in nodes is of the same order as links: however, in networks this is not the case [Ref. 12]. Thus, u(s,t) should be specified for all node pairs or the value of u should be established.

E. KLEITMAN'S METHOD

To verify that u=x in a network involves the verification that $u(s,t) \geq x$ for all node pairs. Using the Ford-Fulkerson algorithm and assigning unity capacity to each link and node other than the source and terminal, it is possible to derive u(s,t) for every pair of nodes. But this requires a maximization procedure for every s-t pair and would not be practical for large networks. The method due to Kleitman does this in an economical fashion [Ref. 1, 12].

Recall that u(s,t) is equal to the number of node disjoint paths between v_s and v_t. First choose any node v_1 and verify that $u(1,t) \geq x$ for every other node v_t; that is, there are at least x node disjoint paths from v_1 to any other node. If successful delete v_1 and all links incident at it. Then choose another node v_2 and verify that $u(2,t) \geq x-1$ for any other node v_t. If successful, repeat this procedure for a thired node v_3 and the condition $u(3,t) \geq x-2$. Since x is usually a small integer the process will soon stop. To test that $u(x,t) \geq 0$ it is only necessary to verify that the graph is connected.

To appreciate the economy of the method, suppose it is necessary to analyze a 1000-node network to verify that $u \geq 3$. This network has 500,000 s-t pairs and, without the method, would require 500,000 flow maximization calculations to do the job. Using the method would require 999 tests for $u \geq 3$, 998 tests for $u \geq 2$, 997 tests for $u \geq 1$ and no test for $u \geq 0$. A total of 2994 flow maximization calculations would be performed, smaller by a factor of more than 100.

V. TOPOLOGY

A. SYSTEM CHARACTERISTICS

Many characteristics of computer networks are determined or influenced by their topology. Some qualitative attributes can be inferred directly from the topology of the network independently of the particular implementation /Ref. 27.

For the purpose of comparison among the various types of networks some attributes will be defined as in the following.

Modularity, the ability to make incremental changes in system capability, can be viewed in two ways. First, the cost of adding an element to the system that will be called cost-modularity and second, the degree to which the place and function of the incremental element are restricted, will be called place-modularity. For instance, if for adding one more processor to the system, the incremental cost is that of the processor, the network exhibits a very good cost-modularity; but, if besides the acquisition of the processor it is necessary to install a communication link to each of the processors of the network, the system has a poor cost-modularity characteristic. There may be places in the network where it is easy to increase a specific performance characteristic by the addition of physical resources (some piece of equipment) and other places where it may be difficult or even impossible; in the former case the system has good place-modularity; this is not true for the later case.

Connection flexibility measures the degree of freedom in adding an element (piece of hardware) to the network. In some systems there is no choice of how to connect a new element, while in others there may be alternatives with different costs.

One important aspect of systems is the ability for graceful degradation. In some systems the failure of one element may halt the operation, while in others degraded modes of operation are possible. The failure-effect is a measure of the consequences of a fault in the network. The cost of graceful degradation is the cost of alternative methods for masking the faults to allow operation in a degraded mode. The cost of graceful degradation is low in systems where there is minimal spare hardware and no explicit reconfiguration is required and is very high where duplication of elements is necessary and complex reconfiguration schemes have to be used. Of course, with minimal spare hardware available, there is little capability for graceful degradation.

In many systems there are inherent limitations in performance due to either the sharing of resources or non-uniformity of communications within the network. This problem is referred to as one of bottlenecks.

The architecture of a system is a major factor in determining the number and nature of the decisions that must be made to perform communications within the system. Logical complexity is a measure of the intricacy imposed by the

type of network on the decisions that have to be made by source, destination or switching elements during the communications process.

B. CLASSIFICATION

Several schemes exist for classifying computer-communication networks. Probably the simplest one divides them into centralized and distributed systems, according to the extent to which processing power and data base management are distributed among the collection of host computers /Ref. 41/.

Centralized networks provide a single host computer to service all users. In distributed networks, a variety of host computers may be accessed by network users. In simple words, centralized networks have one host computer, while distributed networks have two or more host computers. However, the presence of two or more host computers does not necessarily mean that the network is distributed.

C. CENTRALIZED NETWORKS

The most common architectures for centralized systems are the star and the tree shown in Figure 5.1a) and 5.1b). Less commonly used is the loop configuration of Figure 5.1c) /Ref. 41/.

The star and tree configurations are both trees in the sense of graph theory and thus have the same attributes with slight differences in the rank of some. The logical

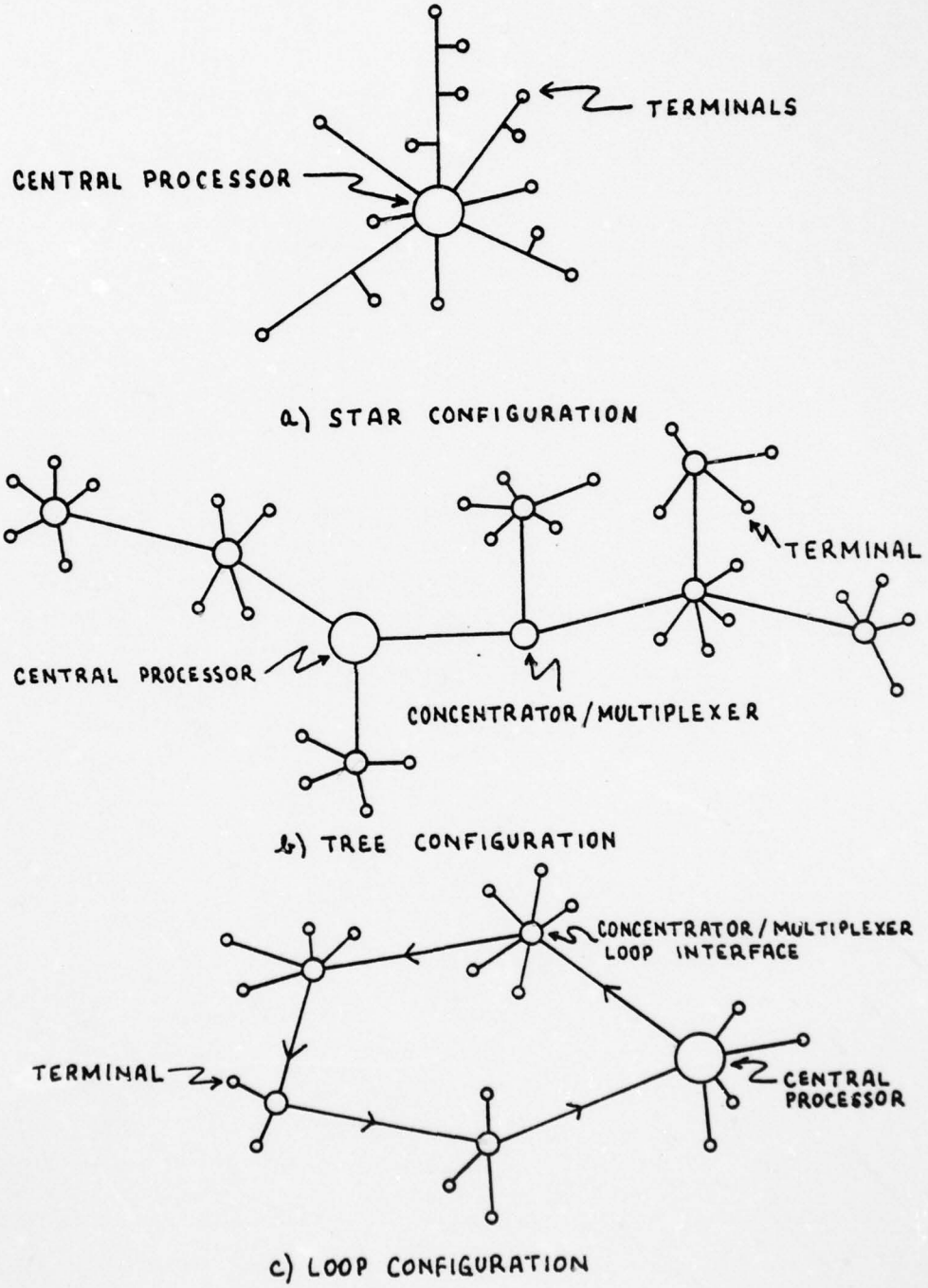
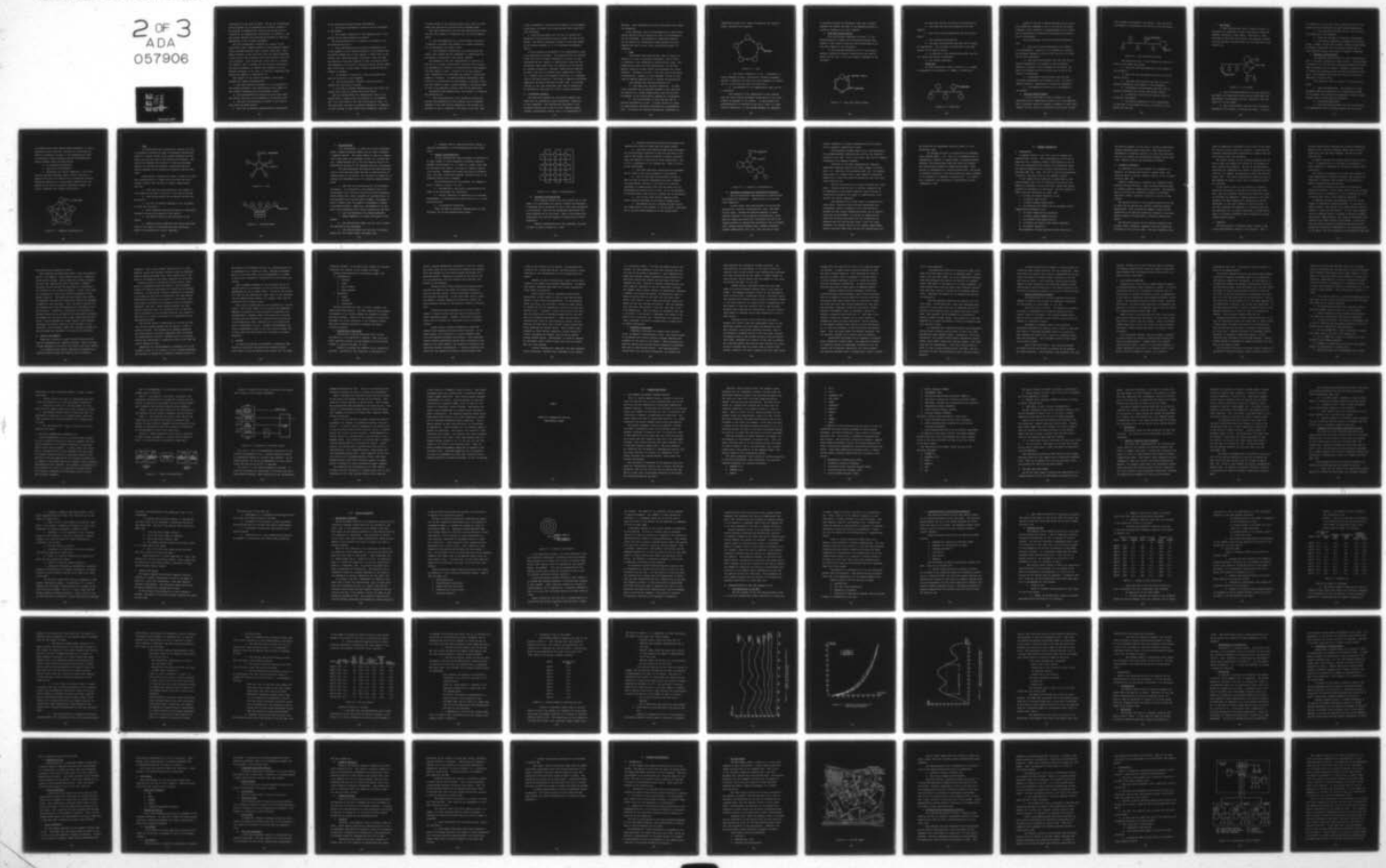


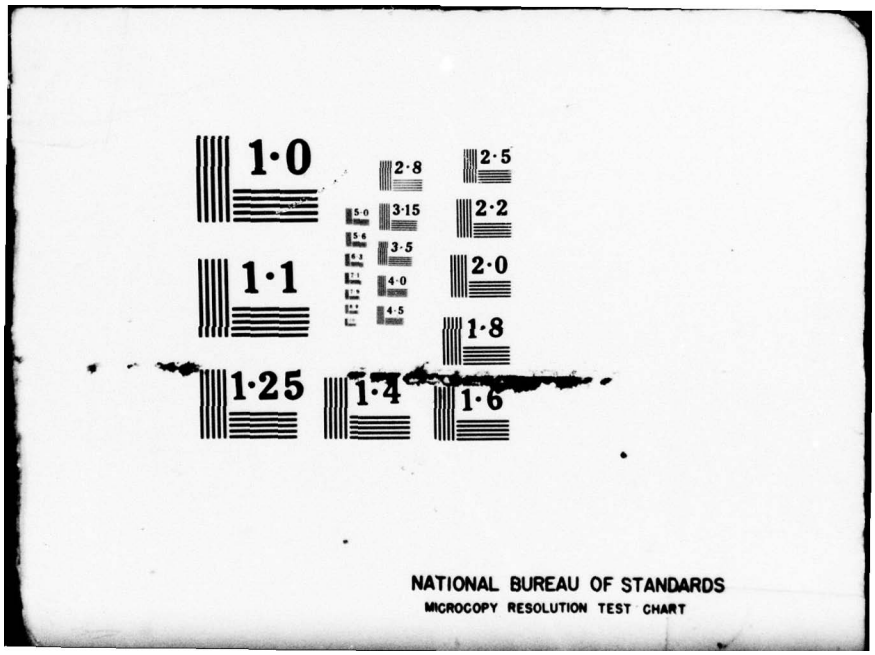
Figure 5.1 - Configurations for centralized networks

AD-A057 906 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 9/2
COMPUTER NETWORKS. ANALYSIS AND A CASE STUDY DESIGN.(U)
JUN 78 I D ROCHA

UNCLASSIFIED NL

2 OF 3
ADA
057906




1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

organization is the same for both. The use of concentrators and multiplexers is an implementation decision related to the economy in communication costs in the network and influenced by the geographical distribution of terminals. Remote access networks and time-sharing systems are typically implemented in one of these topologies.

The loop configuration consists of a series of unidirectional links (simplex channels) interconnecting sequentially arranged stations along a single closed path. Messages circulate in the loop going from the central processor to each station and conversely from the stations to the central processor. Each station may contain one or a cluster of terminals. When a message arrives the loop interface of the station examines its address and transmits it to its adjacent processor in the chain or retains it depending upon whether the message is addressed to it.

With respect to the central processor, centralized systems have common characteristics:

1. poor cost-modularity because it is not possible to add another processor; this characteristic will depend on the internal organization of the processor itself;

2. poor place-modularity because the only place where the central functions can be enhanced is the central site;

3. very poor failure-effect-a failure in the central site stops the whole system;

4. high cost of graceful degradation-only duplication

of the processor provides graceful degradation;

5. the central processor is prone to be the bottleneck of the system;

6. the logical complexity of the communications is low because of the centralization of control.

The star configuration, in relation to terminals, has the characteristics of:

1. fair cost-modularity and good place-modularity up to the capacity of the central processor; a new terminal may be placed anywhere but it will require a direct line to the central site; where multipoint (multidrop) lines are used, the addition of one more terminal to that line may be difficult depending on the control structure and the control equipment of the line; in some cases a direct connection may be simpler or cheaper;

2. no connection flexibility, since the connections must all be direct to the computer;

3. low failure-effect, because a faulted terminal affects only the place it serves;

4. low cost of graceful degradation-the only action the system has to take is isolate the failed branch.

Also the failure-effect of communication links is not severe when it connects only one terminal to the computer, yet it is a bit worse in the case of multipoint lines. On the other hand, the cost of graceful degradation is moderate to high, because in some cases dialed telephonic lines may

be substituted for the failed private lines, while in other cases only duplication provides better degraded modes.

The tree configuration has the same characteristics that the star with respect to terminals with two slight improvements:

1. the cost-modularity is a little better, because it is possible to connect the terminal to a nearby concentrator or multiplexer with short lines;

2. the connection flexibility is a little better because a new terminal may be connected to a concentrator or multiplexer or directly to the central computer.

The drawback of the tree configuration in relation to the star is its poor failure-effect and high cost of graceful degradation in relation to concentrators or multiplexers and high capacity links. A failure in one high capacity line, concentrator or multiplexer may disable a significant number of terminals. To improve graceful degradation duplication of concentrators or multiplexers is required, while for high capacity lines a dialed telephonic line may be used. If this is not possible, private lines can be duplicated.

Centralized loop networks have, with respect to terminals the attributes of:

1. fair cost-modularity, fair connection flexibility and good place-modularity-a new terminal may be connected anywhere in the loop, may be connected to the nearest concentrator or may be inserted in the loop; in the last case

a cost is imposed in the form of the addition of one communication path, a reroute of an existing path and a need for a loop interface;

2. good failure-effect and low cost of graceful degradation-a failed terminal does not affect the rest of the system. The central processor isolates it from the network by not sending messages to it or retransmitting messages from it.

In loop systems the bandwidth of the communication links is an additional bottleneck. The characteristics of failure-effect and cost of graceful degradation are poor in loop architectures with respect to communication links and loop interfaces, where faults halt the operation of the system. To increase reliability, redundant communication paths may be used in the form of bidirectional communication links, i.e., another loop in the opposite direction, or a redundant loop in the same direction. Two loops in opposite directions complicate the design and have not been used. To overcome failures in the loop interfaces, some type of bypass must be activated in the event of a fault in those interfaces.

D. DISTRIBUTED NETWORKS

The definition adopted for distributed computer networks does not necessarily imply geographical dispersion of host computers. The architectures discussed in the following sections may or may not have computers spread through a geographical region; that is an implementation

decision. Some topologies have better attributes than others for dispersion.

In the following, only the characteristics of each architecture derived from the particular type of interconnection will be discussed. The problems of connecting terminals are the same as for centralized systems because each host computer may have its own "local centralized network" of terminals.

1. Loop

Distributed loop architectures, seen in Figure 5.2, consists of a set of individual processors, each of which is connected to two neighbors by unidirectional links. The communication paths could be bidirectional, but the complexity caused by two-way traffic has prevented this implementation. Messages circulate in the loop going from source to destination, stopping at each processor, having its destination address examined until it reaches the destination.

Loop systems have the characteristics of:

a. Very good cost and place modularity. An additional processor can be inserted anywhere in the loop with the addition of a single communication link and the flow of messages is not significantly affected by its presence;

b. The failure-effect is poor and the cost of graceful degradation is high. A single failure in a path or loop-processor interface interrupts the intercommunication. To provide for graceful degradation, redundancy of

communication paths and a means of bypassing the loop-processor interfaces are required;

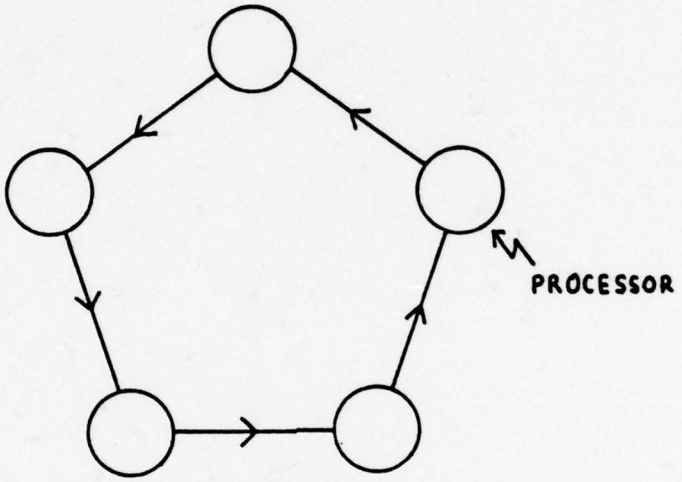


Figure 5.2 - Loop

c. The logical complexity is low. A processor receives messages through a single port, transmits messages through a single port and must only relay messages not addressed to it and strip off those destined for it;

d. Low bandwidth in the communication paths may be a bottleneck.

The bandwidth of the communication links together with the relay through processors causes delay in the transmission of messages in the network. In some systems the place-modularity is enhanced by the use of "soft" or "associative" addressing; in this method messages are addressed

to processes instead of processors; each loop interface examines the address and sees if the addressed process is currently residing in its host computer.

2. Loop With Central Switch

This architecture displayed in Figure 5.3, has characteristics in common with centralized systems with respect to the central switch and the characteristics of a loop with respect to the processors.

In this topology messages are put in the loop by senders, removed for address translation by the central switch and put back in the loop properly addressed to the recipient.

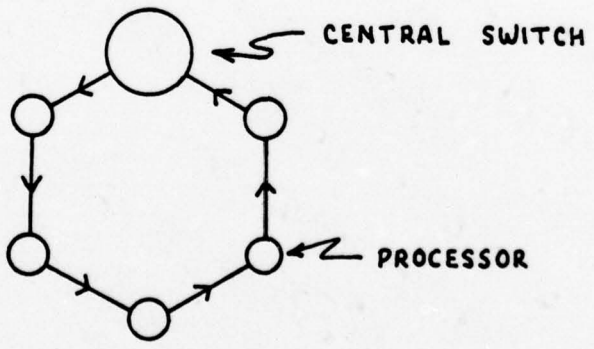


Figure 5.3 - Loop with Central Switch

The loop with central switch has the attributes of:

a. very good cost and place-modularity for processors;

b. poor cost and place-modularity for the central switch;

c. fair connection flexibility;

d. very poor failure-effect and high cost of graceful degradation. The situation is worse than in the loop because of the central switch;

e. in addition to the communication paths, the central switch may become a bottleneck;

f. low logical complexity.

3. Global Bus

In this architecture, shown in Figure 5.4, a number of processors are connected to a common, or global bus.

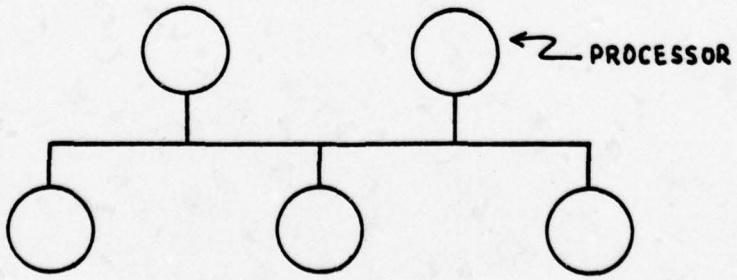


Figure 5.4 - Global Bus

Access to the bus is shared according to an allocation scheme and a message is put on the bus by the source processor, which transmits it simultaneously to all others, and is received by the destination processor upon recognition of the address.

Bus architectures have the following characteristics:

a. good cost and place-modularity with respect to the processors. Generally it is possible to connect a new processor anywhere to the bus with little or no effect on the other processors;

b. very good failure-effect and very low cost of graceful degradation with respect to the processors- generally, failures in the processors do not affect the rest of the system and do not require any action by the system to reconfigure;

c. catastrophic failure-effect and high cost of graceful degradation with respect to bus. In order to improve viability, replication of the bus is necessary;

d. the bandwidth of the bus is a bottleneck in the network.

4. Bus With Central Switch

In this architecture, shown in Figure 5.5, the processors do not communicate directly as in the global bus. When a processor wishes to transmit a message, it must first acquire the bus, which is controlled by the central switch,

then transmit the message to the switch. From the switch the message is transmitted over the bus to the destination.

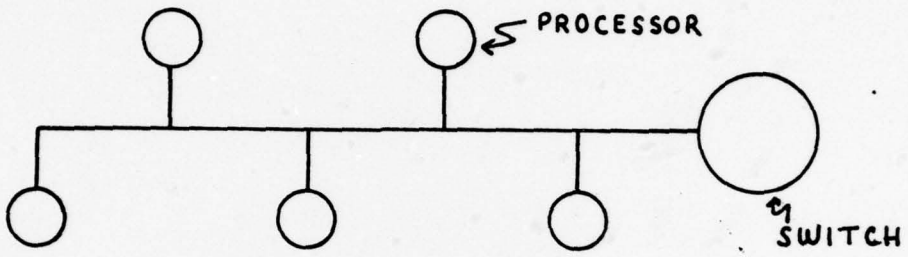


Figure 5.5 - Bus with central switch

The characteristics of this topology are similar to those of the global bus system:

a. good cost and place-modularity with respect to the processor;

b. poor cost and place-modularity for the bus and central switch;

c. very good failure-effect and very low cost of graceful degradation with respect to the processors;

d. catastrophic failure-effect and high cost of graceful degradation for the bus and central switch;

e. in addition to the bus the central switch is a potential bottleneck;

f. the logical complexity of this architecture is less than in the global bus due to the control of the switch over the bus.

5. Bus Window

In this architecture, which is seen in Figure 5.6, processors are connected to buses controlled by switching elements connected to other buses. Switching is, thus, performed by more than one resource, and messages received from one bus may be retransmitted onto the same bus or onto another bus.

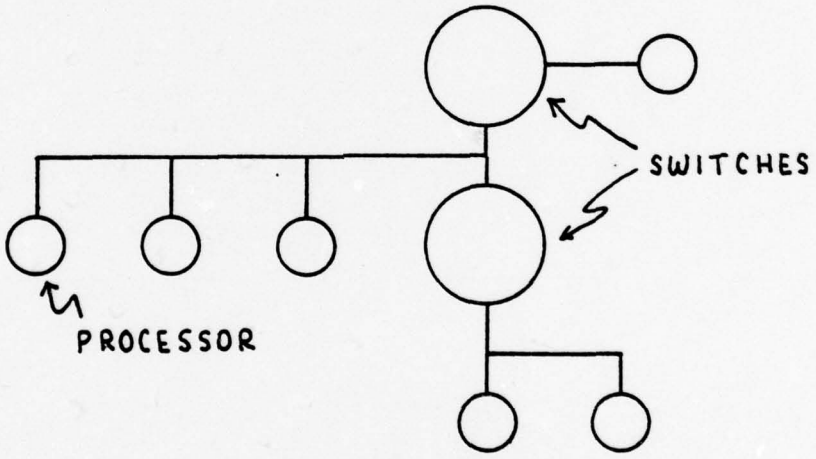


Figure 5.6 - Bus Window

a. very good place-modularity and cost-modularity-processors and communication paths may be added where and when needed, and processors are only required to have one connection to the system;

b. poor failure-effect and high cost of graceful degradation with respect to switches and buses. A failure

in a switch or in a bus can stop a significant part of the system and only replication can add graceful degradation;

c. very good failure-effect and low cost of graceful degradation for processors;

d. buses and switches are potential bottlenecks;

e. the logical complexity increases as the system grows in number of buses and switches, requiring more levels of address translation; also the logical design must be done carefully because this system is subject to deadlock.

6. Complete Interconnection

The complete interconnection or fully connected network is the most conceptually simple network. Each processor is connected to each other by a dedicated communication path. When a processor has to transmit a message to another one, it chooses the appropriate port from the alternatives available. All processors must have the ability of receiving messages at multiple ports.

This topology, shown in Figure 5.7, has the attributes:

a. poor cost-modularity. The addition of a new processor requires connections to all processors already in the system; this requires the processor to have at least one port free to accept the new processor;

b. good place-modularity;

c. very good failure-effect and low cost of graceful degradation-since a failed processor does not affect

the communication paths between other processors, no reconfiguration action other than isolation of the failed processor is necessary; failures in the communication links are handled by routing messages through intermediary processors between source and destination;

d. no connection flexibility;

e. there are no bottlenecks;

f. relatively low logical complexity. Due to the need for switching messages after a failure occurs in a communication path, the design has to incorporate location-addressing capability for interprocess communication; depending on the level of graceful degradation needed, the logical complexity may increase substantially.

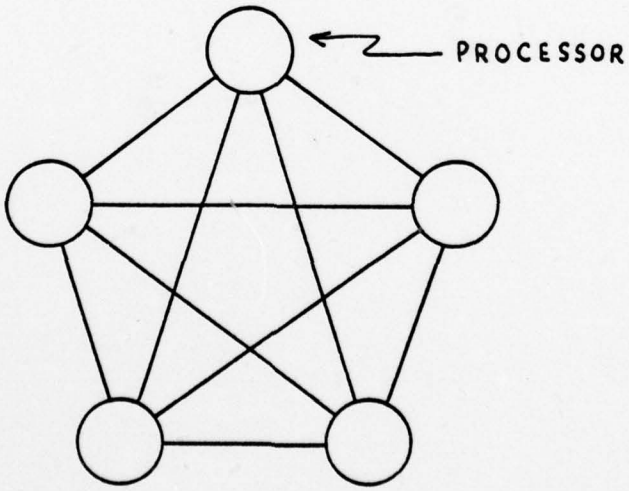


Figure 5.7 - Complete interconnection

7. Star

The distributed star architecture consists of a set of processors connected by means of dedicated bidirectional paths to a central switch, which accepts messages from the source and delivers them to the destination processor. The switch also performs the function of isolating processes running on a particular processor, preventing them from having knowledge of the system and protecting them from each other.

Star systems, schematically shown in Figure 5.8, have most characteristics in common with the global bus architecture, because they too share a central communication facility:

a. good cost and place-modularity with respect to the processors and poor in relation to the central switch;

b. good failure-effect for processors and poor for the switch;

c. low cost of graceful degradation for processors and high for the switch;

d. poor connection flexibility because an additional processor can only be connected to the switch;

e. the central switch is the bottleneck of the system;

f. moderate logical complexity-the switch must keep track of the status of the system and store addressing tables for translation of logical addresses.

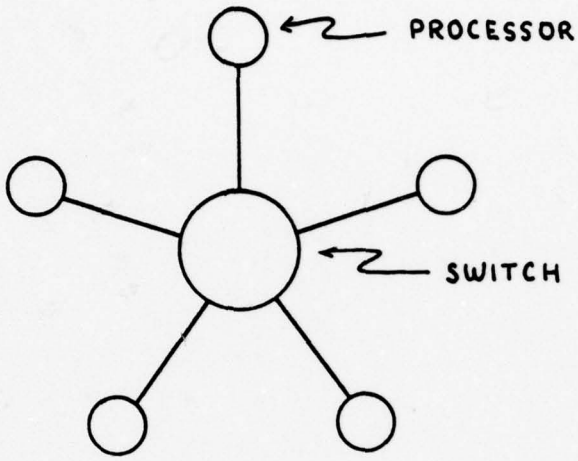


Figure 5.8 - Star

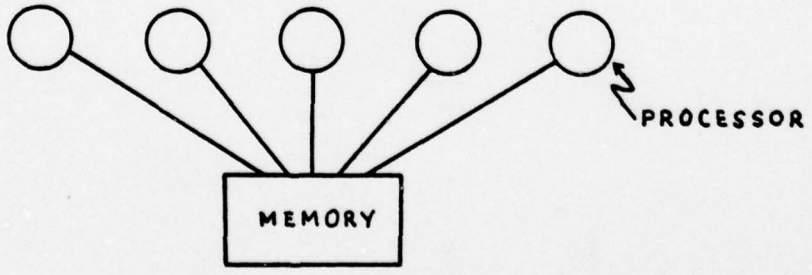


Figure 5.9 - Multiprocessor

8. Multiprocessor

This architecture, in which two or more processors share a common-accessible memory, has not been considered a computer network. The common memory is used as a communication path among the processors as well as a storage medium. Communications can be very fast because they can be performed by pointers to the messages that are interchanged.

The multi-processor architecture has some characteristics of the star and global bus due to centralization of communications in one device, but the nature of memory as an inherent and subordinate part of a computer imposes differences:

a. good cost and place-modularity for processors and memory. It is possible to add processors and to increase memory size; the path structure by which processors access memory has a great influence on cost-modularity; if each processor has a private path to memory, the number of ports in memory limit the number of processors; if memory is accessed through a common bus, cost-modularity is very good, since processors can be easily connected by the bus;

b. poor cost-modularity for memory bandwidth;

c. memory bandwidth is a major bottleneck of the system;

d. very good failure effect and low cost of graceful degradation for processors;

e. poor failure-effect and high cost of graceful degradation for central memory and memory bus;

f. moderate logical complexity-careful design is required nevertheless, to avoid deadlocks and race conditions.

9. Regular Interconnection

In this architecture every processor is connected to an equal number of other processors, providing identical neighbor relationships, or, in terms of a graph, every node has the same total degree. The loop is a special case of this topology. Messages are routed from source to destination, with each intervening processor deciding which of its neighbors should relay a message.

The characteristics of this design, one example of which is shown in Figure 5.10, are:

a. bad modularity and failure characteristics because of the requirement for regularity;

b. logical complexity is moderate-it is increased, nevertheless, if reconfiguration after failure is to a irregular structure;

c. no connection flexibility.

There has been no practical implementation of this structure, due to the characteristics above.

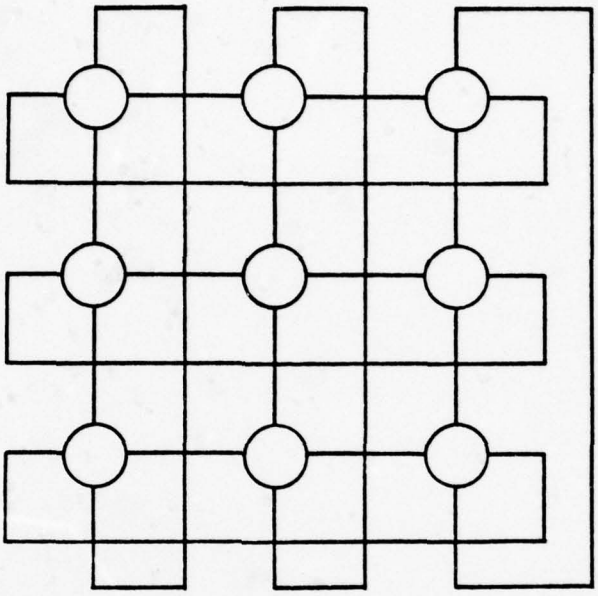


Figure 5.10 - Regular interconnection

10. Irregular Interconnection

This topology differs from the previous one in that there is no requirement for regular neighbor relationships. A processor may be connected to one or more other processors in the network. Most of the implemented distributed computer networks are in this class. Many of the system characteristics vary with the degree of irregularity of interconnection.

General characteristics of this topology, one case of which is seen in Figure 5.11, are:

a. extremely good place-modularity-processors and communication links are added when and where needed;

b. very good cost-modularity-additional processors are architecturally required to have only one path to the rest of the system; other principles of design, nevertheless, may require at least two or three connections: this is not a requirement of the topology but, rather a reliability requirement;

c. very good connection flexibility-any processor may be linked to any other processor in the network;

d. good to very good failure-effect and low to very low cost of graceful degradation-the more irregular the architecture, the less is the effect of a failure (processor or communication link) and the easier are the reconfiguration actions after a fault, due to the existence of multiple paths (node disjoint) between any two processors;

e. high to very high logical complexity-each switch requires knowledge of the overall system status;

f. bottlenecks are not a problem and when saturation takes place in the network, it can be easily alleviated due to the good place-modularity of the architecture.

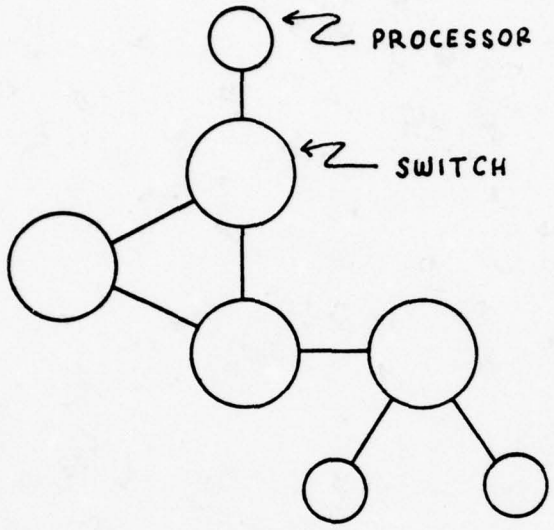


Figure 5.11 - Irregular interconnection

E. Geographic Dispersion of Distributed Topologies

Not all distributed architectures are characterized by geographical dispersion. Communication line cost may limit dispersion.

In this sense, good architectures for distributed networks are irregular interconnection, star, loop types and bus types. Perhaps the dominant topology in large computer networks is the irregular interconnection, which does not impose any pattern for interconnection and is designed according to optimization principles, such as minimum cost, minimum average message delay, maximum throughput, minimum communication cost, etc.; then the cost of high

logical complexity is largely compensated by the minimization of expensive communication lines.

Multiprocessor systems require very large bandwidths for processors-memory communications and thus are generally confined to one room. This is the reason they are not commonly considered to be computer networks.

Virtually all existent geographically dispersed, completely connected networks have small numbers of processors, i.e., less than four processors [Ref. 27]. The communication cost of a network with a large number of processors spread over a large area would be prohibitive and line utilization would be low.

The star architecture can easily be spread over large areas. In many cases the star is the most economical way of implementing a distributed computer network because it intrinsically minimizes the number of communication lines and has low logical complexity.

Loop systems have not been used for dispersion over large areas. Typically they have been implemented with transmission lines to integrate a set of minicomputers Perhaps the main factors preventing greater use of loops is their poor failure-effect in relation to communications links and the delay associated with the transmission of messages when the loop has many nodes. Also, for large distances, the natural solution is to employ leased common carrier circuits; often such circuits are bidirectional and

one direction of transmission would be wasted in a unidirectional loop.

Bus systems are used in conjunction with broadcast radio channels [Ref. 1, 2]. This is an inexpensive way of implementing a network in remote areas; in transcontinental networks, bus architectures used with satellite channels may achieve great economy in communication costs. When the bus is implemented on a point-to-point basis, the system is generally confined to one room because for short distances it is economical to implement a parallel bus. If a serial bus is employed, the system may be spread more by using transmission lines.

VI. NETWORK INTEGRATION

A. INTRODUCTION

A computer network is not just a set of hardware and communication circuits. The transformation of this set of resources into a computer network is performed by a set of rules and conventions which enables the conversation among the network components and an orderly and efficient use of resources [Ref. 9]. Thus, this set of rules and conventions integrates and gives unity to the computer network.

In any communication system and in particular in computer networks such a set of conventions and control procedures is necessary to allow efficient, smooth and correct transfer of information in the system. The main purpose of these rules and conventions are [Ref. 46]:

1. to make the system convenient to use
2. to prevent loss of data
3. to detect message duplications
4. for efficient and orderly use of resources (lines, communication processors, etc.)
5. for error detection and correction
6. to detect system element failures
7. for recovery from system failure
8. to prevent and recover from traffic deadlocks
9. to prevent congestion.

The complexity of these functions varies from low in

centralized networks to very high in irregular distributed networks employing packet switching. Most functions are implemented in software; typically the low level functions such as link control are performed by hardware and the high level, complex function are performed by software.

B. INTERFACE COMPONENTS

An interface is an entity between two components of a system or two systems that serves to connect them. The interface may be a system, component device, or set of specifications /Ref. 10/.

Computer communication interfaces are composed of both devices and specifications. There are many different ways to classify computer communication interfaces. One possible way is to divide the interfaces according to different functional levels: physical, electrical, logical or procedural.

The physical portion of the interface specifies the way in which the two devices are actually connected together mechanically. This includes the number of wires and the dimensions of the physical connectors (generally a male and female connector are specified) in which the devices terminate.

The electrical portion of the interface specifies the voltage levels, frequency, impedance and other electrical attributes of the various leads. The basic capability pro-

vided by adherence to standards at this level is the transfer of bits of data across the interface. These bits may represent characters being transferred or higher level control signals within the interface.

The logical portion of the interface specifies how the bits of data are grouped into fields for the purposes of data transfer and signaling. The use of certain special characters for communications control is specified at this level as, for example, synchronization (SYN) and start-of-header (SOH). The logical specification plays the role of the language by means of which data is exchanged through the interface.

If the logical level of the interface is viewed as specifying the syntax of the data flow across the interface, then the procedural specifications should be viewed as providing the semantics. Specifications at this level determine the legal sequences of communication control characters, or the legal contents of various fields, or the valid commands and responses in controling data flow. The same basic set of control characters or fields may be used in a variety of different ways according to the procedural specifications.

C. PROTOCOLS

The term protocol is generally used to refer to the logical and procedural aspects of an interface. Thus, a

protocol specification includes both syntax and semantics. Semantics also include information about legal use of resources-not only the legal commands, but who can issue them and when.

A complex interface may contain several levels of protocol. An appreciation of this aspect of protocols is quite important in designing and evaluating network interconnections.

Link control disciplines are in most cases capable of being implemented in hardware, although they could be implemented in software, and therefore be considered as a type of process-to-process protocol.

Link control protocols are either bit-oriented or character-oriented. In both cases, there is a way to frame discrete units of information, a set of commands and a way to convey them, and an error detection mechanism.

In computer networks, most protocols are for the exchange of information between processes. In the case of a simple terminal, the actions of the human operator are considered to constitute the process. In fact, when somebody logs on a computer system, a process is created to manage the terminal communications, thus allowing the device to be treated as any other process in the system. In this way, just one type of high-level communication exists in the system-the interprocess communication; the particularities of any device (not just terminals) are disguised by the soft-

ware system which handles the device.

Many different processes may reside in the same physical device. For example, a host computer may have a communications handler process, various operating systems processes, and many user (application level) processes. Messages entering the host through the same physical interface may be intended for any of these processes. In most cases there is a hierarchy of control for delivering messages to the proper recipient. The communications handler must examine all messages; the operating system must examine the messages intended for user processes. Thus, the basic structure for all messages must have some mechanism for identifying level and recipient. Frequently, information for recipients at multiple levels may be conveyed by the same physical message. One example is the method used in most packet switched networks, in which messages are simply nested within messages, according the relative position of the recipient in the hierarchy. Each recipient "peels off" the part intended for him and passes the remainder to the next level recipient.

D. CONTROL REQUIREMENTS

There are a variety of control-related functions that must be performed by all computer networks. These functions include addressing, signaling, flow control and error control.

In a communications system, some means of addressing is required whenever sender and receiver are not directly

connected. With a multi-channel terminal such as a host computer, which also contains different levels of addresses (such as operating system level, user program level), the question of addressing the proper recipient is non-trivial.

Signaling refers to the means by which control information is exchanged inside the network. The two primary classes of signaling techniques are in-band and out-of-band signaling. In the former, control information is exchanged in the same way that data are exchanged, with some special identification marking it as control; in the latter, some other means is provided for exchanging control information, separately from data. Where nested protocols are used, control information sent at a different level of protocol may be considered as out-of-band signaling, even though all the data at all levels of protocol are transported by the same physical means.

Flow control refers to mechanisms for varying the data rate of data exchange between any two points in order to prevent overload. With probabilistic message generation and fixed capacity in network components (such as buffers and communication circuits), overload would be inevitable without such mechanisms to temporarily stop or slow down the rate of message arrivals.

Sequencing and acknowledgements of messages can be considered as part of flow control or can be treated separately. The question of sequencing is generally treated as part of

the service to be rendered, and not as a mechanism that can be implemented in a variety of ways. Message acknowledgements, on the other hand, can be accomplished in a number of ways, and is frequently integrated with flow control procedures.

Just as message exchange can occur between parties at various levels in a hierarchy, so too can flow control be implemented for any of these levels. Of course, the actions of a flow control mechanism at lower levels in the hierarchy (at the communications handler, for example) would also be effective for all higher levels.

Error control mechanisms also are implemented at various levels. While lower levels of an interface may adequately address the question of error detection and possibly even retransmission, somewhat higher levels may have to become involved with the detection of duplicate data occuring due to retransmissions after timeouts and to reinitialization after a catastrophic failure. Message acknowledgement may also be implemented as part of an error control scheme, and in fact, error and flow control can be based on the same mechanism.

E. ROUTING

In tree-like networks the problems of connecting (physically or logically) two points are straight-forward, since there is only one possible path between any two nodes.

Irregular networks, on the other hand, present the greatest difficulty for finding a route between two nodes.

Routing algorithms may be classified as [Ref. 25]:

1. deterministic
    a. flooding
    b. fixed
    c. split traffic
    d. ideal observer
2. stochastic
    a. random
    b. isolated
    c. distributed

Routing algorithms are used in circuit, message, and packet switching networks. For data communications purposes the emphasis is on routing schemes for messages or packet-switching networks. Several of the methods of routing were developed for circuit-switching in the telephone network. They were afterwards extended for message and packet-switching networks.

1. Deterministic Algorithms

Deterministic routing algorithms derive routes according to a rule specified in advance. Each route produces loop-free routing, so that messages can never become trapped in closed paths [Ref. 25, 41].

Flooding is perhaps the simplest of all routing algorithms. According to this algorithm, a node which re-

ceives a message immediately retransmits it over all connected lines except the one from which the message was received. After the message has circulated through the network for a specified period, a message is returned to the node of origin as confirmation that the flooding cycle has been completed for that message.

Flooding always routes messages with minimum delay and does not require large space for storage because it does not maintain current routing information or build delay measuring mechanisms. On the other hand, after a transent period, a network employing flooding will quickly become congested, because of the excess of traffic in the network.

Flooding has been suggested as an initial "path finder" to derive routing and path delay statistics that other techniques require. However, efficiency considerations rule out flooding as a practical policy for network routing.

Fixed routing, another deterministic algorithm, assumes fixed topology and known traffic patterns. It reduces optimal route selection to a multi-commodity flow with well-defined techniques for solution. This algorithm usually obtains appropriate routing from a directory in the memory of the node computer; the directory is fixed for any particular network configuration. A routing directory contains the link address for sending a packet message from

a node to any location in the network. By searching the directory for a given destination, the node obtains a cross-reference to the corresponding link for transmitting the packet.

Because of their inflexibility, fixed routing techniques do not present good graceful degradation. To improve this aspect, modified algorithms that include alternative fixed routes are required.

Split traffic routing, sometimes called traffic befurcation, allows traffic to flow on more than one path between a given source and destination. If two different paths, R_1 and R_2, are available, a packet at node S would be routed over R_1 with probability p and over R_2 with probability (1-p). Similarly, traffic can be split over more than two routes with a different probability for each-the sum of all probabilities being 1. This algorithm uses a directory that lists all the alternative routes, the probability for each route, and a record of past choices that helps establish the current choice. Split traffic, when compared to fixed routing, maintains a better balance of traffic throughout the network, thus achieving smaller average message delays. Nevertheless, in terms of throughput and delay, split traffic always turns out to be somewhat less than optimum.

A fourth deterministic algorithm, the ideal observer routing technique, requires total knowledge of the network

in a continuous fashion. For each new packet entering the network, the node computes a route that minimizes the travel time to the packet's destination. This computation is based upon complete present information about packets that previously entered the network and the routes that were computed for them. Because of inherent network delays, the ideal observer technique is only of theoretical interest. For example, when a packet reaches its destination, completes a message, and leaves the system, the destination node informs the source node of this fact, but the source node doesn't learn about it until after the advisory packet has worked its way back through the network. Thus the observer can't know what the network is like now, only what it was a few moments ago. Also the amount of information that has to circulate in the network to implement this scheme is too great to be of practical use. Nevertheless the concept of "total knowledge" is useful in establishing an upper bound on network performance.

2. Stochastic Algorithms

Stochastic algorithms are probabilistic decision rules, as opposed to deterministic rules. They select routes in accordance with network topology, perhaps combined with estimates of the state of the network. These estimates are based on statistically derived delay information transferred from node to node between packets. Each node maintains a routing table for this delay information, and updates the

table whenever new information becomes available. The algorithms use the information in the table in much the same way that the split-traffic and fixed-routing algorithms use their directories and the number of routes (number of table entries) may be greater than the number of links from the source node [Ref. 12, 25, 41].

Random routing algorithms assume that each node sends its received messages forward along a link chosen at random. The message eventually arrives at the destination after following what is sometimes called a "drunkard's walk". The algorithm can include a bias to guide the message roughly in the right direction, but should retain a substantial random element to cope with possible link or node failures. Although such algorithms are inefficient, they are surprisingly stable in networks having high probability of link or node failure.

Isolated routing, using local delay estimates, assumes that traffic loads are approximately equal in both directions between any given source and destination pair. This technique, also called "backward learning", uses as the estimate a weighted version of the time delay incurred by messages going in the reverse direction from the destination node, considered as a source, to the node in question. The algorithm updates the previous delay estimate for transmitting a message from the current node to another node through a specific link when a message from that node arrives

through this link carrying the time it has traveled across the network. A simple linear recursive equation is used for delay estimate updating. This technique was found in practice to suffer from a "ping-pong" or looping effect, in which messages sometimes return to a node from which they were previously transmitted. In addition backward learning was found to adpat poorly to damaged networks.

Another isolated routing technique, called the isolated shortest queue procedure, stems from the development of an adaptive routing method for military voice communications that would survive well in the event of nodes and links that have high failure probabilities (war zone). The procedure, sometimes called the "hot potato" method, requires intermediate nodes to retransmit a packet as quickly as possible after receiving it. Each node in the network, consulting a ranked list of lines leading to neighboring nodes for every destination, directs packets to the highest rankding free line for a given destination, or, if none are available, to the line having the shortest queue. Thus, the nodes handle the packets (messages) like hot potatoes, getting rid of them as soon as possible.

Distributed algorithms rely on exchange of observed delay information between nodes. This approach introduces an inordinate amount of measurement information into the network and is therefore impractical for large networks. One modified procedure uses a "minimum delay vector"; another

has an "area approach".

A minimum delay vector for a particular node is the delay from that node to each of its adjacent nodes - rather than to all the other nodes, as in the unmodified algorithm. Each node exchanges this vector with each of its adjacent nodes; they update it and pass it on to their neighbors. As these vectors pass along the nodes, they eventually provide each node with a matrix of delays to all possible destinations. Exchanges and updates can be repeated periodically or irregularly.

The area approach partitions the network into disjoint areas. Within each area, every node exchanges information with every other node, but it exchanges similar information with adjacent areas as though each were a single node. This approach can be extended to a hierarchy of routing clusters at many levels. The objective of the area approach is to reduce the amount of routing information that each node must retain.

An algorithm that combines stochastic and topological features uses a network routing center to which each node periodically sends updated traffic information. With this information, the network routing center regenerates routing tables, which remain fixed until the next update. This technique has two disadvantages: like other centralized networks, it has a single switching point (the NRC), and single paths for each source-destination pair constrain system behavior.

A hybrid algorithm with respect to centralization of routing divides routing decisions into two categories: those that affect the network only locally, implemented at the node level, and those with global effect, entrusted to the network routing center. This algorithm, called delta routing, appears to take advantage of the favorable aspects of both centralization and distribution of the routing function, but still has the weakness inherent in a central control facility.

3. Routing Performance Measures

The performance of a routing algorithm must be considered in terms of four factors of fundamental importance for the network as a whole [Ref. 12, 25, 31].

The first factor is delay. The theoretical minimum for delay is determined by the network topology and the traffic levels of the moment. The routing program should come as close as possible to that minimum, particularly with respect to interactive traffic.

Throughput is another factor to be evaluated. Throughput considerations parallel that of delay in the sense that there is a maximum throughput level for a given network topology and traffic mix. There is a tradeoff between delay and throughput. High throughput rates are most important for bulk traffic.

Cost is the third factor. The routing algorithm can affect the cost of network utilization by its demands for three resources: line bandwidth, node bandwidth, and node

storage. Routing over as few lines and nodes as possible and choosing under-utilized lines and nodes can keep bandwidth demands down; and keeping network queues short can reduce storage requirements.

F. FLOW CONTROL ALGORITHMS

Congestion can occur within a nework employing packet- or message-switching either globally or locally. It can be relieved by the use of a hierarchy of protocols that indicate which of several alternative actions is appropriate, on the basis of information from message and packet control fields. The hierarchy consists of host-to-host, source-to-destination, and node-to-node protocols, corresponding respectively to action at the message, packet, and link levels of a packet-switching network /Ref. 25/.

Node-to-node or link protocols are local; host-to-host and source-to-destination are global, or end-to-end, since they control traffic into the network. Three flow control schemes among many utilize one or more of these protocols: isarithmic, buffer storage allocation, and special route assignment.

In an isarithmic network, the total number of packets is held constant by replacing outgoing data-carrying packets with dummy packets. A dummy packet has a special identification block, is part of a one packet message, is always addressed by any node to its adjacent node in the path in which the dummy is inserted, and contains a

meaningless text block. New packets enter the network in place of the dummy packets.

With buffer storage allocation the source node requests allocation of message reassembly space in the destination node before transmitting the message. The alternative, to go ahead and transmit the message, occasionally would find the destination node's receiving buffer full when the traffic was heavy, in which case the packets would not clear the last link in the path from source to destination and would pile up in the last node short of the destination. This in turn could fill up the buffers of all the nodes adjacent to the destination, creating a wall around the destination. If the destination node were a source of packets moving in the opposite direction, it would not clear its transmitting buffers. Soon the whole network would lock up.

Another alternative to buffer storage allocation is to have the destination notify the source when it cannot accept a packet after the packet has arrived at the destination. This method requires the source to transmit these packets at least twice. Advanced allocation of reassembly buffers, resulting in occasional transmission delays, is more efficient than recovering from discarded packets. However, neither method is considered an adequate procedure for real-time or data-sharing users.

Still another alternative to buffer storage allocation is the method of assigning special routes, based on status in-

formation received from adjacent nodes, and on traffic patterns encountered by the node over the last few seconds. Such a special route should not be used, however, in response to a rapid but short-lived increase in traffic flow. Therefore, measurements of the rate of change of traffic along each path are combined with a predefined interval of time, and alternative routing is established only if the traffic change persists. The algorithm has a number of rules:

a. The routing selection is performed independently by each node, on the basis of information received from adjacent nodes and of traffic patterns;

b. The algorithm guarantees that individual routing decisions possess global continuity for the network;

c. Routing tables are always updated synchronously, with additional asynchronous updating when necessary;

d. In selecting routes, the network is decomposed into a union of identical and overlapping subnetworks with separate routing computed for each subnetwork;

e. For an unloaded net, a path is made which crosses the fewest nodes on the way to the destination;

f. For a loaded net, traffic is diverted from fully occupied links whenever possible;

g. Changes in routing occur only when a new traffic pattern is sustained for a minimum time;

h. Additional paths can be established for a given

destination to allow individual packets to depart on separate links;

i. Traffic flow on any link in a subnetwork can occur in only one direction at a time (half-duplex operation);

j. Direction of flow in a link may change, but only after the link remains unused for a short interval of time;

k. The maximum allowed traffic through each node in a subnetwork is regulated so as to change slowly; this provides stability and allows adjustments for increased traffic flow;

l. For each subnetwork, loops in routing are quickly detected and broken.

G. THE CCITT STANDARD HOST INTERFACE-X.25

CCITT recommendation X.25 is an international standard for the "interface between data terminal equipment and data circuit-termination for terminals operating in the packet node on public data networks". In other words, this interface is intended for use by host computers and other types of customer terminals which are intelligent enough to format data in the way specified by the interface standard and to respond in the proper ways to the complex control signaling requirements of the interface. The interface is designed for multi-channel use; that is, it provides for communication with multiple independent user processes within a single user terminal (computer).

The X.25 recommendation is structured into three independent parts as follows:

Level 1 - the physical, electrical, functional, and procedural characteristics to establish, maintain and disconnect the physical link between the data terminal equipment (computer) and the data communications equipment;

Level 2 - the link control level for the interchange of data between the data terminal equipment and the network;

Level 3 - the communication control level defines the formatting of information and control procedures used between the data terminal equipment and the network for the purpose of transferring user data through the network and for establishment of end-to-end connections.

Figure 6.1 shows schematically the typical connection of data terminal equipment to a network and the physical point of interface where the standard is applicable.

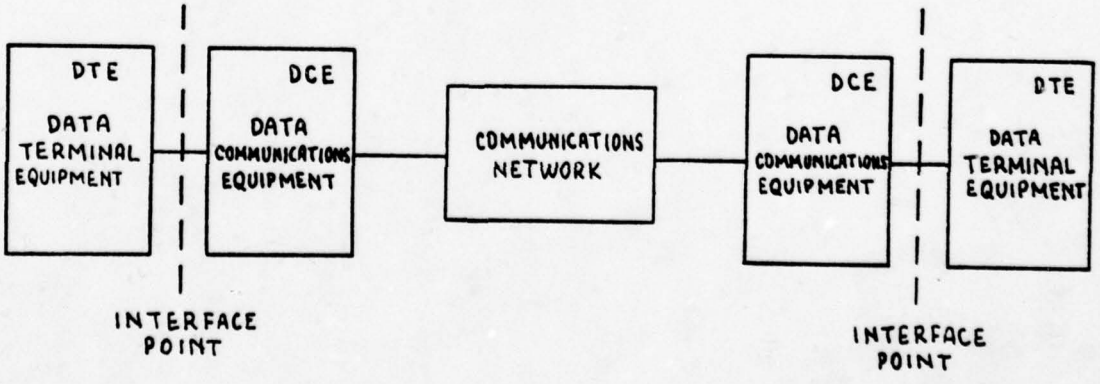


Figure 6.1 - Typical Interconnection

Figure 6.2 displays the logical relation of the hierarchical levels to the network components.

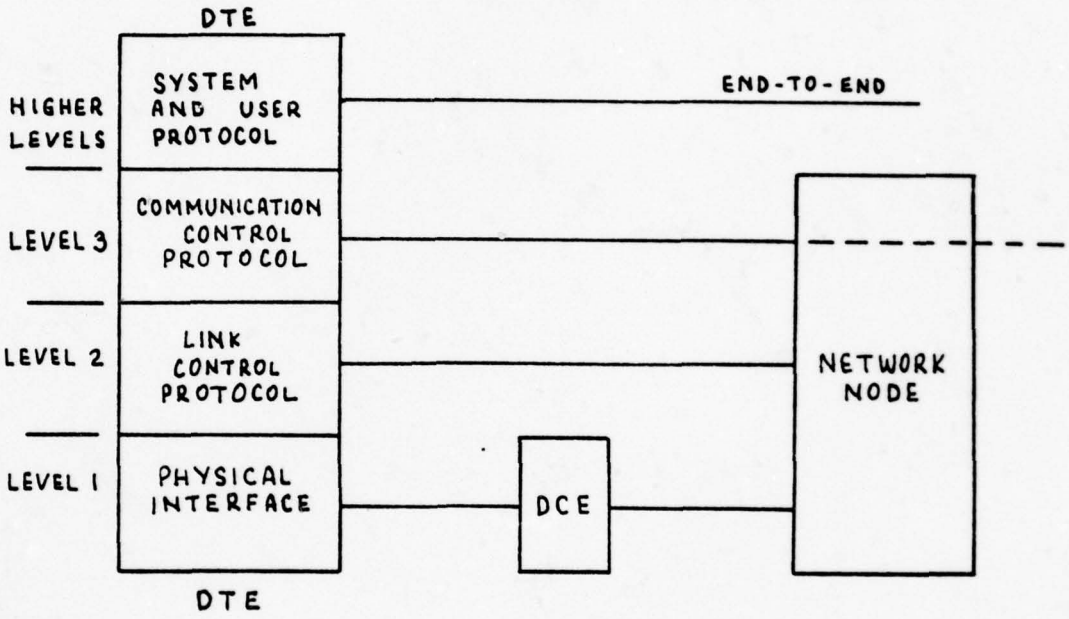


Figure 6.2 - Hierarchy of interface levels

For level 1, the X.25 recommendation specifies the use of CCITT recommendation X.21, a general-purpose interface for synchronous operation in public data networks. For an interim period, the use of recommendation X.21-bis (essentially the same as EIA RS-232) is approved.

For level 2, a link access procedure is defined. It uses the principles and terminology of the High Level Data Link Control Procedure (HDLC) specified by the International

Standards Organization (ISO). This is a bit-oriented line discipline suitable for use in a variety of environments.

Level 2 provides the function of error and flow control of the access link between the DTE and the network. Each frame has a check sequence to detect errors. Error frames are retransmitted when requested by the receiving end. Flow control is accomplished through sending of Receiver Ready (RR) commands or Receiver Not Ready (RNR) commands (buffer storage allocation).

Level 3 of X.25 defines the packet formats and control procedures for exchange of information between a DTE and network. The type of service supported by the current X.25 is the so-called virtual circuit service, which is a public service for virtual circuit switching. Establishment of a virtual circuit is initiated from a calling DTE by a Call Request packet. The called DTE is notified that a "circuit" is being established by an Incoming Call packet. Subsequently, the calling DTE is notified that the circuit is established by a Call Connected packet. Data packets are then exchanged between DTE's. Upon completion of the call, the circuit can be disestablished by either a DTE Clear Request packet or DCE Clear Indication packet, as appropriate, followed by a Clear Confirmation packet response.

The capability for multiplexing up to 4096 logical channels (virtual circuits) on a single access link is also provided by X.25. Each logical channel can be used for

virtual calls or a permanent virtual circuit. Each packet exchanged across the interface has it associated logical channel number identified. Each logical channel operates independently of others. The data packets are also each identified by a sequence number, which is used for flow control on individual logical channels. RNR packets are used to stop transmission on a channel while RR packets permit transmission. The sequence numbering scheme may be based upon either modulo 8 for normal operation or modulo 128 for extended transmission delay conditions. The sequence numbers are then recycled every 8 or 128 packets, as appropriate. Packet formats for the different types of packets are specified. Data packets are limited to a maximum data field length. All networks must allow a maximum of 128 octets (8 bit byte), while some networks may also support maximum lengths of 16, 32, 64, 256, 516 and 1024 octets or 255 octets on an exception basis. Thus, no packet assembly/disassembly capability is assumed by the interface (DCE). Networks supporting the X.25 protocol are free to disassemble the octet data fields into still smaller packets for internal switching, if this is required.

PART 2

DESIGN OF A NETWORK FOR THE NPS TIME SHARING SYSTEM

VII. PROBLEM DEFINITION

A. THE PRESENT NPS CENTRAL COMPUTER FACILITY

The W. R. Church Computer Center, located in the first floor of Ingersoll Hall, is the organizational unit of the NPS which has the responsbility of providing campus-wide computer services. Its services are available to all faculty, staff, and students of the school for use in connection with instruction, research, or administrative activities. The center offers two modes of operational service, i.e., batch-processing and general purpose multi-access time-sharing.

The present equipment, based on an IBM/360, includes two model 67 processing units; four different levels of storage, including 2 M bytes of core, 4 M bytes on a drum, 24 disk drives with 29 M bytes each and 8 disk drives with 100 M bytes each and 9 magnetic tape units; two high-speed plotters, fifty remote hard-copy and video terminals, and an IBM 2550 Graphical Display Unit. The two processors are identical and, by means of a configuration control unit, can access directly, or control, all components of the system including core storage modules, input/output controllers and devices.

The allocation of resources of the system to each CPU, using the configuration control unit, is static and mutual-exclusive, that is, one unit can only be used by just one CPU at a time; thus, the physical resources of the system are divided between the two CPU's.

Typically, during working hours, the computer center operates with two independent systems - one CPU under the OS/360-MVT operating system, runs the batch processing and the other one, under CP-67 with CMS (Cambridge Monitor System) runs multi-access time-sharing. The batch environment is a punched card oriented one in which jobs are typically submitted to the system by means of a deck of cards. There is little, if any, coupling between the two systems (batch and time-sharing); there is no way of automatically transfering one application program from one environment to the other; for example, if one edits and tests a program in the time-sharing system and wants to run it in the batch system, one has to first have the program punched and then has to submit the card deck to the batch processor.

With the exception of remote terminals and modems, all the hardware is concentrated in Ingersoll Hall. Thus, the input card decks have to be hand carried across the campus in order to submit a job in the batch mode; also, printed outputs have to be picked up in the computer center, even printer outputs of the time-sharing system.

The center possesses a wide variety of software resources. In the batch environment, under OS/360-MVT, the following language processors are currently available:

a. FORTRAN IV G;

b. FORTRAN IV H;

c. WATFIV;

d. PL/1;

e. PL/C;

f. Assembler 360;

g. ANS COBOL;

h. GPSSV;

i. SIMSCRIPT;

j. ALGOL W;

k. WATBOL;

l. PLM;

m. FORMAC;

n. BASIC.

In addition to the software facilities which are part of OS/360-MVT and those routines embedded in each language processor, the center maintains a large library of programs for public use. The library is composed of programs from many origins: IBM "Scientific Subroutine Packages", computer center-supplied programs, faculty and student-supplied routines, International Mathematical and Statistical Library and others. Some large applications packages which, in effect, provide special language capabilities are available in the library:

a. Mathematical Program System (MPS);

b. Biomedical Statistics Packages;

c. Electronic Circuit Analysis Program (ECAP);

d. Digital Simulation Language (DSL);

e. Linear Systems Analysis (LISA);

f. Matrix Language (MATLAL);

g. SORT/MERGE (IBM);

h. Continuous System Modeling Program (CSMPIII);

i. Statistical Package for the Social Sciences (SPSS);

j. SNAP/IEDA, Introduction to Explanatory Data Analysis;

k. Eigensystem Package (EISPACK);

l. Linear Systems Package (LINSYS);

m. Function Package (FUNPACK).

The public library stores programs in three forms:

a. Source programs-those in high level languages;

b. Object programs-those compiled but not directly executable programs;

c. Load programs-those which can be directly executable.

In addition to the public library, the center supports private libraries, which are programs with access limited to one or few users and which are kept on-line for future use by their owners.

The time-sharing environment allows the use of the following languages:

a. FORTRAN G;

b. Assembler 360;

c. BASIC;

d. ALGOL W;

e. PLM;

f. APL.

The public library of CP/CMS is divided in three parts:

a. System library (SYSLIB), which is the set of software facilities embedded in CP/CMS;

b. SSPLIB, which comprises FORTRAN scientific routines in object language form;

c. IMSL which is also in object language form.

The APL interpreter does have an extensive library; also the APL processor is able to put in the user's terminal explanatory notes about the use of its library programs.

Users of the time-sharing systems are classified into two categories: general users and private users. General users share common disk file storage; the files of general users are stored in the space allocated to the terminal where they were entered; a general user can only access files in the terminal where they were created.

Private users have private disk space of fixed size, which is normally 256 K bytes. The space of a private user cannot be accessed by other users, unless they know his identification and password.

The time-sharing system has some Tektronix 4012 Display Terminals which have the capability for interactive graphical work. There is no capability of off-line plotting, since the plotters are used with the batch system.

B. THE NEED FOR A NEW SYSTEM

One of the first steps in planning the substitution of a system should be to try to investigate the reasons for the

change. This has two objects; the first is to make sure that the change is really necessary; the second is to learn from past experience and, consequently, direct the new design in the right way, incorporating the missing and needed features and avoiding the flaws of the old system.

The Future Computer Planning Committee at the NPS has conducted two surveys, regarding user requirements. Based on the analysis of those surveys and another one performed by a student, the justification for a new computer system is summarized below. From these findings some of the goals and requirements for the new system can be readily derived.

1. Reliability

The present system has been operating for more than ten years. Its failure rate in recent years has been unacceptable.

2. Needfor a Powerful Batch Computer

The school has responsibilities for conducting advanced education and research and requires powerful computers to support these goals. Science and technology have grown to a level of sophistication which demands enormous computational power. Even today's super computers are still inadequate to deal with current and future computational demands. For example, going from a two dimensional aerodynamic simulation-the current state-of-the-art capability-to a more realistic three dimensional formulation increases the computational demands by a factor of one hundred.

Doubling the resolution of today's climate models, already taxing the current super computers, increases the computational demands by a factor of ten. Many other examples can be found in research areas of interest for the school such as control engineering, operation research, signal processing, physics, oceanography, etc. The school's batch system is very modest when compared to the capabilities of super computers. Thus, the school is already behind in computational capability required for today's research needs.

The analyses that were conducted have shown that the batch system performance was adequate for most of the educational needs, but it lacks capability to handle many of the school's research projects. Students submit the greatest number of jobs. The percentage of CPU hours used by students is much lower than their percentage of jobs. In contrast, the faculty has a much higher percentage of CPU time than its fraction of jobs. It was found that faculty research projects are large and tend to be CPU bound [Ref. 32].

The study of aggregate data concluded that more than half of the jobs are executed in less than 10 seconds and that 80% of the jobs use less than 150 k bytes of memory [Ref. 32]. Clearly, these numbers are heavily influenced by student jobs. In contrast, an entirely different situation is indicated with regard to research computational requirements.

The following conclusions were derived in the study of the data provided by the principal batch processing users (researchers) /Ref. 39/:

a. the school needs a computer at least 10 times faster in processor and memory speed than the IBM 360/67;

b. a computer with at least 4 times the present batch system memory capacity will be required;

c. other capabilities which are required are graphics terminals and a good data base system;

d. research progress is inhibited, grant opportunities are threatened and NPS research is becoming less competitive due to inadequate computing services.

3. Inadequacy of the Present Time-Sharing System

The survey that was conducted among major time-sharing users did not enable the derivation of a functional specification, as the survey of batch users did, because of the inconsistency of the data collected /Ref. 40/. Nevertheless, important findings and conclusions were derived. Also, the previously mentioned analysis conducted by a student presents some facts and comments extracted from interviews of representative users and academic officials /Ref. 32/.

The aggregate of those two studies regarding the time-sharing system portrays the present situation:

a. Since 1975 the usage of the time-sharing system, measured in average CPU hours used, has increased steadily;

b. Presently, response time under CP/CMS is considered inadequate when more than a few terminals are in use (hours of intense use);

c. There is now a great demand for terminals, even early in the quarter; students are standing in line to use terminals; the demand on weekends is almost as high;

d. Sometimes it is not possible to use a terminal because of the inability of making a connection via phone company lines (the way almost all remote terminals are connected to the system);

e. The use of APL is intense;

f. Professors are generating a greatly increasing time sharing load due to course work;

g. There is a need for communication between the batch and the time-sharing system;

h. There is a need for faster terminals;

i. Electronics Engineering, Mathematics, Operations Research and Aeronautics all request availability of remote plotters so that graphics routines can be used in an interactive mode.

On the whole, these facts show the inadequacy of the present time-sharing system. In particular, a), f) and i) suggest that the usage of time-sharing will increase if the facilities are provided. But, b), c) and d) show that the present system is already saturated. From b) and d) it can be concluded that the addition of more terminals would degrade

even more the performance of the system and, thus, is not recommended.

In contrast with the batch system, in time sharing the major users do not represent a significant portion of the system load. The major time sharing users account for /Ref. 40/:

a. 7% of the total number of users;

b. 11% of the total number of sessions;

c. 16% of the total terminal time;

d. the average time per session of the major users is 1.4 times that of all users;

e. only 37% of the major users polled have more than 256 k bytes of private disk space.

These figures are of great importance in view of the planning for the new time sharing system. They suggest that the majority of time sharing usage is dedicated to educational purposes versus research.

C. SCOPE OF THE DESIGN

The design proposed by the Future Computer Planning Committee is toward a system able to work in two modes of operation: batch and time-sharing. Each mode presents different requirements. The functional specifications for the batch computer have already been derived.

In this analysis the existence of a batch system is assumed. The design of this batch system is beyond the scope of this study.

The objectives of this work are:

a. Development of a distributed system approach for the future time sharing system of the NPS;

b. Presentation and analysis of some distributed architectures which can satisfy time sharing requirements;

c. Integration of batch and time sharing into a unified system.

d. Presentation of a data communication structure for support of time sharing and remote batch processing.

VIII. DESIGN BACKGROUND

A. MAN-MACHINE INTERFACE

One of the major concerns of the design or selection of an interactive computer system should be the interface to the users, that is, the appearance of the system to the user. This aspect of the operating system of a time sharing facility will determine its usefulness. Good principles of design in this area, also called "human engineering", should be also present in the software facilities of general use such as library programs.

There are two properties of an interactive system which are essential to the usefulness of the system but which compete with each other to some extent. These properties are: a) effectiveness and b) simplicity of use. By effectiveness it is meant the capability to solve the required problems in a reasonable time. To provide this power there must be generality and flexibility. The simplicity of an interactive system is inversely dependent on the quantity and complexity of actions required by the user [Ref. 43].

It is easy to see how a requirement of simplicity competes with a desire for effectiveness, since as more generality and flexibility are put in a system, more choices (actions) must be made and more complex information must be given by the user to the system to tailor its power to the particular problem at hand. The problem of resolving the conflict between generality and flexibility versus simplicity

of use is one of the significant factors in the selection or design of an interactive system.

One way of resolving conflicting objectives of simplicity of use, generality and flexibility at the expense of increased software, is through the concept of levels of invisibility /Ref. 1/. Operating systems designed under this concept present to the user a linear structure of software modules. This structure is often represented as a ring structure consisting of successive levels or layers as in Figure 8.1. Each successive layer conceals details meaningful only to lower levels. Thus, modules of inner levels are more general and flexible but less simple to use. At the highest level the system has most of the commands needed by the common user in pre-defined modules which are very simple to use. A skilled user, on the other hand, may shape his own high-level utilities, using the lower level modules.

There are several general design principles that should be embodied in a user oriented interactive system. Some of them are /Ref. 43/:

1. self-explanatory;
2. self-help (user assistance);
3. simple interface with user;
4. interaction by anticipation;
5. optional verbosity.

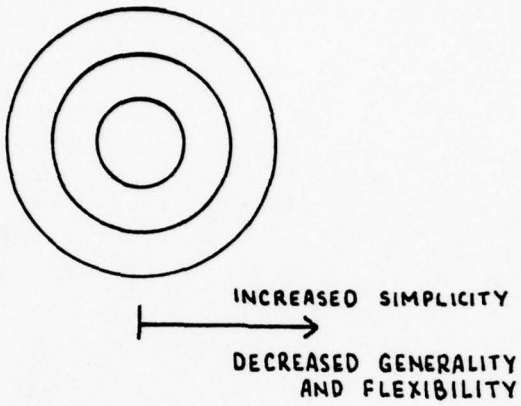


Figure 8.1 - Levels of abstraction

A self-explanatory system is one which displays to the user sufficient explanatory information about the process being carried out to enable him to carry on without reference to some external source of explanation (the system's manual, for example). This can be accomplished by displaying certain basic items and allowing the user to ask for optional additional tutorial material.

A self-help system provides checking of user inputs to the system and provides reminders or instructional displays on user request. Input items are checked for validity or reasonableness, and if appropriate, a diagnostic message is sent to the user and the system awaits his further instructions.

Simple interfacing with the user is accomplished by ensuring that the actions required of him are short, simple

and obvious. One aspect of the interface is the grouping of options presented. For example, if many options are available, it is probably easier for an on-line user to make his choice if the options can be presented in subgroups of five to seven items.

A desirable feature of any on-line system is interaction by anticipation. By this it is meant that all possible desires of a user are anticipated (hopefully) and choices are presented which include all those possibilities. This method allows the user to select a desired option rather than specity that option by entering a correctly spelled command. With a teletype, typewriter or CRT terminal the options could be numbered sequentially as they are printed, and the user would select by simply entering the number corresponding to his choice. The problem of diagnostic messages is alleviated by employing interaction by anticipation.

An interactive system featuring optional verbosity can be considered as a system with two levels of detail in the interface with the user. For the novice or first-time user, the interface will contain detailed explanations to insure that the user understands what he is expected to do and what the computer is doing. On the other hand, an experienced user might choose the mode of operation with few or no explanations and abbreviated communications (concise messages) both to and from the computer. This optional verbosity is especially desirable with teletypes or typewriter

terminals which have slow printing rates, making lengthy messages time consuming and boring the experienced user. With a CRT terminal with fast display the verbosity option is not essential in messages coming from the computer but it is still desirable in the user to computer direction.

Another important factor to be considered is the distinction between reversible and nonreversible requests. A reversible request is one which can easily (without significant computer utilization) be reversed. A nonreversible request cannot be reversed easily (uses a significant amount of computer time) or even cannot be reversed at all (for example, inputs are lost or modified in the process). Reversible requests should be carried out immediately by the computer. Nonreversible requests, on the other side, should require confirmation by the user; the system should send a message calling the attention of the user by echoing the command or explaining the actions to be performed upon execution of the command; after confirmation by the user the required action will take place. This procedure can avoid common disasters such as erasure of the wrong file, unwanted modification of a data base, etc.

B. CHARACTERIZATION OF THE TIME SHARING AT NPS

1. Characteristics of the Work

The main purpose of the time sharing system at NPS is to provide computational power accessible to a multitude

of users, students, faculty and staff, in an interactive way. Thus, users go to the terminal looking for a problem solver, a number cruncher or a clerical helper. Each user submits a specific and different job. Transactions cannot be defined or characterized as opposed to a business information system where they are standardized. Viewed this way the system can be characterized as a computational system.

There is little interaction among users and consequently little terminal to terminal communication. But this capability should not be ruled out in the new system because group projects may require communication among the group components for coordination; this communication may be performed either directly on a terminal to terminal basis or by means of a "mail box" where users drop messages which are stored until they can be delivered when their addressed users log on the system.

The majority of communication is in both directions between user and processor. Here delays must be minimized since they incur degeneration of the system response time.

Typical usage of the time sharing includes:

a. program edit;

b. text edit (word processing);

c. debugging of programs;

c. execution of application programs from the system library or implemented by the user.

2. Characteristics of the Software Resources

The software resources of the time sharing system are the operating system and the data bases. The operating system manages the use of the system resources and interfaces the users to the system. The operating system incorporates a set of software utilities callable through a command language.

Typical software utilities of the operating system include:

a. commands for execution of specific tasks;

b. commands for coordination of tasks;

c. language processors;

d. line editor;

e. text editor;

f. debugger.

The data bases incorporate application software and data. They can be divided into:

a. System library-the set of software utilities not included in the operating system and intended for public use on a read-only basis; the utilities consist typically of programs and routines in high level or machine language which can be incorporated into the user's application software, extensive software packages similar to language processors for specific applications and software helps and files of data for specific use;

b. User library-collection of application programs and files of data kept on line by the users for future use; typically they have restricted access and are not intended for public use.

3. Analysis of Data

The computer center compiles statistics of the time sharing system every month. A set of these data covering the period from February of 1977 to January of 1978 was studied. The data was furnished in aggregated form; a precise definition of the terms used, the calculation process of the parameters and the way the measures are performed were not available. So, reasonable assumptions were made regarding these missing definitions. Time lost due to system failures is not included in the statistics; the assumption is that its influence is small.

The analysis which follows is useful for qualitative characterization of the problem; for the reasons given above an extrapolation of parameters as a basis for designing the new system would be dangerous. A complete analysis, using the raw data gathered by the CP/CMS usage monitor, is recommended for this purpose.

a. Volume of Use

Table 8.1 displays three measures of the volume of use of the system:

1. Number of sessions-total number of sessions performed during one month at all terminals.

2. Number of users-total numer of different users which have used the system during one month;

3. Terminal time-total time that all terminals in the system were active in one month.

The data are displayed in absolute value and in the form of percentage increments relative to the month of February 1977.

	Number of Sessions		Number of users		Terminal time	
Month	ABSLT	INCRMNT (%)	ABSLT	INCRMNT (%)	ABSLT (HOURS)	INCRMNT (%)
FEB/77	3377	0	225	0	2174	0
MAR/77	3168	-6.2	230	2.2	2315	6.5
APR/77	4328	28.2	256	13.8	2833	30.3
MAY/77	4461	32.1	239	6.2	2828	30.1
JUL/77	2879	-14.7	189	-16.0	2001	-8.0
AUG/77	3514	4.1	229	1.8	2211	1.7
OCT/77	3299	-2.3	248	10.2	1852	-14.8
NOV/77	5727	69.6	309	37.3	3858	77.5
DEC/77	3314	-1.8	255	13.3	2346	7.9
JAN/78	3828	13.4	286	27.1	2162	-0.6

Table 8.1 - Volume of time sharing use

Data referring to June 1977 were missing and that of September 1977 were inconsistent and discarded.

An examination of the table shows:

1. By every measure, the volume of use increased during the period examined, while not steadily, but in ascend-

ing cycles; in fact, the application of linear regression analysis to the three measure discloses:

- an average growth in the number of sessions of 55.9 sessions per month
- an average growth in the number of active users of 6.4 users per month
- an average growth in the terminal time of 15.2 hours per month;

2. The cycles match approximately the quarters of the academic year with the greatest use occuring in the second month of every quarter.

b. Terminal Usage

Table 8.2 displays several data relative to terminal usage:

1. Sessions per hour-average of the number of sessions derived by dividing the total number of sessions by the total number of operational hours of the system in one month; it is presented in absolute value and incremental form relative to February of 1977;

2. Session time-average duration of one terminal session in the overall system;

3. Interarrival time-average time between two successive arrivals to one terminal;

4. Terminal utilization factor-average fraction of terminals in use or average terminal capacity that was utilized; it was calculated in two ways:

- method 1 - the session time was divided by the interarrival time;
- method 2 - the total time that all terminals in the system were active was divided by the total number of operational hours of the system times the number of terminals.

MONTH	SESSIONS PER HOUR ABSLT	SESSIONS PER HOUR INCRMNT (%)	SESSION TIME (MIN)	INTERARRIVAL TIME (MIN)	TERMINAL UTILIZATION FACTOR METHOD 1	TERMINAL UTILIZATION FACTOR METHOD 2
FEB/77	15.0	0	38.6	122	0.32	0.19
MAR/77	11.0	-26.7	43.8	167	0.26	0.16
APR/77	16.5	10.0	39.3	112	0.35	0.22
MAY/77	16.3	8.7	38.0	114	0.33	0.21
JUL/77	12.2	-18.7	41.7	153	0.27	0.17
AUG/77	11.6	-22.7	37.8	159	0.24	0.12
OCT/77	11.2	-25.3	33.7	166	0.20	0.13
NOV/77	16.9	12.7	40.4	107	0.38	0.23
DEC/77	12.9	-14.0	42.5	146	0.29	0.18
JAN/78	13.5	-10.0	33.9	137	0.25	0.15

Table 8.2 - Terminal use

Analysis of Table 8.2 discloses:

1. The average number of sessions per hour decreased slightly during the period; the trend was a decrease of 0.09 sessions per hour per month by a linear regression computation; considering that the number of sessions in-

creased in the same period, this shows that the adoption of extended hours of operation by the computer center overcompensated for the growth in usage;

2. The monthly average session duration is an almost constant characteristic of the time sharing usage at NPS; indeed, the figures in the column "session time" have an average of 39 and a standard deviation of 3.4, which is small compared to the mean; it is worth noting that these numbers do not give information about the distribution of session times; beyond this monthly mean nothing is known, because the data were aggregated; an average of forty minutes seems brief for the duration of a typical student session; other users may account for enough short sessions to lower the mean; caution should be exercised in using these figures.

3. If the column of absolute values of number of sessions per hous is ordered according to the increasing figures, the corresponding interarrival times will be perfectly ordered in decreasing values; this shows a good degree of consistency between these columns of data; nevertheless, for a system which has user terminal queues, the interarrival times displayed are large compared to the session times; possible explanations for this fact are given below;

4. Utilization factors of terminals calculated by mode method 2 are consistently lower than those calculated

mode method 2, thus making the assumption of small influence of system failures suspect; an important fact is that both methods yield figures which are low in relation to queue sizes observed at terminals; the above may be explained by one or more of the following:

- there are many underutilized terminals (this is readily noted in the computer center data), which may be due to:
  - low availability (combination of failure rate and repair time)
  - many private terminals which have low usage
  - 'hidden' public terminals;
- there is little 'diffusion' of users, i.e., they usually do not look for vacant terminals in other than the site near their work place; if terminals are not evenly distributed around the campus some terminals may be subject to a high demand because they serve a more active community;
- the distribution of arrival times and service times (average session time) have particularities which lead to congestion; for example, the demand for terminals may be concentrated in a small fraction of the system operational time or the interarrival times happen to be short in a period of large service times.

c. CPU Utilization

Table 8.3 condenses the following figures relative to CPU utilization in the CP/CMS system at the school:

1. CPU time-the sum of virtual CPU time utilized by all users during one month; it is presented in absolute value and incremental form relative to February of 1977;

2. CPU time per session-the average virtual CPU time spent in one terminal session;

3. CPU time per user-the average virtual CPU time utilized by each user during one month;

4. CPU utilization factor-average fraction of the operational time that the CPU served the users or average CPU capacity that was utilized; it is presented in two ways:

- basic-the total virtual CPU time (simply CPU time above) was divided by the total system operational time, both during one month
- adjusted-an arbitrary overhead factor equal to 1.25 (realistic) was applied to the CPU time and another arbitrary reliability factor of 0.9 was applied to the operational time to account for failures during operational hours; then the calculation above was performed;

5. Terminal time per CPU time-the total time that terminals were active was divided by the CPU time; this

is the number of seconds of terminal session which corresponded to one second of CPU time in the particular month;

6. Active terminals-expected value of active terminals obtained by multiplying the total number of terminals by the terminal utilization factor (method 1).

MONTH	CPU ABSLT (MIN)	TIME INCRMNT (%)	CPU TIME PER SESSION (SEC)	CPU TIME PER USER (SEC)	CPU UTILIZATION FACTOR BASIC	CPU UTILIZATION FACTOR ADJUSTED	TERMINAL TIME PER CPU TIME	ACTIVE TERMINALS
FEB/77	4058	0	72	1082	0.30	0.42	32.1	16.0
MAR/77	6759	66.6	128	1763	0.39	0.54	20.6	13.0
APR/77	5697	40.4	78	1335	0.36	0.50	29.8	17.5
MAY/77	5451	34.3	73	1368	0.33	0.46	31.1	16.5
JUL/77	3333	-17.9	69	1086	0.24	0.33	36.0	13.5
AUG/77	3389	-16.5	57	888	0.19	0.26	39.1	12.0
OCT/77	3304	-18.6	60	799	0.19	0.26	33.6	10.0
NOV/77	5211	28.4	54	1011	0.26	0.36	44.4	19.0
DEC/77	5564	37.1	100	1309	0.36	0.50	25.3	14.5
JAN/78	2381	-41.3	37	497	0.14	0.19	54.5	12.5

Table 8.3 - CPU utilization

Analysis of Table 8.3 follows:

1. The total CPU time demanded per month varied considerably during the period and showed a tendency to decrease; by a linear regression calculation this decrease is

in average 195.6 minutes per month; this is in contrast with the volume of time sharing use which increased; the explanation may be that the increase in volume was due to the entry of novice users demanding little computing power;

2. The CPU time per session and the CPU time per user during each month are modest, thus showing that the time sharing system is used for simple problems;

3. The CPU utilization factor is low even when corrected by the factors estimated above (adjusted value); to justify the claim that the response time is inadequate when more than a few terminals the following factors should be considered:

- most probably the system is I/O bound-the bottleneck is not the CPU, but some I/O or storage device
- there is a large amount of overhead in the operating system which is larger than the 25% assumed above
- the use of the system is concentrated in a small portion of the operational time
- the MTTR (mean time to repair) is larger than the 10% of the operational time assumed above
- the data are not reliable;

4. A comparison of the last two columns shows that, in terms of CPU, the system has on the average a reasonable margin of capacity.

d. Frequency of Use of the System

The average number of sessions per user is displayed in Table 8.4. The 'average user' has around 16 sessions per month. The trend in the period was a decrease of 0.17 sessions per user per month, a result which again may be explained by the entry of new users with light usage of the system, probably students.

MONTH	SESSIONS PER USER
FEB/77	15.0
MAR/77	13.8
APR/77	16.9
MAY/77	18.7
JUL/77	15.2
AUG/77	15.3
OCT/77	13.3
NOV/77	18.5
DEC/77	13.0
JAN/78	13.4

Table 8.4 - Average number of sessions per user

Figure 8.2 provides a better idea of the frequency of use of the system; the community of active users each month was grouped according to the number of terminal sessions held by each. The communities are not necessarily the same every month; also, users may change classes from

one month to another. It is possible to divide arbitrarily the users of each month into three classes:

- constant users, which have more than 16 sessions and represent approximately 30% of the total
- regular users, which have more than 4 and 16 or less sessions and amount to approximately 35% of the total
- sporadic users, which have 4 or less sessions and are around 35% of the total.

The relative position of the curves in Figure 8.2 suggest that the monthly distribution of users and sessions is approximately the same for all months. This fact is corroborated by Figure 8.3, which shows the cumulative distribution of users versus sessions for the months of October, November and December of 1977. The existence of constant users with a large fraction of the total number of sessions and of sporadic users with a light fraction but representing a significant portion of the number of users is obvious:

- 10% of users held more than 30% of the sessions
- 30% of users held less than 5% of the session.

e. Distribution of the Workload by the Time of the Day

Figure 8.4 displays the curve of the average of the maximum number of active users in slices of 30 minutes

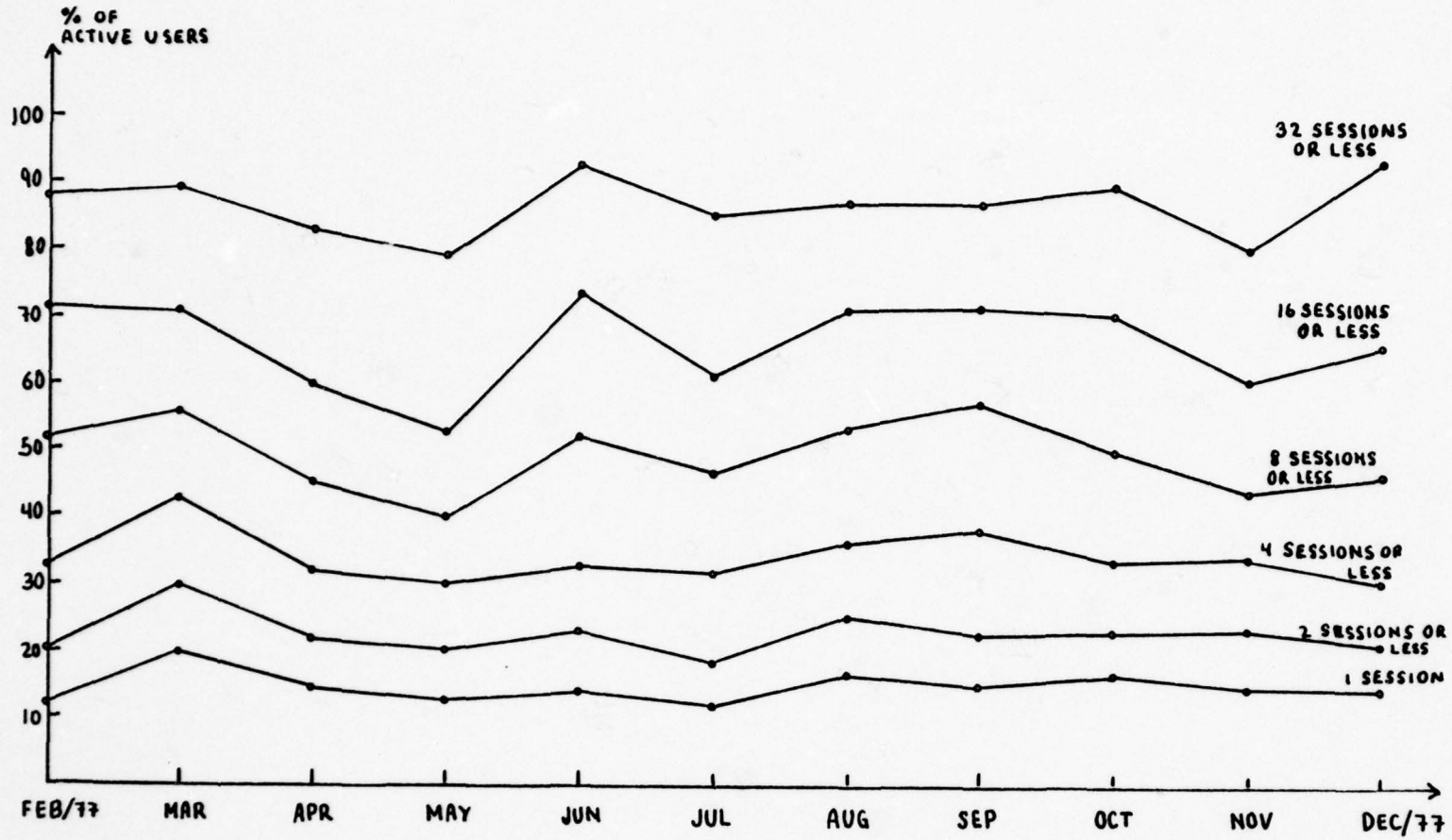


Figure 8.2 - Number of sessions versus percentage of active users each month

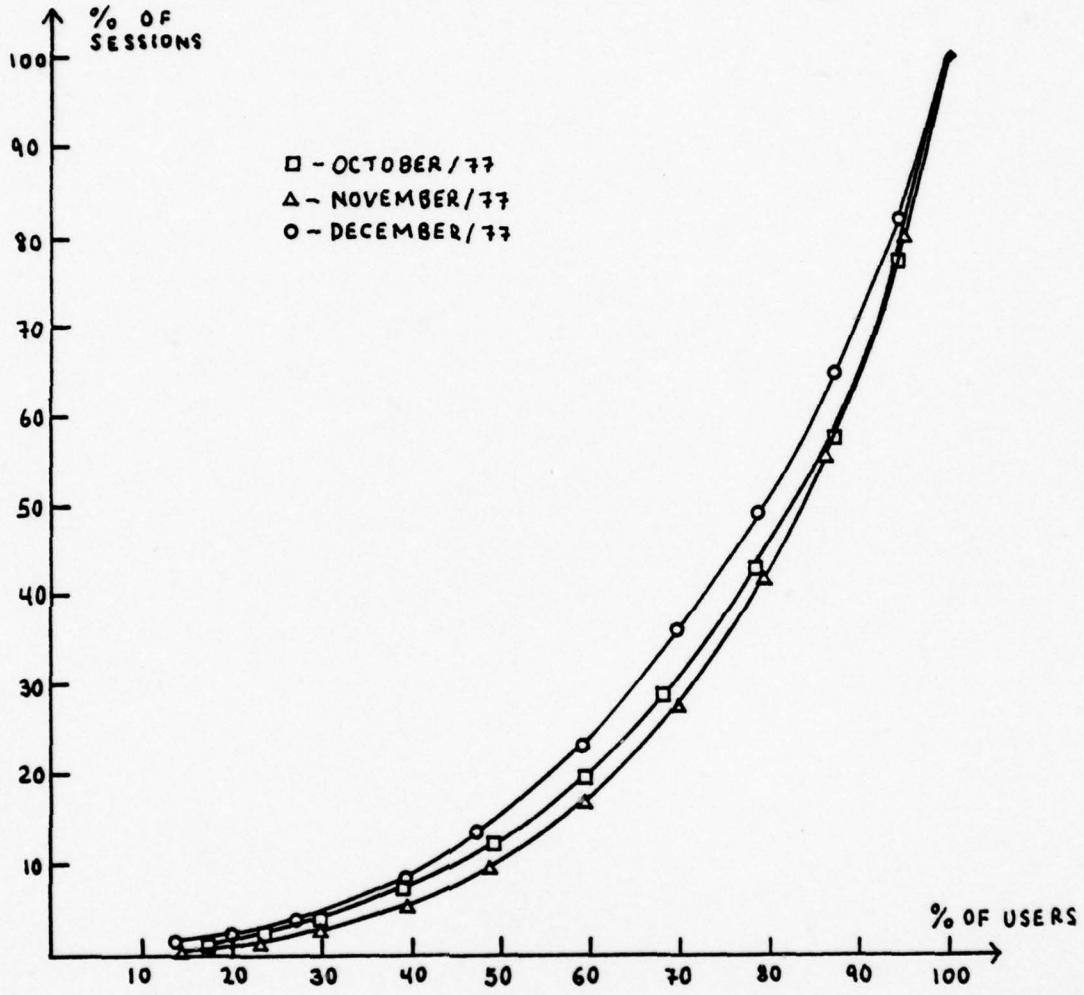


Figure 8.3 - Cumulative distribution of users versus sessions

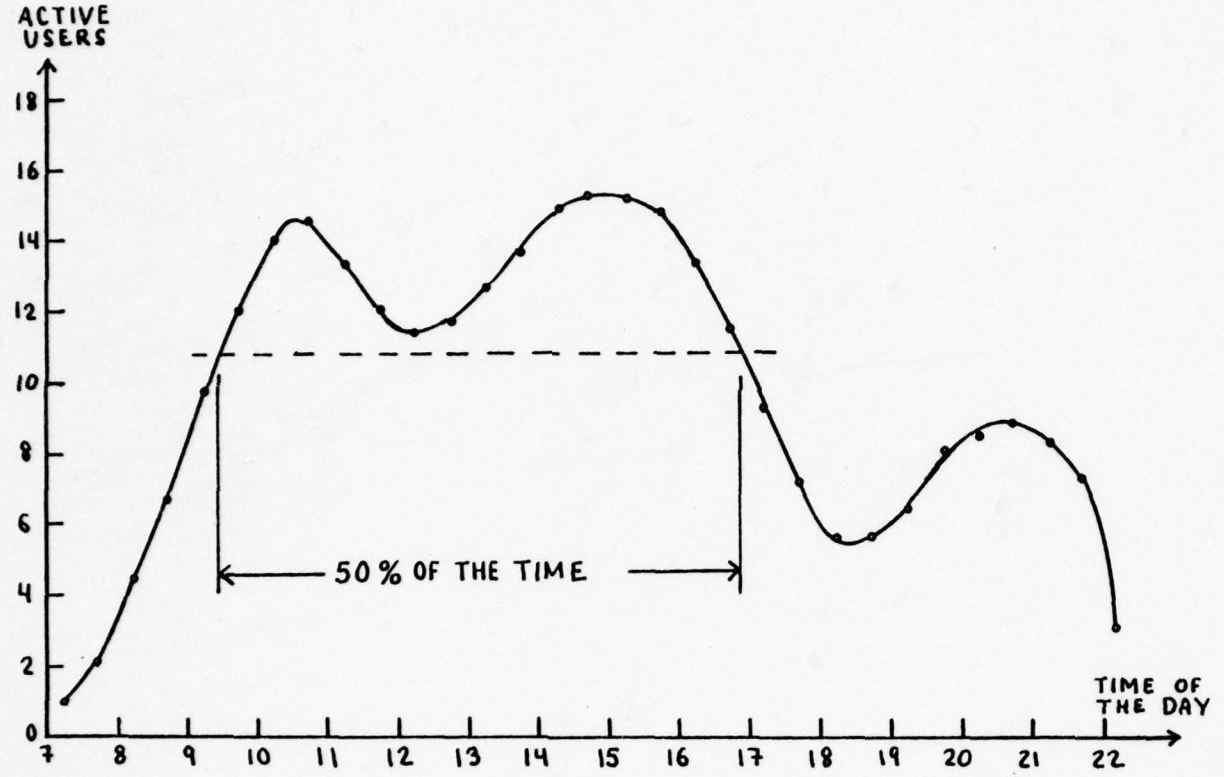


Figure 8.4 - Expected value of the maximum number of active users by the time of the day

during a day calculated over the entire period studied with the exception of June and September of 1977. This curve was obtained by summing the maximum number of simultaneous active users in each half-hour slice over the period and dividing this sum by the total number of periods for each slice. The calculation did not include failures, but, if a uniform distribution of failures along the day is assumed, this factor would not change the shape of the curve.

Three local maximums are recognized:

- around 1030 in the morning
- around 1500 in the afternoon, which is also the absolute maximum
- around 2030 in the evening.

Two local minimums are present:

- around 1200
- around 1830.

The afternoon peak is equal to 15.1 and lasts longer than the morning peak.

The dashed line divides the curve into three regions. The upper part represents 50% of the system working hours and is the period where most of the work is performed. In this period, which extends from 0930 to 1700, the expected value of the maximum number of simultaneous active users is always larger than 11.

The steep slope of the curve by the end of the operational time suggests that users could extend their work

beyond 2200 if the system were available.

The curve of Figure 8.4 suggests that the terminal utilization factors calculated by method 2 in Table 8.2 are reasonable because they give expected values between 6 and 13 for the number of connected terminals, which is consistent with the curve; if the values provided by method 1 are used, this does not occur.

The maximum number of simultaneous users logged on the system in the last quarter of 1977 was 26. The maximum recorded in the examined period was 35 simultaneous users.

C. DESIGN GOALS

Based on the characterization of the problem and the characteristics of the NPS environment, a set of desirable attributes for the new system is developed in the following.

1. Accessibility

The computer power should be spread throughout the campus where the need for it exists. Terminals for the time sharing system should be evenly distributed; points for entry of jobs and reception of outputs of the batch system should be dispersed around the campus to avoid crossing the campus to do computer work.

2. Modularity

The average life span of a computer system of this type is about 8 years. In this time the needs of the NPS in terms of computing power may increase considerably in

volume. The system should have an architecture which can easily absorb this growth by allowing expansion to occur smoothly.

3. Adaptiveness or Evolvability

The school is a research center. The nature of the research in the government and in the military is very dynamic, reflecting changes in national policies, and world conditions. Also, technology is progressing at unprecedented rates, making predictions very difficult. The new system should be adaptive, i.e., it should be adaptable to changes in needs and technology.

4. Reliability

A large part of research and academic work depends on the use of the computer for its completion. The system measure of interest to the user is availability, which is the expected fraction of the operational time in which the system is available to its users. Availability is a function of the MTBF (mean time between failures) which is a measure of reliability, and the MTTR (mean time to repair) which depends on many administrative and technical factors. The MTTR plays an important role in the system availability but, at the time of initial system design, there is little or no control of it. On the other hand, reliability is strongly influenced by the system architecture. Thus, for the purpose of comparing architectures, reliability is more meaningful. Criteria for reliability must be established.

The following criteria meets the purpose of this study: the probability of no failure in a specified time interval such that would put one or more sites out of operation. The goal for the NPS time sharing is a highly reliable system.

5. Performance or Effectiveness

Performance or effectiveness is the degree to which the system meets the demands of its assigned tasks. In time sharing systems the performance is mainly measured by the system's response time, i.e., the time elapsed between the moment that a user submits a request for service and the reception of output. Loosely speaking, the system should present an adequate response time to users' requests; response time is subjectively judged to be adequate when it meets user expectation, which depends on the complexity of the request made.

To handle the multitude of different needs at the NPS with adequate quality of service, the system should exhibit generality and flexibility. Generality relates to the ability of handling diverse problems and flexibility to adapt to particular problems presented by the users.

Another important factor in system performance is the ability of graceful degradation, which is a scheme whereby performance will be reduced as elements of the system fail, avoiding a total collapse, up to a limit which depends on the type and combination of the failed elements. The NPS system should provide graceful degradation to allow

work to continue during failure periods.

6. Simplicity of Use

The existence of a significant number of users with a reduced number of sessions each month was substantiated. It is reasonable to assume that many users are permanently included in this class of light usage (this fact cannot be deduced from the data). Such users typically do not develop great intimacy with the system. Also, many students are introduced to the system every quarter. Thus, the system should provide simplicity of use for fast learning.

7. Ease of Operation

A corollary of the simplicity of use goals is that users should be required to have only the minimal physical contact with equipment and should not be involved in operations. Thus, professional operators should assist in the use of the system. This also enhances the availability of the system. But, practical and cost considerations dictate the minimization of the number of operators. Thus, the new system should exhibit the characteristic of ease of operation in the sense that it should not require many operators.

D. DESIGN CONSTRAINT

The only design constraint to be imposed is proven technology. This means that the design should be based on commercially available products, hardware or software. Construction of special hardware, large software development and

revolutionary technologies will not be considered. Nevertheless, small modifications in hardware equipments and in software products may be taken into account.

Cost constraints were not explicitly identified. Nevertheless, cost considerations will be considered.

E. REQUIREMENTS

Many requirements for the time sharing system were identified in this and the last chapters. These and others will be briefly listed in the following:

1. High Level Languages
   a. APL
   b. BASIC
   c. FORTRAN
   d. PASCAL
   e. A system programming language.
2. Debugging Facility

Most of the debugging facilities are embedded in the language processors. In addition to these the system should provide an interactive debugger with general application to facilitate program development.

3. Line Editor

A line editor of general application should be provided for the editing of programs in any of the languages offered.

4. Text Editor

Editing of text is needed for preparation of reports

and theses. Provisions for handling mathematical format should be considered (even if all mathematical symbols are not included in the character set).

5. Integrated Batch-Time Sharing

A program interactively developed in the time sharing mode should run in batch mode. Conversely, a program entered in batch mode should be callable from time sharing.

6. System Library

A set of high level and pre-compiled routines to be read and incorporated into users' programs.

7. User Library

Means of storing users' files for future use.

8. Calculator Mode

The calculator mode of terminal operation pertains to an algebraic language of easy use with extended precision, which is intended for solving computational problems in the manner of a hand-held or desk calculator.

9. Mail System

A system for exchange of messages should be considered for communication among users even if the recipient is not logged on the system at the time that the message is sent.

10. Data Base Management

A data base management system for interactive use should be provided. The exact needs in this area have yet to be studied; the NSA, AS and administrative departments

need this capability.

11. Graphics Terminals

There is the need for graphics terminals for interactive graphical work. This usage is typically analysis of curves for engineering, mathematical and statistical work. Terminals based on a DVST (direct view storage tube) seem to be adequate for most of these purposes; they have the advantage that they do not require refreshing, which alleviates computer processing considerably. They should also work in 'character mode' for common use and have a means for copying screen contents.

12. Command Language

This is an integral part of any time sharing system and pertains to the set of commands and rules available to the user for utilization of the system resources. It should exhibit good characteristics of human engineering as dictated by the goal of simplicity of use, while allowing complex operations as dictated by the performance goal.

13. Security

Security in the computer field is viewed in many aspects. Those aspects relative to classified work will not be addressed; they should be studied in light of the specific regulations to determine the suitability of the system and the methods needed for implementing this kind of work.

Security may be viewed as involving authority of access, that is, as a question of establishing who (users,

processes) can be allowed to access what (files, processes, systems resources in general). Security may be externally imposed, as in the protection of files from unwarrented examination (privacy), or may be internal, e.g., protection against a program bug. Internal security is included in most operating systems.

External security may reflect a hierarchy of authority which may be very intricate. Also, security measures consume system capacity and may be very costly. Thus, security measures should not be imposed beyond their value. It has been said that the only system with perfect security is the one that does no useful work.

The implementation of security is a matter of building locks and keys. Such locks can be implemented in hardware, software or both.

For the NPS, where the entire computer system is on campus, low level external security might be adequate. A hierarchy of authority dividing users into three classes is suggested:

a. super users-those with unlimited access, usually the operators;

b. priviledged users-those with access granted to most of the system resources, but subject to some constraints, such as access operating system software and certain data bases; these users are usually members of the staff and faculty;

c. general users-those restricted to the resources of general use.

The users' library should have three kinds of access priviledge; proprietary files with access restricted to the owners, read-only files with general access to read, and read-write files with total access allowed to anyone. The owners may be a group or a single person. Access priviledge should be established by the owner when the file is created and only he should be allowed to change the access priviledge.

A simple scheme based on the use of passwords should be adequate. For system access and file access the user must specify the correct password prior to receiving access permission.

IX. PROPOSED ARCHITECTURES

A. INTRODUCTION

In this chapter four system architectures will be presented. The degree to which each one meets the design goals and needs previously identified will be discussed. They will be presented in an evolutionary way, from the simple to complex. The aim is to disclose their strengths and weakness in light of the postulated design goals. These goals are adapted in the following way:

1. accessibility is excluded from the consideration because all system architectures presented satisfy it equally;

2. the factor of most significance in the performance goal, during this initial phase of design, is graceful degradation; this will be emphasized in the discussion;

3. simplicity of use pertains almost exclusively to the implementation part of the design and depends on the characteristics of the interactive operating system adopted and thus will be not emphasized.

The level of abstraction of the block diagrams presented will be the highest possible; only those details of significance to the discussions will be shown.

The existence of a batch system will be assumed, as already mentioned; the remote job entry units (RJE), which are initially considered part of the batch system, will appear in the drawings and will be considered in the communications needs and in the system integration discussion.

1. The NPS Campus

The NPS Campus, shown in Figure 9.1, is not large compared to many universities. A small region inside the campus contains most of the academic work. This region is the rectangle formed by Ingersoll Hall, Root Hall, Spanagel Hall, Bullard Hall and Halligan Hall. The external dimensions of this academic rectangle are 317 meters by 97 meters, while the yard internal to those buildings measures 256 meters (Ingersoll-Spanagel) by 41 meters (Root-Halligan).

The time sharing system will be fully contained in the academic rectangle; needs outside this area, such as terminals for administrative use and for laboratories not included there, were not explicit and will not be considered. Nevertheless, the extension of the system to be presented to other regions can be easily accomplished when necessary because of the modularity of their architectures.

Ingersoll Hall, where the computer center is located, will be referred to as the central site and will house the batch system and a part of the time sharing system, which will vary according to the architecture. Terminals will also be located in this building to support its users.

Three remote sites are identified:

a. Root Hall

b. Spanagel Hall, and

c. Halligan and Bullard Halls.

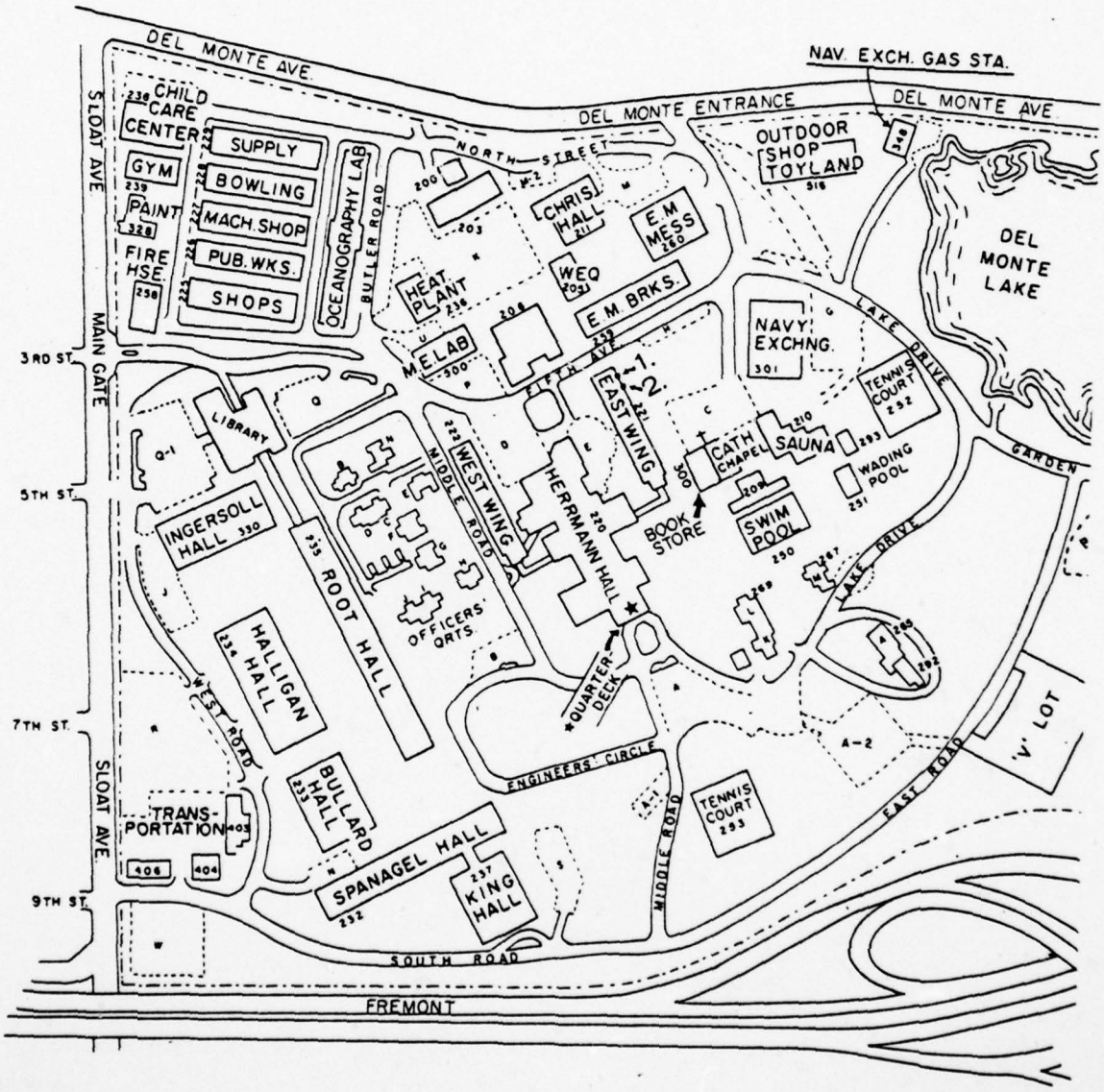


Figure 9.1 - The NPS Campus

Each of these remote sites will contain a remote job entry station (RJE) for the batch system, terminals and other hardware.

Two characteristics are of importance in the definition of the communications facilities to be employed:

a. system is entirely in-house, and

b. very short communication distances.

The best solution, considering these two factors, is to build NPS proprietary communications facilities for the new system. Physical transmission lines are the most suitable technique for such short distances because of low cost, high reliability and absence of maintenance requirements. The cost of all communications links needed will be, in this way, a small fraction of the total cost of the system. Unlike geographically spread networks, minimization of communication cost does not have importance here.

2. Centralized and Distributed Solutions

By a centralized system is meant a computer system based on one CPU; by contrast, distributed systems are those based on more than one CPU's, each performing part of the total work.

The computer field was dominated since the beginning by the trend of centralization. With the advent of minicomputers this trend began to reverse and it is now clearly reversed. The main advantages attributed to centralization are generality, ease of control and economy of scale. But,

compared to distributed systems, they have, in general terms, poorer modularity, evolvability, reliability and graceful degradation. These are an important part of the goals for this design. Thus, the approach of this study is in favor of distributed systems.

The question now is how well a distributed architecture matches the NPS workload. Clearly, by reasons already mentioned, there is the necessity of a powerful machine for support of research in the NPS. It is also evident that a set of minicomputers cannot effectively handle many classes of large scientific problems that a powerful mainframe can handle. But two considerations have to be taken into account:

a. the requirements for the batch computer have already defined a large mainframe;

b. the average work that is performed in the NPS time sharing system does not require large computational power; indeed, the 'average user' employs a modest amount of CPU time per session (72 seconds for a session of 40 minutes).

Today's minicomputer-based time sharing systems are being used in engineering and scientific environments with the same class of problems as those of NPS. These minicomputers may be adequate for handling problems similar to those found at NPS.

This design is based on the principle that the batch system will be in charge of the more complex problems that require much memory and computational power. It might be possible to have priviledged users working interactively on

this system with schedule restrictions. Most of the interactive work will be performed in the distributed time sharing system.

B. ARCHITECTURE 1

Figure 9.2 displays a simplified block diagram of architecture 1, which has the following characteristics:

1. The RJE of each remote site is directly connected to the FEP of the batch computer (BC);

2. The time sharing computers (TSC) communicate directly with the BC;

3. The TSC's are independent from the BC and are self-sufficient, having their own operating system, and managing their own system and user library;

4. The FEP of the BC performs communications between the BC and TSC's or RJE's, among the TSC's and between TSC's and RJE's;

5. All TSC's are identical and run under the same operating system;

6. In each site the number and type of the terminals are determined according to the local needs;

7. TSC's may work in three modes:

   a. stand-alone time sharing;

   b. time sharing with support of the BC for some processes;

   c. transparent mode in which they act as bridge to users logged on the BC.

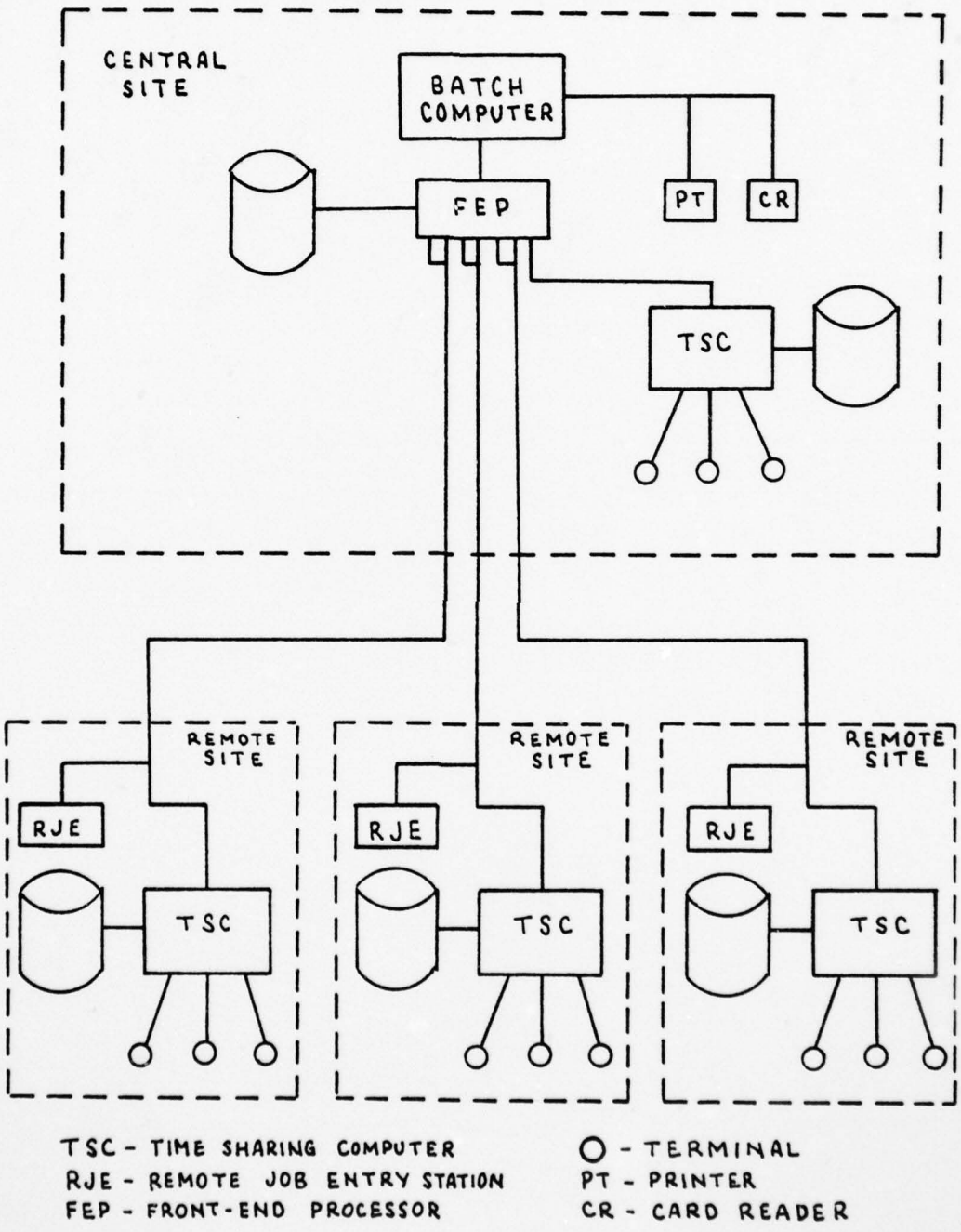


Figure 9.2 - Architecture 1 block diagram

This type of system is often called hierarchical, because the system components are interconnected to form a tree-structured hierarchy. The capability of the components grow with the level which they occupy in the 'hierarchy'; at the highest level is the central computer (the BC here) which provides supporting services for the lower levels. The degree of dependence of lower levels on higher levels vary; for example, the central computer may provide compilation, linking and simulated testing of programs for the lower level computers. In the extreme there is full dependence in a master-slave relationship. In this design the dependence of the TSC's on the BC was kept at a minimum due to reliability and graceful degradation goals.

RJE's in this architecture coul be connected to the TSC's, but, being connected to the FEP, the TSC's are alleviated of the communications functions relative to them and the overall graceful degradation of the system is better. This improvement in the graceful degradation is due to the fact that the RJE's are used for both the time sharing and batch modes; if they were connected to the TSC's, the batch inputs and outputs through a particular RJE would stop in the event of a failure in the TSC. On the other hand, the time sharing work is only slightly affected when an RJE does not function for any reason. Finally, there is no reason to connect RJE's to the TSC's; the TSC's would be handling the communications and input/output control via the RJE in this way; then the

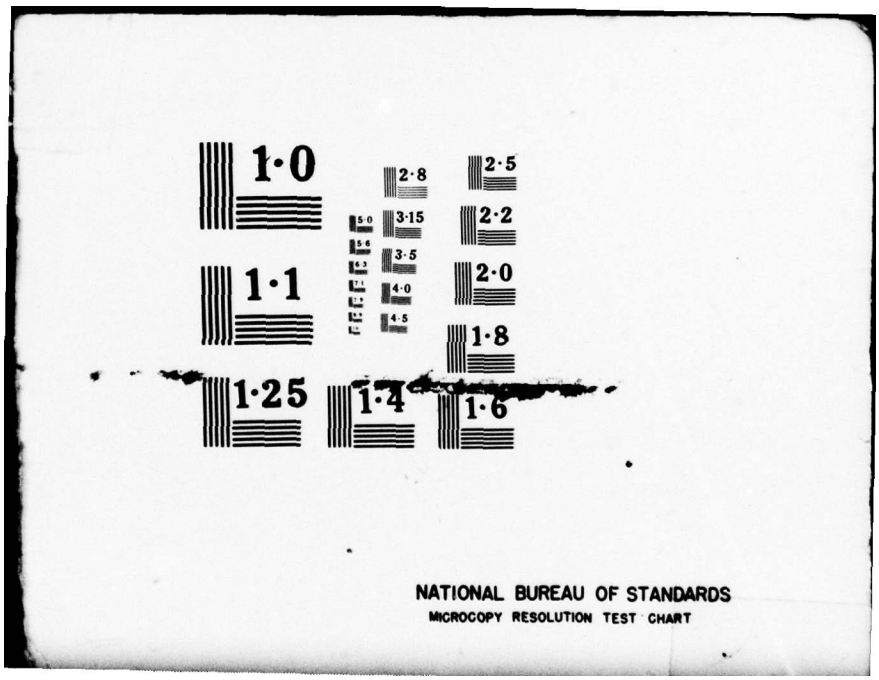
AD-A057 906 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 9/2
COMPUTER NETWORKS. ANALYSIS AND A CASE STUDY DESIGN.(U)
JUN 78 I D ROCHA

UNCLASSIFIED NL

3 OF 3
ADA
057906



END
DATE
FILMED
10-78
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

use of line printers and card readers would be more cost effective.

Users being allowed to log in at any public terminal poses a problem regarding the policy of allocation of files in the user library. A user may need a file which he had previously created but which is not located in the TSC he is logged onto. A simple solution is to store the user library in the BC, but this will result in poor graceful degradation in the event of failure at the central site. Obviously, the user's library has to be distributed. Two schemes may be devised: static allocation and dynamic allocation. In static allocation the user's files have fixed residence reflecting the adherence of the user to a particular site. When the user is working outside his 'base', the files needed are shipped to him. At the end of his session, all the files which he wants to save are shipped back to his local 'residence'.

In dynamic allocation the files migrate according to need; current storage is where a file was last used. The problem is to find a particular file; this may be done by storing a directory in the BC, which requires extensive tables, and is not good for graceful degradation purposes, or by having the BC poll all the TSC's until the needed file is located; this is practical for only a small number of TSC's.

Because of its simplicity and taking into account the fact that users in the NPS tend to be localized, static allocation is recommended.

One advantage of this architecture is that the facilities of each site may be tuned to the specific needs of its users. This may seem atractive but, considering that the objectives of this design are a general purpose system and simplicity of use, this is not recommended beyond providing special routines in the system library at each site. The operating system and the bulk of the system library must be the same at every site.

The type of terminals to be employed at the lowest level of the hierarchical tree may be as diverse as desired. The architecture does not impose any restriction in this regard. On the other hand, maintenance efficiency and the simplicity of use goal both call for standardization. Then the diversity of terminal types should be kept to the minimum necessary. Microprocessor based terminals with stand-alone capability are attractive because they may perform text and program editing, thus relieving the TSC of such tasks; if, in addition, they are coupled to floppy disk units, users can keep their files on diskettes, which saves space in the TSC disks. However, this solution has two drawbacks: it may not be cost effective and it does not provide good simplicity of use because users have to learn to work in two environments, viz, the terminal monitor and the TSC operating system.

The evaluation of this architecture according to the design goals follows.

a. Modularity

The architecture has good modularity. If the time sharing needs grows, the number of TSC's may increase accordingly.

b. Evolvability

Evolvability is fair. The architecture does not prevent the incorporation of new technology; there are two solutions for accomplishing this in a campus-wide, beneficial way; the first is to add the new capability to every site, which may not be economical; the second is the enhancement of the BC, which is not intended for general time sharing use, so this is not recommended.

c. Reliability

Reliability is only fair. Failures in the central site does not affect time sharing, but one TSC failure puts one site out of operation.

d. Graceful Degradation

Graceful degradation is poor. When one TSC fails, the files stored at this TSC are no longer available to users. Duplication of hardware at each site improves graceful degradation but is too costly.

e. Ease of Operation

Ease of operation is poor. Operators have to be spread throughout the campus or have to leave the central site and walk to the remote sites to fix bugs, perform necovery or start-up routines and make checks or tests.

A factor to consider is the division of terminal load, among TSC's. Since each site will have one or two TSC's, the number of terminals connected to them may vary from site to site to the point where one TSC is overloaded, while the other one is underutilized.

C. ARCHITECTURE 2

The evolution from architecture 1 to architecture 2, shown in Figure 9.3, is quite natural due to two factors:

a. the remote sites have identical hardware;

b. savings in communications bandwidth are irrelevant for the NPS system.

Architecture 2 has the same logical organization as architecture 1. The only new element is the interconnection unit between the terminals and the TSC's. It provides freedom of allocation of terminals to TSC's. It may be a passive unit and may be as simple as the patch-boards used in communication stations. From the point of view of the user architectures 1 and 2 are the same. But architecture 2, concentrating system resources at the central site, is much more advantageous in relation to the design goals:

1. Modularity

Modularity is very good. The system can grow in small increments because the TSC's are shared by all sites. The central position of the TSC's enable them to serve more than one site; consequently the total number of terminal lines at the central site can be divided among the sites

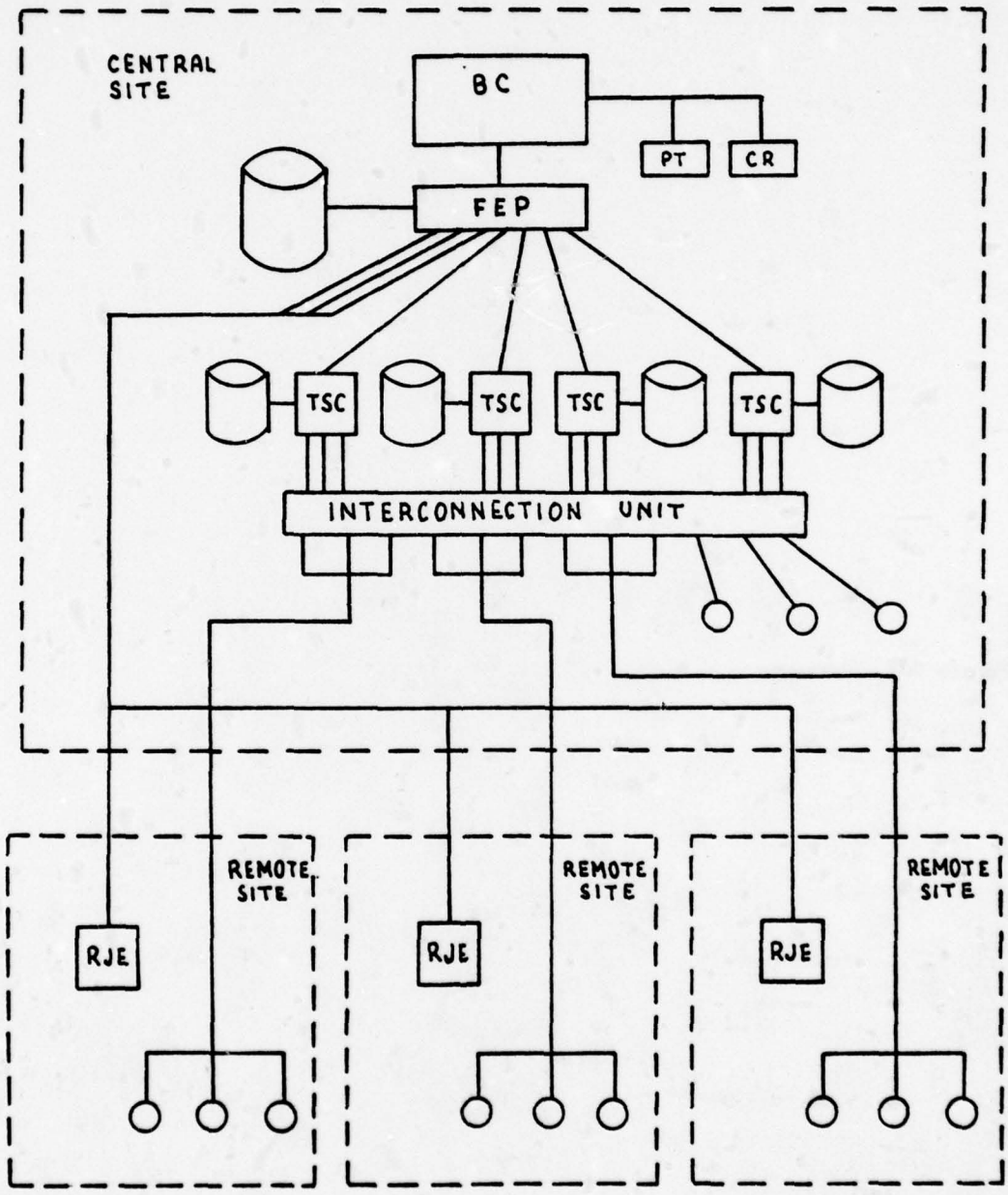


Figure 9.3 - Architecture 2 block diagram

according to needs.

2. Evolvability

Evolvability is good. The incorporation of new or specialized technology in the central site may be shared easily by all sites; the problem is only that a new special processor will work with dedicated terminals.

3. Reliability

Reliability, according to the criterion established, is very good. High reliability is achieved by allocating an equal fraction of the number of terminals of each site to each TSC. The interconnection unit has a reliability greater than any of the other system components.

4. Graceful Degradation

The graceful degradation of architecture 2 is very good. If the TSC's are not used to their full capacity in terms of number of terminals, the system has an excess of capacity. It is worth noting that this excess serves to provide better response time. When a TSC fails, the corresponding terminals can be reallocated to other TSC's. User files are not locked in the failed TSC, because operators can transfer the disk drives or even the disks packs to a good processor.

5. Ease of Operation

Ease of operation is good as a result of the concentration of hardware at the central site. However, many reconfiguration procedures have to be performed manually.

The division of load is another advantage of this architecture over the previous one. In architecture 2 the TSC's may be allocated equally to the terminals. This can be referred to as statical load balancing.

Due to the fact that at each site there are terminals supported by every TSC, the user file traffic among TSC's will be high because in each session a user may use a different processor, even at the same site. This should not be a problem because this file transfer is done mostly at log-on and log-out using the statical allocation scheme.

D. ARCHITECTURE 3

In architecture 3, displayed in Figure 9.4, there is no hierarchy. All the processors, independent of type and capability, are at the same level.

The message switch complex (MSC) is a network (or subnetwork) of message switches (or message concentrators acting like switches). Each message switch has a dual configuration for reliability purposes. This is not shown in Figure 9.4. The MSC uses a standard interface to all processors. In the terminal side there is no interface standard, so any terminal may be made. At the computer side the line speed is of the order of 50 to 200 kpbs, while terminal line speed may vary from 10 to 10,000 bps. Each message switch is typically connected to three processors and two other switches. Processors are connected to more than one switch.

The MSC has the following functions:

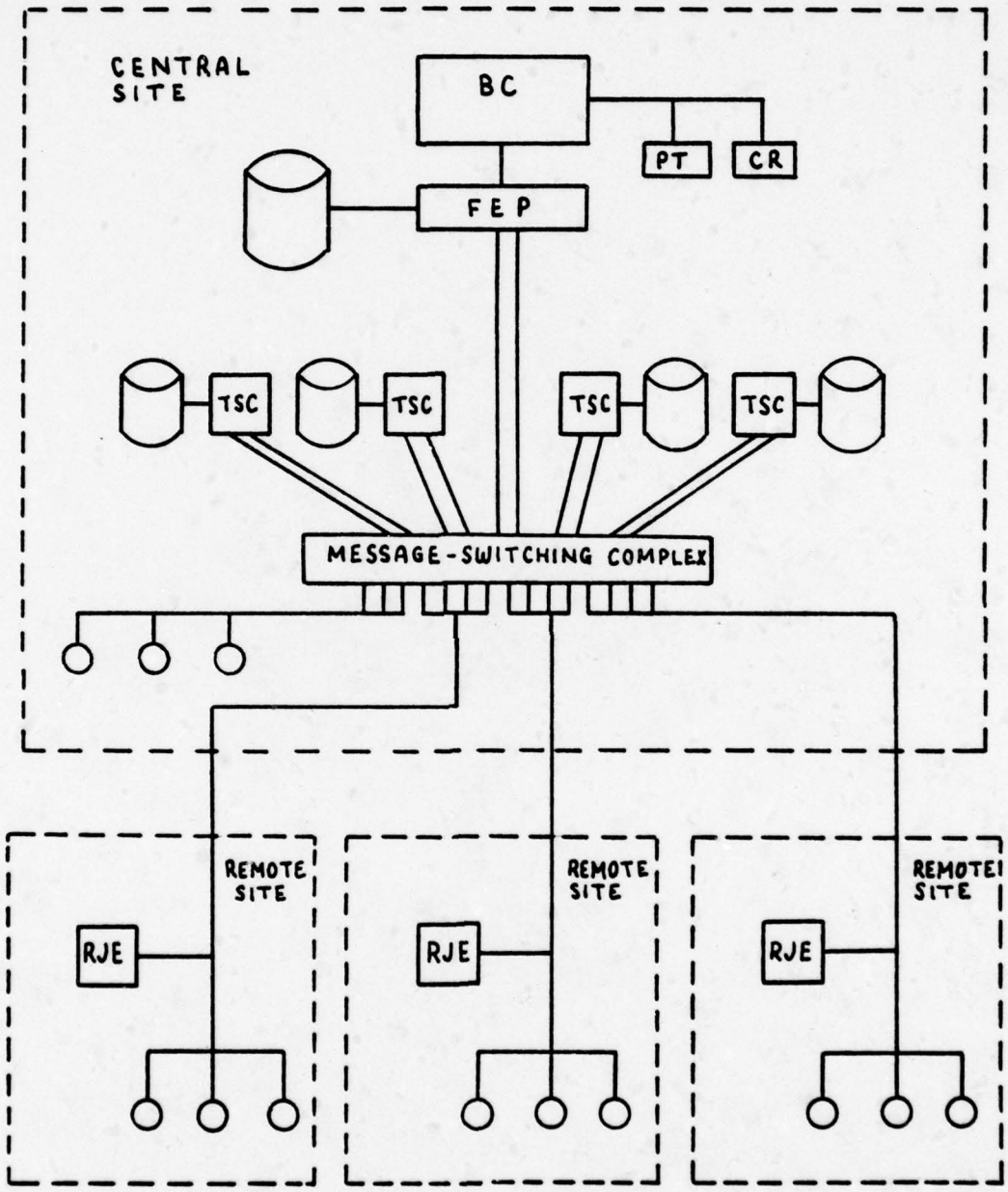


Figure 9.4 - Architecture 3 block diagram

a. to perform log-on and log-out procedures;

b. to route messages from terminals to computers and vice-versa;

c. to perform inter-computer communication;

d. to route RJE-processor communication.

With the exception of task 1, all other tasks are typical of those performed by a message switch. The simplicity of this task, however, should not burden the MSC.

Log-on procedures may be of two types:

a. Invisible - where the user does not specify the resource he wants; he is automatically assigned to one of the general purpose TSC's;

b. Directed - where the user specifies resources (generally the BC or special hardware).

To avoid unnecessary system load and increase in the logical complexity of the MSC software, the security function (permission to log on) is distributed, being handled by each processor.

Each processor has its own system library; except for special processors, it should be the same in all TSC's. The user library should be statically allocated for simplicity. The information about the residence of users' files may be stored in each TSC or may be concentrated in the MSC. The former solution is preferable in order to take the load of controlling file transfers out of the MSC.

TSC's in architecture 3 work in two modes:

a. stand-alone time sharing;

b. time sharing with concurrent processing, in which more than one processor works coordinately to provide service to one user.

The evaluation of architecture 3 according to the design goals follows:

1. Modularity - Modularity is very good.

2. Evolvability - Evolvability is very good; the standard interface for computers in the MSC enables the addition of hardware to the system; also, special resources may be globally shared on campus.

3. Reliability

Reliability is good. Reliability is dominated by the nodes of the MSC. High reliability is achieved by duplication of hardware at each node, by connecting TSC's to more than one switch and by double connectivity inside the MSC. The price of high reliability is increased cost and complexity. Terminals at each site should be equally distributed among the switches of the MSC; in this way a failure in one switch has equal effect on all sites but does not cause a particular site to stop operation.

4. Graceful Degradation

Graceful degradation is fair. When one switch fails, some of the terminals go down; reconfiguration capability for terminals is not present in the block diagram in Figure

9.4; because the number of switches is small, an interconnection unit between terminals and the MSC is not of much help; this fact imposes a stringent requirement for highly reliable switches; it is worth noting that it is not possible to have an interconnection unit in parallel with the MSC because terminal and computer lines are not compatible; for the rest of the system components the graceful degradation is very good; failed processors are isolated by the MSC and user files are transferred to other processors (manually by operators or automatically by a reconfiguration unit).

5. Ease of Operation

Ease of operation is very good. One advantage of this architecture is that it may provide dynamic load balancing, i.e., users are assigned to computers in a way which tends to equalize the load as measured by some parameter, for example, the number of users on each TSC.

One problem with architecture 3 is that the characteristics of message exchange among computers are different from those between terminal and computers. The conversation between terminal and computers tends to be done in short blocks while file transfer among computers is generally performed in large blocks. Both types are handled by the MSC; the file transfer can interfere badly with the interactive traffic causing delays. Also, the speed of the lines on the computer side (50 to 200 kbps) is not high for file transfer. A favorable point is that the average size of

files is not large in the NPS environment. One solution is to provide a parallel system dedicated to the intercomputer traffic; the MSC is then released of the task of file transfer and can be tuned to the characteristics of interactive traffic.

E. ARCHITECTURE 4

In architecture 4, shown in Figure 9.5, there is no subordinate relationship among processors; all are at the same level and are independent. They are grouped into a network by a loop communications facility. The loop consists of very high speed unidirectional lines and loop interface units (LI) with parallel by-pass. The capacity of the lines is on the order of several megabits per second; for example, 10 Mbps lines may be used; in this case, a 1 Mbyte file is transmitted from one LI to its neighbor in 800 miliseconds; thus, fast file transfer is achieved, even if a file has to travel through several LI's. The delay imposed by an LI is small because, after it recognizes that the file is not addressed to its processor, it may begin the transmission to the next LI without the need for buffering the entire file; in this way, error correction, detection and acknowledgement are only performed at the destination LI. To achieve a balance between file transfer and interactive traffic in the loop and also to limit the size of buffers in the LI's, a maximum of message block must be established.

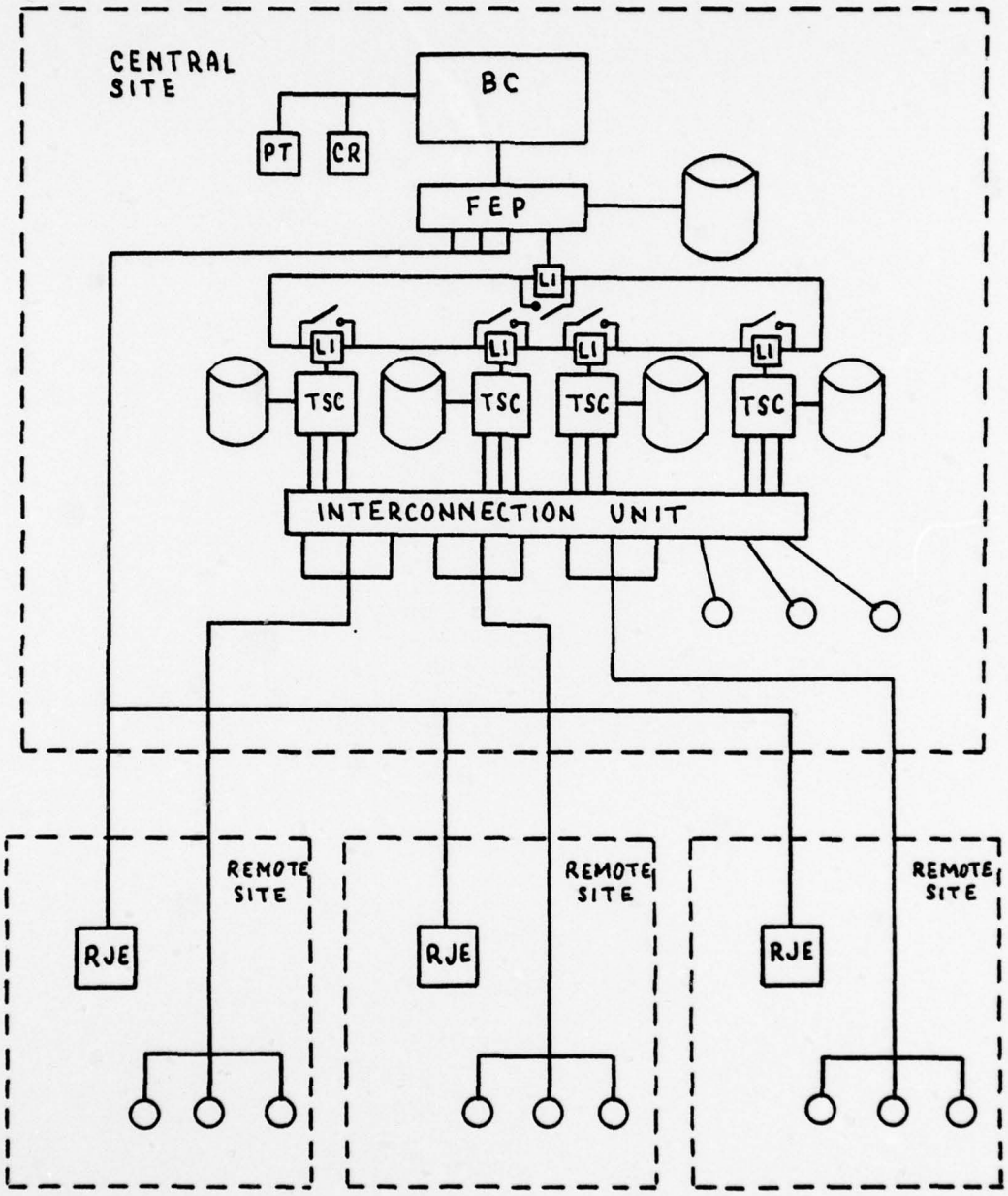


Figure 9.5 - Architecture 4 block diagram

Assemblage and disassemblage of messages may be handled by the processors.

Due to the proximity of the processors, the loop lines may be implemented in parallel form; in this case the transfer speed between LI's can go to several tens of megabits per second. A more simple and economical way of implementing the loop will suffice for the NPS. Super high speed connections are not needed because the bulk of the work is performed inside one single TSC. The use of high speed serial links for information transfer in parallel with low speed links for signaling (out-of-band signaling) is suggested; this can lead to easy LI implementation.

The RJE's are directly connected to the FEP of the BC. Inputs to TSC's via RFE's are stored in the FEP and transmitted to the respective TSC through the loop upon request. Output of TSC's for printing are routed directly to the specified RJE.

Addressing in the loop may be hard or soft. In hard addressing, messages are addressed to machines, while in soft addressing, they are addressed to processes. Soft addressing has the advantage of machine independence but the address recognition in the LI is more involved. Hard addressing, on the other hand, is simple to implement and minimizes the message stop in LI's. For this reason hard addressing in the loop is preferred; in this case the mapping function may be performed by the LI or the processor.

The by-pass (see Figure 9.5), in parallel with the LI's, serves the purpose of 'disconnecting' failed LI's from the loop.

The interconnection unit is identical in characteristics and functions to the one of architecture 2.

The problems of distribution of libraries and file allocation are the same as those of architecture 3, because in both architectures the processors are independent. Thus, the solution of static allocation for user files is again recommended. The system library should, of course, be replicated in every TSC. TSC's have three main modes of work in this architecture:

1. stand-alone time sharing;
2. time sharing with concurrent processing;
3. transparent mode, in which a user whose terminal is connected to one TSC is really logged onto another processor; the 'local' TSC acts as a repeater or bridge.

An overall evaluation of architecture 4 follows:

1. Modularity

Modularity is very good. Processors may be added to the loop as needed. Even processors not located at the central site may be integrated by means of an LI located there. While it is not recommended, the loop can be extended throughout the campus.

2. Evolvability

Evolvability is very good. Any type of processor can

be integrated; by the use of a standard interface, the loop acts as intermediary between two different processors.

3. Reliability

Reliability is very good. Failure in one TSC does not stop all the terminals of a site and does not affect the others. The only concern is the reliability of the loop links, but they have a reliability well beyond that of any other system component.

4. Graceful Degradation

The graceful degradation is very good. Terminals may be connected to any TSC, which enables the replacement of a failed processor. Failures in LI's are isolated by the by-pass.

5. Ease of Operation

Ease of operation is very good because of the concentration of hardware at the central site.

The simplicity of this architecture makes implementation relatively easy. LI's range from very simple to sophisticated units; this depends on the division of interface tasks between each processor and its LI.

One disadvantage of architecture 4 is the possibility of a bottleneck in the loop. The characteristics of work in the NPS, where most of the work can generally be done on one TSC and where the amount of file transfer is smaller than interactive traffic, leads to the belief that a well designed loop will have a very low probability of congestion.

A very desirable feature of systems like those discussed in this chapter is the ability to perform dynamic load balancing. It is quite possible in those systems to have one processor overloaded while others are idle. Architecture 4 may provide for dynamic load balancing by having the TSC's periodically exchange load information; when a new log-on request if completed, the TSC involved will render the service or will transfer the user to another processor, depending on the load equalization policy implemented. Another possibility is to make a slight modification of the architecture; by replacing the interconnection unit with a parallel set of switches (message switches or PBX's), the responsibility of load balancing can be delegated to the switching complex. This is similar to the MSC of architecture 3 with the difference that switches communicate with processors by terminal-like lines. This solution increases the cost of the system and does not have a good effect of reliability, but it decreases the complexity of computer interaction for load balancing purposes and it diminishes the traffic in the loop. Because of the reliability problem it was not recommended.

F. COMMUNICATIONS FACILITIES

The use of physical transmission lines was previously recommended for the system interconnection. The suitability of the options available for the implementation of the commu-

nications facilities can be analysed in the light of the communications requirements.

An approximate derivation of the requirements is performed in the following. Based on that, transmission lines and associated concentration techniques (for multiplexing or concentration) will be compared.

The situation to be studied is that of architectures 2 thru 4, where the remote sites have one RJE and a set of terminals which communicate with the central site.

1. RJE Communication Data Rates

The majority of the RJE's available in the market require communication lines of 9600 bps or less of speed. Since the type to be used in the NPS system has not been determined, an upper bound in terms of speed for RJE communication will be derived.

It is reasonable to assume that RJE's will be used for printing and card reading only and that printing speeds of 1000 lpm and card reading speeds of 1000 cpm are adequate for the remote sites. Considering the present school facilities and usage, it can be concluded that these specifications are adequate.

RJE maximum communication speeds are related to maximum printing and reading speeds. The output transmission speed should be equivalent to the reading speed; if the former is larger than the later, line underutilization results; if it is slower the buffer size needed grows. Simularly input

speed should be matched to the printing speed in order to keep the input buffer size small and the line utilization high. A very simple estimation will be done.

The reading speed of 1000 cpm is equivalent to approximately 10700 bps, considering 80 columns in a card and a character code of 8 bits.

The printing speed of 1000 lpm is equivalent to 17,600 bps, considering 132 characters in a line and a character code of 8 bits.

Line protocols and error control should be added to those values. Since the printer will not be utilized 100% of the time and buffers are employed, the line speed can be below 17,600 bps. However, a conservative, value of 20,000 bps will be used for this study.

2. Terminals Data Rates

CRT terminals are offered on the market with line speeds ranging from 110 to 9600 bps.

Employing speeds of 2400 bps, a full line of 64 characters is displayed on the screen in less than 250 miliseconds, based on an overhead of line protocol and error control of 25%. This time is adequate since the user's reaction time is much slower. Thus the speed of 2400 bps for video terminals will be used.

For graphics terminals a speed of 9600 bps, the maximum available with many graphics terminals, will be employed in this study.

3. Approximate Number of Terminals

The model adopted for the estimation of the number of terminals is the multiserver queue. Often when a user goes to a terminal room and does not find a free terminal, he goes away and comes back later. The quality of such systems is measured by the probability that a user will not find a free terminal, sometimes referred as 'grade of service'.

The computer center terminal usage data does not distinguish between public and private terminals. The calculation to be made is applicable to public terminals. The geographical distribution of users around the campus is also not known. Some assumptions are made to enable the calculations:

a. users are equally divided among the four sites considered;

b. presently there are 35 public terminals;

c. the average session on public terminals is equal to 40 minutes (as in the general case);

d. the average user interarrival time for public terminals is 60 minutes (pessimistic because the worst case found for one terminal in one year was 57 minutes);

e. poisson arrival rate;

f. exponential service times (session time);

g. public terminals will be concentrated at one place at each site.

Following these assumptions, the overall traffic rate in Erlangs is:

$$Mp = 35X\frac{40}{60} \simeq 24$$

where M is the number of terminals and p the utilization factor for terminals. Then, in one site the traffic rate is 24/4=6.

The probability that no terminal is free is given by:

$$P_B = \frac{\frac{(Mp)^M}{M!}}{\sum_{N=0}^{M} \frac{(Mp)^N}{N!}}$$

Fixing P_B at 0.01 and using Mp=6 the required number of public terminals at one site is 13.

The number of private terminals is then assumed to increase to 20 and to be equally divided among the sites.

Then at each site there will be 18 terminals. Among these it is assumed there are 12 CRT terminals and 6 graphics terminals.

Confidence in these results is obviously poor because of the lack of data. The numeric assumptions made were all biased in favor of increasing the number of terminals. These values can then be used to estimate communications needs.

4. Total Bandwidth Required at Each Remote Site

One remote site will require on the average:

a. 1 RJE (20,000 bps);

b. 12 CRT terminals (2400 bps each);

c. 6 graphics terminals (9600 bps each).

The total data rate at one remote site is then 106,400 bps.

5. Communications Techniques Options

Considering the distances involved and the communications requirements derived above, the options for communications are:

a. TDM and fiber optics;

b. TDM and coaxial cable;

c. modems coupled to telephonic FDM transmitting through a twisted pair;

d. multipair cable of twisted pairs;

e. TDM and twisted pairs.

Options a. and b. are equivalent; the choice is between fiber optics and coaxial cable. There are TDM equipment available in the market that go to 1.54 Mbps well above the requirement of 106 Kbps.

In option c. voice FDM is used in connection with two twisted pairs (equalization needed). In each voice channel a modem is connected and serves one device at the site. For example, if a 24 channel FDM multiplexer is used, 18 modems are required for terminals (2400 and 9600 bps) and one high speed modem of 19.2 kbps (higher speeds are available on the market) is connected to one or more voice channels (for example, to a 12 KHZ channel) as required. Thus 21 channels are utilized

and 3 are free. The advantage of FDM is its modularity (not found in TDM), but his option is more expensive than the others.

Option d. is the most simple of the solutions. Considering the fact that the greatest distance involved is 256 meters in a straight line, it is possible to connect directly the terminals to the computers by means of multi-twisted pair cable (telephonic cable). Allowing for non-straight paths in ducts and wall climbing it may be assumed that, in the worst case, the distance is 750 meters. If the terminals and computer ports employ the differential transmission method of the RS-422 balanced interface, the maximum rate through a single 24 AWG twisted pair at 750 meters is 105 Kbps. Then, employing RS-422, terminals and RJE's can be directly connected to the central site by means of two twisted pairs for each device. The problem is that RS-422 is not widely marketed; in some cases the more popular RS-232 direct connection is possible (depending on duct distance).

In option e. TDM equipments employing RS-422 transmit through one twisted pair.

The final choice depends on many factors. An overall ranking of the options is more a function of cost and practical considerations than of technology.

X. CONCLUSION

In the first part of this work the principles of computer networks were exposed and analysed.

In the second part, a study was conducted with the objective of presenting alternative solutions for a new time sharing system for the Naval Postgraduate School based on distributed systems. An analysis of the characteristics of the computing work at the NPS was performed. Quantitative and qualitative information was derived from the data on the usage of the CP/CMS system at the school. Based on these factors, design goals were formulated. Four system architectures were discussed in the light of the design goals. It was shown that these distributed systems achieve the goals better than centralized systems. Many other particular points of interest of each architecture were discussed. Those architectures are not necessarily tied to minicomputer implementation. Medium scale computers may have advantages in terms of capability and satisfaction of the complex requirements.

An overall evaluation of the four architectures depends on additional factors which were not considered such as cost, administrative policies and market availability. Purely in light of the principles formulated, architectures 2 and 4 are recommended. If simplicity of implementation is stressed, architecture 2 is the solution. If generality and flexibility are most important architecture 4 should be adopted.

LIST OF REFERENCES

1. Abramson, N. and Kuo, F. F., Computer-Communication Networks, Prentice-Hall, 1973.

2. Anderson, E. A. and Jensen, E. D., "Computer Interconnection Structures: Taxonomy, Characteristics, and Examples", ACM Computing Surveys, Vol 7., No. 4, December 1975.

3. Ashenhurst, R. L. and Vonderohe, R. H., "A hierarchical Network", Datamation, February 1975.

4. Becker, H. B., "Let's Put Information Systems Into Perspective", Datamation, March 1978.

5. Beizer, B., The Architecture and Engineering of Digital Computer Complexes, Vol 2, Plenum Press, 1975.

6. Both, G. M., "Hierarchical Configurations for Distributed Processing", Proceedings of COMPCON (Fall 1977).

7. Brown, J. M., "Distributed RJE Terminals Combine with Central DP", Data Communications, December 1977.

8. Carruthers, J. C. and Laughlin, J. R., "Novel Architecture Increases the Data Throughput", Data Communications, March 1978.

9. Chou, W., "Computer Communication Networks-The Parts Make Up the Whole", Proceedings AFIPS National Computer Conference, May 1975.

10. Cotton, T. W., Computer Network Interconnection: Problems and Prospects, National Bureau of Standards, U.S. Department of Commerce, April 1977.

11. Davey, J. R., "Modems", Proceedings of the IEEE, Vol. 60, November 1972.

12. Davies, D. W. and Barber, D. L. A., Communication Networks for Computers, John Wiley, 1977.

13. "Data Communications Terminals-A Survey", Telecommunications, November 1973.

14. Doll, D. R., "Multiplexing and Concentration", Proceedings of the IEEE, Vol. 60, November 1972.

15. Doll, D. R., "Relating Networks to Three Kinds of Distributed Function", Data Communications, March 1977.

16. Emmons, W. F., "Data Network Protocol Standards", Proceedings of IEEE/NBS Computer Networking Symposium, December 1977.

17. Falk, G. and McQuillan, Y. M., "Alternatives for Data Network Architectures", Computer, Vol. 10, No. 11, November 1977.

18. Farber, D. J., "A Ring Network", Datamation, February 1975.

19. Farber, D. J., "Networks: An Introduction", Datamation, April 1972.

20. Fiekowsky, N. S., "Computerized PBXs Combine Voice and Data Networks", Data Communications, June 1977.

21. Folts, H. C., "Interface Standards for Public Data Networks", Proceedings of IEEE/NBS Computer Networking Symposium, December 1977.

22. Frank, H. and Frisch, T. J., Communication, Transmission and Transportation Networks, Addison-Wesley, 1971.

23. Frank, H., "Providing Reliable Networks with Unreliable Components", Proceedings Third IEEE Data Communications Symposium, November 1973.

24. Gray, J. P., "Line Control Procedures", Proceedings of the IEEE, Vol. 60, November 1972.

25. Greene, W. and Pooch, U. W., "A Review of Classification Schemes for Computer Communication Networks", Computer, Vol. 10, No. 11, November 1977.

26. "Guide to Modems", Computer Decisions, October 1973.

27. "Guide to Multiplexers", Computer Decisions, October 1973.

28. Henning, C. O. C. and Arbogast, G. W., "DCS Channel Packing System", Signal, November/December 1976.

29. Kimbleton, S. R. and Schneider, G. M., "Computer Communication Networks: Approaches, Objectives and Performance Considerations", ACM Computing Surveys, Vol. 7, No. 3, September 1975.

30. Kleinrock, L., "On Communications and Networks", IEEE Transactions on Computers, Vol. C-25, No. 12, December 1976.

31. Kleinrock, L., Queuing Systems Volume II: Computer Applications, John Wiley, 1976.

32. Loomis, B., "Systems Analysis and Design for General Purpose Computing at the Naval Postgraduate School", paper presented in the CS-4200 course at NPS, 12 September 1977.

33. Martin, J., Systems Analysis for Data Transmission, Prentice-Hall, 1972.

34. McCalley, R. D. and Barrett, K. J., "Network Design Allows Diverse Access to Host", Data Communications, February 1978.

35. McNamara, J. E., Technical Aspects of Data Communication, Digital Press, 1978.

36. Newport, C. B. and Ryzlak, J., "Communications Processors", Proceedings of the IEEE, Vol. 60, November 1972.

37. O'Leary D. and Stevens, R. L., "Word Processing Net Speeds FTC's Law Enforcement Task", Data Communications, October 1977.

38. Orr, J. N., "Computer Graphics", Mini-Micro Systems, February 1978.

39. Schneidewind, N. F., "Analysis of Requirements of Principal NPS Batch Users", Internal Memorandum to the Future Computer Planning Committee of NPS, 28 November 1977.

40. Schneidewind, N. F., "Time Sharing Analysis", Internal Memorandum to the Future Computer Planning Committee of NPS, 25 January 1978.

41. Schwartz, M., Computer-Communication Network Design and Analysis, Prentice-Hall, 1977.

42. Shannon, C. E. and Warren, W., The Mathematical Theory of Communication, University of Illinois Press, 1975.

43. Smith, L. B., "The Use of Interactive Graphics to Solve Numerical Problems", Communications of the ACM, Vol. 13, No. 10, October 1970.

44. Spetz, W. B., "Microprocessor Networks", Computer, Vol. 10, No. 7, July 1977.

45. Spilker, J. J., Digital Communications by Satellite, Prentice-Hall, 1977.

46. Stackpole, D., "Programmable Front-End Processors", Electronics Deskbook, McGraw Hill, 1972.

47. Stone, H. S., Introduction to Computer Architecture, Science Research Associates, 1975.

48. Stutzman, B. W., "Data Communication Control Procedures", ACM Computing Surveys, Vol. 4, No. 4, December 1972.

49. Svobodova, L., Computer Performance Measurement and Evaluation Methods: Analysis and Applications, Elsevier, 1976.

50. Sugarman, R., "Computer Concepts", Electronic Engineering Times", February 20, 1978.

51. Users Manual, Computer Center of NPS.

52. Wilcov, R. S., "Analysis and Design of Reliable Computer Networks", IEEE Transactions on Communications, Vol. COM-20, June 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
2.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
3.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
4.	Associate Professor D. A. Stentz, Code 62Sz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Professor N. F. Schneidewind, Code 52Ss Department of Computer Science Naval Postgraduate School Monterey, California 93940	2
6.	Commander Naval Telecommunications Command 4401 Massachusetts Avenue, N.W. Washington, D.C. 20390	1
7.	Lieutenant Commander Ivano A. Rocha Rua Barao do Amazonas, 599-terreo Niteroi, Rio de Janeiro 24000 Brazil	4
8.	Commander John J. Vinson, Code 62Vn Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
9.	Defense Documentation Center Cameron Station Axexandria, Virginia 22314	2