ON SOME DESIGN ASPECTS OF COMPUTER COMMUNICATION NETWORKS

A THESIS
Submitted in fulfilment of the
requirements for the award of the degree
of
DOCTOR OF PHILOSOPHY
in
ELECTRONICS & COMMUNICATION ENGINEERING

By

ARUN KUMAR

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE-247 667 (INDIA)

MAY, 1986

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled, 'ON SOME DESIGN ASPECTS OF COMPUTER COMMUNICATION NETWORKS' in fulfilment of the requirement for the award of Degree of Doctor of Philosophy, submitted in the Department of Electronics and Communication Engineering, University of Roorkee, Roorkee, is an authentic record of my own work carried out during a period January 1983 to May 1986 under the supervision of Dr. Suresh Rai, Reader; Department of Electronics and Communication Engineering.

The matter embodied in this thesis has not been submitted by me for the award of any other degree.

ARUN KUMAR

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

SURESH RAI
READER
Electronics and Communication
Engineering Department;
University of Roorkee,
ROORKEE -247667, INDIA.

MAY 30 ,1986.

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my thesis adviser Dr. Suresh Rai; Reader, Department of Electronics and Communication Engineering, for his invaluable guidance, painstaking supervision and constant and affectionate encouragement during the course of the present work. Without his meticulous scrutiny, it would have been impossible to bring out this thesis in its present form. My association with him has been a very pleasant experience. I once again express my deep sense of gratitude to him.

I am thankful to the Head, Department of Electronics and Communication Engineering for providing necessary facilities to carry out the work.

I am also thankful to Sri Manoj Mishra; an M Tech Student, Department of Electronics and Communication Engineering, for his valuable help towards the development of a computer program.

Special thanks are due to Mr. E.V.Prasad; a QIP research scholar in the department for his useful discussions and also reviewing the manuscript.

The moral support and the acceptance of divided attention by the author's family is recognised with warm appreciation.

Last but not the least, I wish to thank all those who in some way or the other helped make this work a success.

ARUN KUMAR

ROORKEE

CONTENTS

Page

ABSTRACT

NOTATIONS AND ACRONYMS

1. INTRODUCTION 1

1.1 Computer Communication Network - An Idea 1

1.1.1 Definition and General Description 1
1.1.2 CCN Classification 2
1.1.3 Transport Technology 4

1.2 Problem Formulation and Organisation of the Work 7

1.3 Conclusion 11

2. DESIGN ISSUES - AN OVERVIEW 12

2.1 Topology Design 12
2.2 Capacity Assignment 14
2.3 Routing Technique 15
2.4 Reliability Consideration 18

2.4.1 Fault Tree, Reliability Diagram and Graph Model 18
2.4.2 Performance Index 20
2.4.3 Classification of Reliability Evaluation Techniques 21

2.4.3.1 Path and cutset enumeration 22
2.4.3.2 EMD term generation 23
2.4.3.3 One step methods 26

2.4.4 Reliability and Capacity Integration 28

2.5 Branch Availability Computation 29

2.6 Conclusion 35

Page

3. CAPACITY CONSIDERATION IN PERFORMANCE MEASURE OF A CCN 36
  3.1 Capacity Consideration – A Review 36
  3.2 Technique for Performance Index Computation 37
    3.2.1 Assumptions 37
    3.2.2 Procedure Development 38
    3.2.3 Method 39
  3.3 Capacity Assignment 46
  3.4 Conclusion 54
4. ROUTE SEQUENCE ENUMERATION 55
  4.1 Augmented Route Tree Approach 56
  4.2 Algebraic Method 59
    4.2.1 Path Sequence Enumeration 59
    4.2.2 Computing Route Sequence 64
  4.3 Conclusion 72
5. POINT TO POINT CONGESTION COMPUTATION 73
  5.1 Using Reliability Evaluation Method 73
  5.2 Recursive Technique 78
    5.2.1 Graphical Approach 78
      5.2.1.1 Preliminaries 79
      5.2.1.2 Algorithm 85
      5.2.1.3 Discussion 87
      5.2.1.4 Comparison with HMW method 90
    5.2.2 Algebraic Approach 94
      5.2.2.1 Procedure development 94
      5.2.2.2 Algorithm EBRA 99
      5.2.2.3 Discussion 103
  5.3 Conclusion 107

Page

6. CONCLUSION AND FUTURE SCOPE OF THE WORK 109

6.1 Summary and Conclusion 109
6.2 Future Scope of the Work 111

REFERENCES 113

VITA

APPENDIX - 1 COMPUTER PROGRAM

ABSTRACT

Computer communication network (CCN) have evolved very rapidly during the last decade. Recently, as a part of continuing advancement in the networking technology, several impressive developments have been witnessed in the form of broadcast networks, local area networks, and integrated voice and data networks. ARPANET (Advanced Research Project Agency), the first major network, has generated tremendous interest in the field of computer communication network. Today CCN of various kinds are currently in existence or in the process of being set up. These include ALOHA (developed at University of Hawaii), AUTOVON (Automatic Voice Network), commercial network architectures SNA and DNA (developed by IBM and DEC respectively), and public data networks TYMNET and TRANSPAC (developed by Tymshare and PTT of French Government).The number is endless and growing larger literally day by day.

The major design problems in computer communication network include topological optimization for cost, delay and throughput ; capacity assignment to links, routing techniques, network reliability, node to node call blocking and so on. Out of these, the last four problems are studied in detail in chapters 3 thru'5 of this dissertation.

For the reliability analysis, it is generally assumed that requisite amount of information is always transmitted

from source to destination whenever a path is available between them. This implication is neither valid nor economically justifiable in the design of a CCN. In practice, a communication link has a limited capacity because of the cost of a CCN being mainly dependent on it. In chapter 3, we consider a design problem and define the integration of the link capacity with its reliability as a measure of the performance index of a CCN. A method is proposed to solve this problem. It uses the concept of terminal reliability evaluation. The method is simple and advantageous as it applies even for a large network.

The routing issue is the problem of establishing a continuous path, usually incorporating several links in a network, between any pair of source and destination node along which message is to be sent. The objective of any routing procedure is to obtain a minimum time delay while maintaining high throughput. Chapter 4 of the thesis deals with this problem. An algorithm is given to enumerate all the paths between any node acting as source node to another node acting as terminating node, along which message can be routed. The method utilizes the concept of simple algebra. The method is straightforward, easy to computerize and computationally efficient. The concept of path enumeration is, further, extended to obtain the path-loss sequence in various routing plans such as successive-office control (SOC), originating-office control (OOC) and OOC with spill-forward. These strategies are quite practical. The European AUTOVON network employs OOC with spill-forward and CONUS (CONtinental US) uses SOC. As against usual augmented route tree approach for obtaining path-loss sequence, the proposed method is simple and can be implemented easily over a general purpose computer.

In a circuit-switched network one important measure of quality of service is node to node grade of service (NNGOS) which is blocking probability of calls originated at source node and destined for terminal node. NNGOS is the totality of conditions under which a call at source node is not allowed to reach terminal node. This congestion value is mainly dependent on link blocking probabilities, topological layout of network and call control strategy considered above. Chapter 5 of the thesis discusses this problem for its various relevant aspects. Path-loss sequence enumerated in chapter 4 is considered as input and utilizing an efficient reliability evaluation technique the NNGOS in a circuit-switched network is computed. Note the two-step procedure uses the path-loss sequence information which, in turn, depends on routing strategy. The complexity involved in enumerating path-loss sequence may be large in some CCNs. Therefore, another method which does not require the apriori knowledge of path-loss sequence is also proposed. The method is recursive and is derived from link by link call set up procedure. Moreover, it is a particular version of decomposition approach in which a succession of key-stone element(s) is choosen. The method applies to network having unreliable nodes described in terms of input and output switch blocking probabilities. The advantage with this method is that it requires lesser number of steps as compared to existing methods for computing node to node congestion in a circuit-switched network and is also applicable to arbitrary routing plan (OOC with spill) besides being applicable for SOC and OOC strategies. The algorithm results symbolic reliability expression for a general network under minor modifications discussed in the text. Chapter 6 finally concludes the dissertation. It also presents a number of problems as a future scope of the present work.

NOTATIONS AND ACRONYMS

A	2D – array
A_M	modified 2D – array
A_j, B_j	connectivity set of node n_j
a_{ij}, b_{ij}	path of size 1 between two nodes n_i and n_j
a_i	offered load to link i
b	total number of branches
$B_G(K)$	point to point congestion of graph G $(1 - R_G(K))$
[C]	connection matrix
CP_i	capacity of path i
CQ_r	cutset r
$\{CQ(s,t)\}$	set of all cutset
CS_i	capacity of subnetwork formed by 1 – branches in state S_i
C_{max}	capacity of the network with all branches assumed to be 1 – branches
C_i	capacity of link i
C	total capacity (cost) of the network
d_i	indegree of node n_i
D	set of d_i's
E	exclusive operator
EMD	exclusive and mutually disjoint
Δe	error
EEB	end to end blocking

F	Boolean expression of minimal cutset
G	reliability graph
k	total number of nodes
L_j	loss sequence j
m	total number of paths
MST	multiple sourcesto a terminal
n_{avg}	average number of links traversed
n_j	node j
$N_j, \overline{N}_j$	logical success, unsuccess of node n_j
n_L	fictitious loss node
NNGOS	node to node grade of service
$N_i^o, \overline{N}_i^o$	logical success, failure of node i [outgoing]
$N_i', \overline{N}_i'$	logical success, failure of node i [incoming]
OOC	originating office control
PT_{ij}	paths between node n_i and n_j
PL_{ij}	path loss sequence between node n_i and n_j
P_i	path i
P_{ij}	j th term of path i
$\vec{P}$	link reliability vector
PL	path loss polynomial in SOP form
PI	performance index
PS_i	probability of network being in state S_i
p_i, q_i	reliability, unreliability of branch x_i $(p_i + q_i \equiv 1)$

$p^o_{n_i}$, $q^o_{n_i}$	outgoing availability, blocking of node n_i
p'_{n_i}, q'_{n_i}	incoming availability, blocking of node n_i
$\vec{Q}$	link blocking probability vector
R_{st}, $R_G(K)$	terminal pair reliability
S_s	system success function
SF	system success function considering capacity
$\|S\|$	system success determinant
SOP	sum of product
$S_i(.)$	logical signal at node i
SOC	successive office control
S_i	state i of the network
S	set of S_i's
S_{ij}	sum of 1 -and 0 -branches present with P_{ij}
SMT	a source to multiple terminals
T_i	average time delay on link x_i
$V[CQ_r]$	value of cutset CQ_r
w_i	normalized weight of link x_i
x_i	branch i
$X_i,\bar{X}_i$	successful, unsuccessful operation of branch x_i
$(x_{j_1},\ldots,x_{j_n})$	set of branches
z_i	carried load by link x_i

ϕ	null set
γ_{ij}	traffic in message/sec between nodes n_i and n_j
λ_i	average message/sec on link x_i
γ	total number one way message/sec entering the network
μ	avarage message length in bits
λ	total one way link traffic on the average
$\textcircled{X}$	concatenation operation on two paths
X	cartesian product
$\in$	belongs to
$\forall$	for all elements of
$\cup$	Boolean sum or conjunction
$\equiv$	implies identity or definition
$\rightarrow$	replaced by

CHAPTER 1

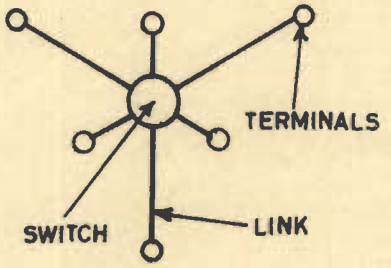
INTRODUCTION

In this chapter, the various aspects of computer communication network (CCN) are discussed. Section 1.1 first describes the general idea of a CCN and explains some of the terms commonly used in it. Lastly, we consider the problem formulation and layout of the dissertation in section 1.2.

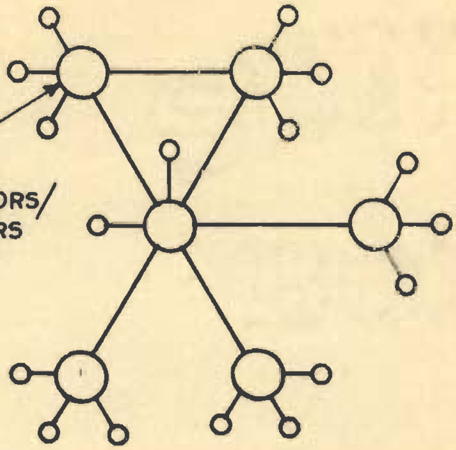
1.1 COMPUTER COMMUNICATION NETWORK-- AN IDEA

1.1.1 Definition and General Description - A computer and communication network is an interconnection of several autonomous computers. Two computers are said to be interconnected if they are capable of exchanging information [1]. By autonomous it is meant here that master/slave relationship is not present [1-3]. Thus, any large computer with remote card readers, printers and terminals is not a computer communication network. Examples of well known CCN include ARPA, AT & T, SITA, TRANSPAC, European AUTOVON, etc [1-5].

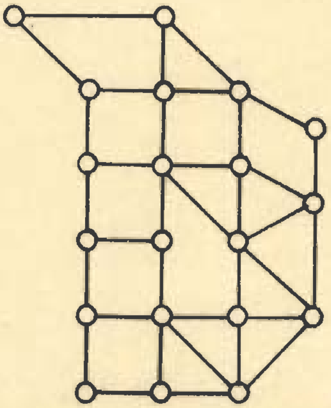
In a CCN there exists a collection of machines called hosts (ARPA net terminology) intended for running user program. Hosts are connected by the communication subnet. The job of subnet is to carry messages from host to host. A subnet consists of two basic components: switching elements and transmission lines. A switching element is a specialized computer, called IMP (interface message processor), and acts as a switch. A host computer, in this context has seperate function than



(a) CENTRALIZED NETWORK

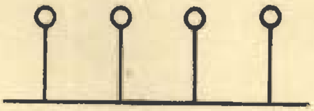


(b) DECENTRALIZED NETWORK

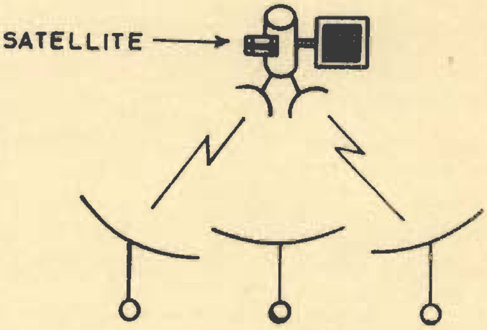


(c) DISTURIBUTED NETWORK

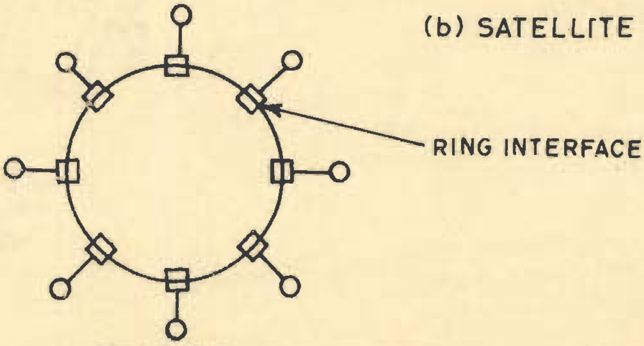
Figure 1.1. Network Topology.



(a) BUS



(b) SATELLITE



(c) RING

Figure 1.2. Communication Sub-Network Using Broadcasting.

that of an IMP. However, some designs permit a single computer to be both IMP and host. The transmission line is also termed as circuit or channel [3]. Physically, it is realized using a pair of wires, cable, laser, satellite etc.

1.1.2 CCN classification– From topology point of view there are following two categories of CCN [3] –

a. point to point

b. broadcast or multipoint

Point to point channel– The network contains switched or leased links or cables; each providing a connection between a pair of nodes. Two nodes with no direct linkage communicate via other nodes.

Networks are further classified: centralized, decentralized and distributed [6]. A centralized network, with all messages flowing inward to some central processing facility, is essentially a star configuration (figure 1.1a). It is the simplest form of network topology and reqires a link to be dedicated between central and terminal nodes. The reliability of the centralized network depends mainly on the controlling node, whose failure causes the failure of the whole network. To increase the reliability, an expensive duplication of the central switching element is required.

A decentralized network of figure 1.1b is an expanded centralized network with some multiplexers or concentrators whose switching power is, to some extent, independent from that of any other node. The distinction between centralized and decentralized network lies in the organisation of the switching function. From graph theory point of view, a decentralized network is described as mixture of star and mesh components, where a mesh is a completely enclosed region [6]. A decentralized network is obviously more reliable than centralized one due to additional computers (nodes) and corresponding connecting links which permit some paths (but not all of them) to be duplicated.

When there are at least two disjoint paths between any pair of nodes, the network is called distributed (figure 1.1c). The distributed network consists of a set of mesh subnetworks in which each node is connected to at least two other nodes. A topology of this type is inherently reliable [7]. Each node, in a distributed network, can switch systematically between links according to routing algorithm choosen to maximise the capacity and minimise the delay of a particular network.

The loop network, a variant to distributed topology, is quite common in local area network. Here, each node is connected to atleast two adjacent nodes and a message is passed from node to node without using routing decisions. A node simply retransmits message to next node until destination is reached. The problem of reliability in loop network is studied in [8].

Broadcast channel – In this design there exists a single communication channel shareable amongst all nodes; since a message broadcasted by a node is to be received at all other nodes. Figure 1.2 shows some of the possibilities for broadcast subnets. In bus topology, at any time there is only one bus master which controls the bus. For this, arbitration mechanism is employed to resolve conflict from two or more nodes requesting the bus. This arbitration mechanism is either centralized or distributed [3].

A satellite channel is another example of broadcast channel. Each node requires an antenna through which it can send and receive messages. A satellite channel provides a broadband capacity but propagation delay is large. Therefore, satellite links may be used to provide high volume data transfer among distant terrestial nodes [4].

The third type of broadcast system is ring topology. In a ring, each bit propagates around the ring on its own, not waiting for the rest of the message to which it belongs. A major issue in the design and analysis of ring nets is the physical length of the bit [3]. This topology is generally preferred for local area network.

1.1.3 Transport Technology – Communication between each pair of computer systems may not necessarily be direct. Transmission of a message from one node to another is, then, made through intermediate nodes. An intermediate node puts an input message on its output link as decided by network routing strategy. Designers distinguish CCNs based on the message transport techniques: circuit switching, message switching and packet switching. Figure 1.3 shows the event timing analysis for these three message switching techniques [9].

In circuit switched network, a message is transmitted by providing a path comprising of links from the originating to the destination node. The path set up is done with a special signalling message. A response to this message from the destination node informs the originating node to proceed with data transfer. Data are then transmitted over all the channels in path with no intermediate store-and-forward delays. The entire fixed-delay, dedicated communications path is allocated to this transmission, until the sender releases the path. The most common example of circuit switching is the telephone system [9,10]. Other examples include TYMNET and European AUTOVON [1].

A message switched network transmits a message between two nodes by moving the message through various transmission links and message buffers. Here, the message is first stored and then retransmitted to the next node along the message path. A message transmission from a node does not start until a buffer at next node en-route is allocated for it. The

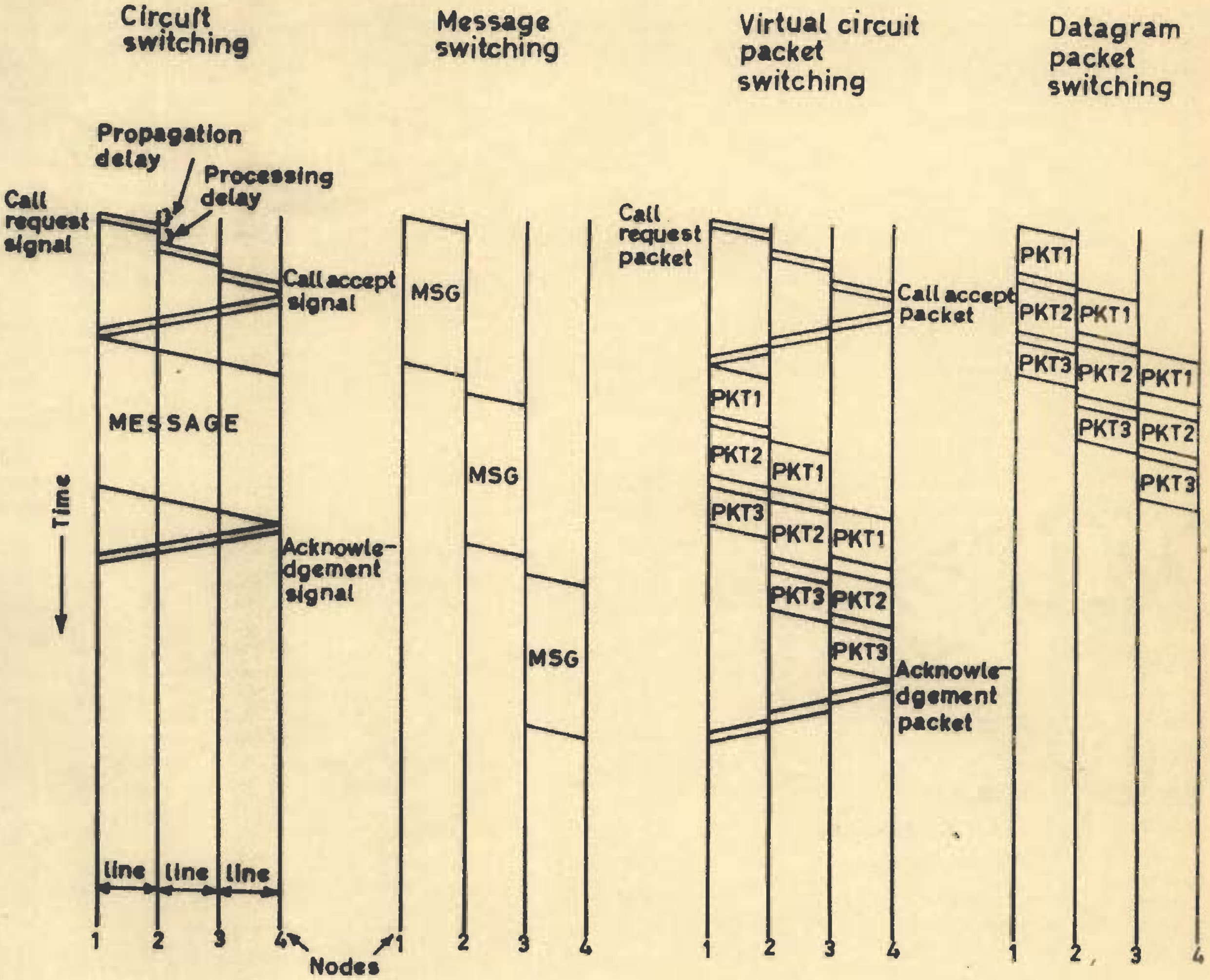


Figure 1.3. Event Time for Various Communication Switching Techniques [9].

path for the message transmission may be fixed or it may be determined dynamically as the message progresses towards destination node [4]. A message switching node is typically a general purpose minicomputer. It performs message coding for entry into the network, carries out message combining and buffering and also routes message to the next destination. Main advantage of message switching over circuit switching lies in its increased link utilization. Nonetheless, message switching suffers from a major drawback which makes it unsuitable for interactive traffic.

Packet switching is either implemented in datagram (DG) or in virtual circuit (VC). DG(VC) utilizes the concept outlined for message (circuit) switching. In either form a messaage is divided into packets of fixed length of bytes to suite the buffer size. This technique is usually considered for packet switched networks (PSN).

Dynamic resource sharing facility of packet switching distinguishes it from circuit switching. In circuit switching the required bandwidth is reserved in advance whereas in packet switching, it acquires and releases depending upon need since circuits are never dedicated. When traffic becomes heavy on a circuit switched network, some calls are blocked. In packet switched network, messages are still accepted; the delivery delay however increases. Whether to use circuit switching or packet switching depends on many factors including mean message size, inter-message arrival and a conversation length [9]. Generally, for transmitting a long, continuous stream of data a circuit switched connection makes good sense. On the other hand, if the data flow is bursty, some form of resource sharing is used to great advantage. Packet switching is then an effective choice [5].

1.2 PROBLEM FORMULATION AND ORGANISATION OF THE WORK

The major design problems in a CCN include

a. topological optimization for cost, delay and throughput
b. capacity assignment to links
c. routing
d. network reliability; node to node call blocking
e. flow control.

Out of these, the problem of capacity assignment to links, routing, network reliability and node to node call blocking are considered in this dissertation.

Chapter 2 reviews problems a through d. Topology design relates to the optimal decision of geographical layout of concentrators or multiplexers and IMPs and their inter-connection amongst themselves or with IMPs such that the traffic and cost constraints are fulfilled. For a known topology the problem of link capacity assignment has been studied in [1,3]. It minimizes the network delay maintaining a fixed total cost. Section 2.3 discusses the problem of routing for its various relevant aspects. A fundamental consideration in the design of a CCN is the reliability and availability of the communication paths between all pair of nodes. These characteristics depend on the topological layout of the communication links, in addition to the reliability and availability of individual computer system and communication facilities. Section 2.4 reviews the various method for it. Lastly, section 2.5 discusses a method to compute the link blocking probability.

The problem of link capacity assignment is studied indirectly. Most methods discuss it considering minimum network delay and a fixed total cost. The reliability of the network, another important criterion, is usually ignored. Alternatively, for the reliability analysis it is assumed that requisite amount of information (capacity) is always available from source to destination whenever a path between them is operative. This implication is neither valid nor economically justifiable in the design of a CCN [11]. In practice, a communication link has limited capacity because the cost of a CCN being mainly dependent on it. Chapter 3 considers this problem and integrates the link capacity with its reliability to define a new measure of performance index (PI) for CCN. A method is proposed to solve it. Here, the concept of terminal reliability is used. The method is simple and applies to large networks too. Section 3.3, then, considers the link capacity assignment problem. Solutions from different link capacity assignment methods are constrained against the reliability figure for the network. A method fulfilling certain minimum/ maximum reliability is finally used for assigning capacity to the link(s).

The routing issue involves establishing a continuous path, usually incorporating several links in a network, between any pair of source and destination node along which message is to be sent. The objective of any routing procedure is to obtain a minimum network delay while maintaining high throughput. Chapter 4 of the thesis deals with this problem. An algorithm, given in section 4.2.1 enumerates all the paths between any node acting as source node to another node acting as terminating node. The method uses the simple concepts of algebra and is easy to computerize. Appendix 1 lists the computer program for it. The path enumeration notion is, then, extended in section 4.2.2 to obtain the path- loss sequence in various routing strategies viz. successive office control (SOC), originating office control (OOC) and OOC with spill - forward (arbitrary routing plan). The European AUTOVON network employs OOC with spill - forward and CONUS uses SOC [12]. As against usual augmented route tree approach for obtaining path -- loss sequence, the proposed method is simple and its implementation over a general purpose computer is easy.

In a circuit switched network an important measure of quality of service is node to node grade of service (NNGOS) which refers the blocking probability of calls originated at source node and destined for terminal node. Thus, the NNGOS defines the totality of conditions under which a call originated at source node is not allowed to reach terminal node. This congestion figure is mainly dependent on link blocking probabilities, topological layout of the network and call control strategy considered in section 2.3. Chapter 5 of the thesis discusses this problem for its various relevant aspects. Section 5.1 describes a procedure in which path-loss sequence enumerated in chapter 4 is considered as an input to any Boolean method of reliability evaluation and obtains NNGOS for a circuit-switched CCN. Note that the path-loss sequence information, in turn, depends on routing strategy. The complexity involved in enumerating path-loss sequence may be large in some CCN and, therefore, another method which does not require the apriori knowledge of such sequences is also described in section 5.2. The proposed method is recursive and is derived from link by link call setup procedure. The advantage with this method is that it requires lesser number of steps as compared to existing methods for computing point to point congestion in a circuit switched network. It is applicable to arbitrary routing plan besides being applicable for SOC and OOC strategies. It also applies to a CCN having unreliable nodes described in terms of input and output switch blocking probabilities. Moreover, it results symbolic reliability expression for general network under minor modifications discussed in section 5.2.2. Finally, chapter 6 concludes the dissertation. It also presents a number of problems as a future scope of the present work.

1.3 CONCLUSION

Chapter 1 has introduced the topic. First, a CCN is defined. The classification of the CCN in accordance with topology and message switching technique is considered next. The problem formulation and layout of the dissertation appear in section 1.2. Chapter 2 reviews some of the design issues raised in it.

CHAPTER 2

DESIGN ISSUES - AN OVERVIEW

The major design issues in a computer communication network include topological optimization for cost, delay and throughput; capacity assignment to links, routing techniques, network reliability, node to node call blocking and so on. Section 2.1, first, overviews some aspects of topological design of a CCN. For a known topology, the problem of link capacity assignment requires apeiori information of the total cost and network traffic. Section 2.2 considers methods for it. Routing pertains to the process of establishing and selecting an appropriate path for message, and directing the message to follow that path through a network. Section 2.3 discusses it for its various relevant aspects. A fundamental considerations in the design of a CCN is the reliability and availability of the communication paths between all pairs of nodes. These characteristics depend on the topological lay out of the communication links and the reliability and availability of individual computer system and communication facilities. Section 2.4 describes this problem of network design. Lastly, section 2.5 discusses a method to compute the link blocking probability. Examples illustrate various relevant points.

2.1 TOPOLOGY DESIGN

The problem of topology design relates to the optimal decision of geographical layout of concentrators (multiplexers) and IMPs and their interconnection amongst themselves or with IMPs fulfilling some measure of traffic expected between the various sources. Note that topological issue, although fundamental to least cost design of networks, is quite complex as it involves several performance criteria [1,4,5]. The time delay criterion imposes the requirement of a specified value of delay. The reliability constraint, on the other hand, is interpreted in several ways. One way insists that alternate path routing be available at all times. Other interpretation, particularly useful to multipoint networks, emphasizes the disconnection of no more than a specified number of terminals or concentrators in case of a link failure. Flow constraint is also important and is usually included. It implies a maximum link time delay in the network. References [1-5,10] discuss a variety of methods to solve the problem of topology design.

For the concentrator and terminal connection, heuristics are generally used because of the complexity of the networks involved. They give results nearly same to the optimal design [1]. Recently, Aggarwal et al [13,14] have discussed optimal topological layout of links which gives maximum terminal pair reliability or network reliability. They assume that nodal positions are known and maximum permissible cost of establishing the links are also specified.

2.2 CAPACITY ASSIGNMENT

The link capacity assignment is made such that network provides a specified node to node grade of service [4]. It assumes apriori knowledge of topology, traffic statistics namely rate of traffic occurrence, average number of messages flowing through the network and cost to be invested. Much of the work on capacity assignment is based on the work by Kleinrock [5]. In its simplest form, the cost is considered linearily proportional to capacity. Holding the overall network cost fixed is, then, equivalent to keeping the total capacity fixed. The linear cost – capacity relationship, however, does not truly represent a real problem where cost depends non-linearily on host parameters; capacity of leased line, length of line etc [1]. Nonetheless, linear cost approach is widely considered as it provides insight into network operation. It enables link capacities to be relatively quickly chosen for complex networks. It also allows comparison between other assignment techniques chosen from a fixed number of capacities, equal or proportional assignment amongst others, and is readily extended to take network routing into account. A detailed description of several capacity assignment criterion along with link and network-reliability is included later in section 3.3.

Examples of nonlinear cost-capacity include those in which incremental cost is proportional to the capacity, the relative cost thus decreasing with capacity or in which the cost of a particular link is based upon a complex array of many variables such as time of day,type of traffic etc. Such problems are handled by computer search routines in which a minimum cost or class of minimum cost networks is chosen after a series of iterative trials. These approaches rely on the use of algorithms that are efficient and coming up with solutions that are optimum or closer to optimum in a reasonable number of trials. They, however, loose insight into the network design process ie, very little is known about the relation between such capacity assignments, message delay and throughput [1,3].

2.3 ROUTING TECHNIQUE

The efficient utilization and sharing of the communications and nodal processing resources of CCN require various type of control, most important of these being routing technique. The objective of any routing procedure is to obtain a good performance while maintaining the high throughput [15]. Good performance usually means low average delay through the network. Routing requires establishing a continuous path incorporating several links in a network between any pair of source and destination nodes along which messages are to be sent. Network routing is implemented either through a message header or by a routing table [9]. In case of message header, all nodes in the message route are included in the header. For routing table approach, all routes are included in routing table. The row and column indices of routing table represent the origination and destination nodes of a route. The table and associated look up table procedures are incorporated in the memory and software of the computer at that node [1]. For the above two approaches trade-off occurs between transmission link cost (message header) and memory cost (routing table) [4,15].

The routing algorithms used in CCN are the variants, in one form or the other, of shortest path algorithm that route messages from source to destination over a path of least cost/ distance [15,16]. The path cost can be assigned using whatever cost functions seem appropriate; the only essential property being that the path cost is computed as the sum of costs of the links comprising that path. Normally, least cost path is regarded as path of minimum delay [1]. Routing techniques tend to vary in implementation and place at which algorithms are run. The routing algorithm may run in centralized fashion or distributed way. Thus, there are various ways of classifying routing procedures such as deterministic, stochastic, fixed, adaptive, centrally controlled or distributed controlled [1,3,15,16]. Petrovic et al [17] have included reliability as a parameter to determine route to avoid congestion.

To allow more than one path to route a call, it is necessary to establish a routing strategy for determining the route(s) to be used and its (their) order of preference. This type of routing is known as alternate routing procedure [5].This procedure chooses its alternate paths either deterministically or at random from among the operating links. The first choice route is called as primary route, and all remaining routes as alternate choice routes [12]. A routing strategy, in this case, is completely described by routing table and call control rule. A number of call control rules is presently used in various communication networks. Some of these are discussed below [12].

Successive Office Control (SOC) Rule--Let the nodes in the n_i to n_j block of routing table be $(n_1, n_2, \ldots, n_r)$. A call reaching at node n_i and destined for n_j is routed to some adjacent node by the use of first available link from the set of links $(n_i - n_1, n_i - n_2, \ldots, n_i - n_r)$, searched in some prescribed order. If all the links in the set are unavailable, the call is lost.

Originating Office Control (OOC) Rule--If node n_j is the originating node for a call, then call is routed to some adjacent node in the manner described under SOC. Else, a call reaching at an intermediate node n_j and destined for node n_r is routed only to the node appearing as first choice node in n_j to n_r block of routing table. In the event the corresponding link is unavailable the originating node is informed of the condition, and next choice link outgoing from the originating office is attempted.

OOC with Spill Forward Rule – To enhance the routing flexiblity at an originating office with a few links connected to it, a modified form of OOC which allows spill forward to some adjacent specified node may be employed. The control of routing is, spill forwarded to an adjacent node which then assumes the role of originating office for routing the call and, hence, permits alternative choices of outgoing links. This routing strategy is also termed as arbitrary routing plan. Specification of such spill forward is part of the information contained in routing table. European AUTOVON network employs OOC with spill and CONUS uses SOC [12].

Chapter 4 considers rule 1 through 3 in detail. A method is also given to compute route sequences. Chapter 5 combines the concept of SOC, OOC and arbitrary routing plans with the reliability of links.

2.4 RELIABILITY CONSIDERATION

2.4.1 Fault Tree, Reliability Diagram and Graph Model – A preliminary step in all reliability evaluation techniques is to model the system by its fault tree or its reliability graph. Fault tree is the translation of a physical system into structured logic diagram, in which certain specified causes lead to one specified top event of interest. Top event(s) is (are) taken from a preliminary hazard analysis. This (these)event(s) is (are) usually undesired system state(s) that can occur as a result of subsystem functional faults [18,19]. A fault tree is constructed using the event

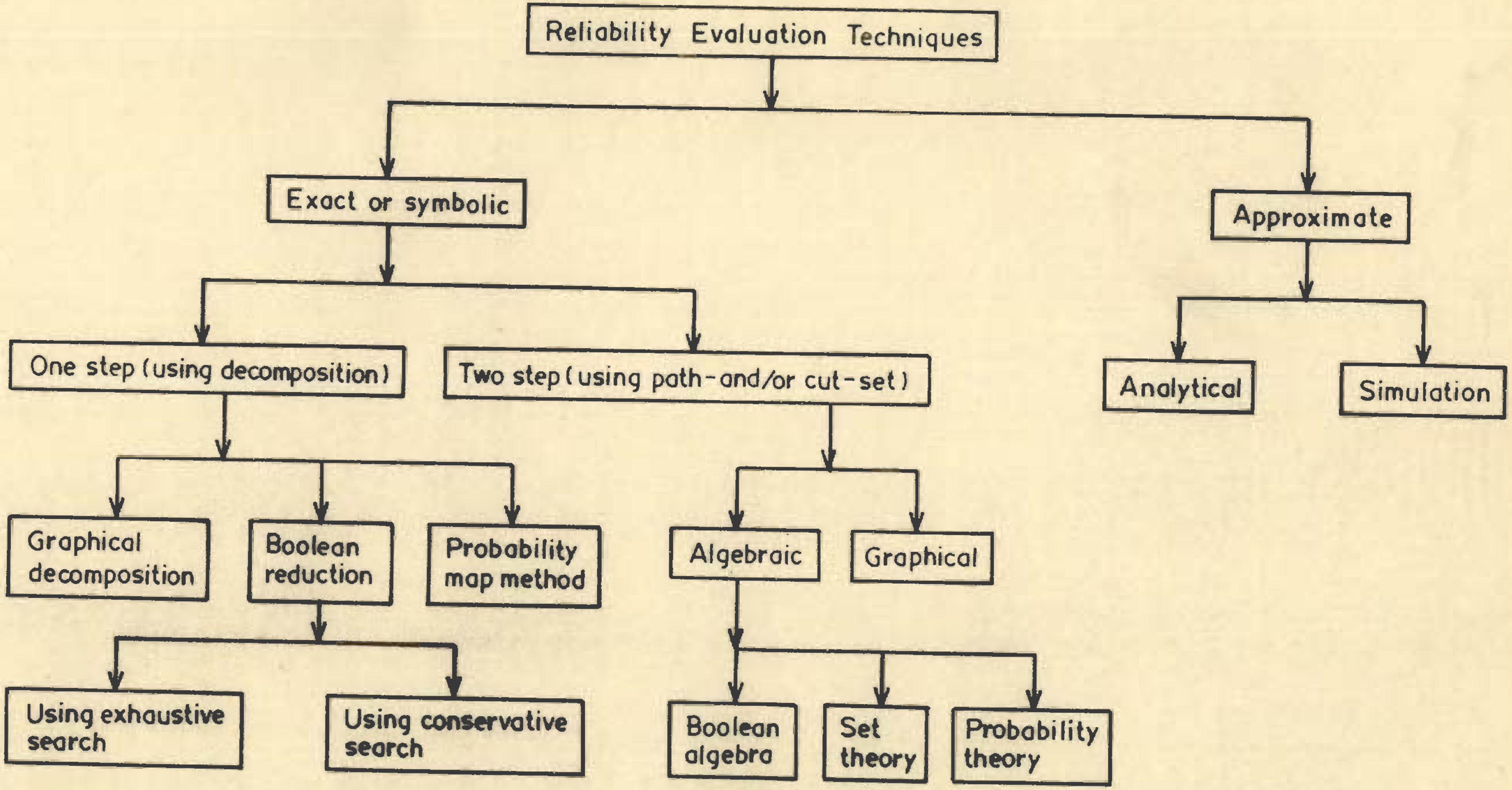


Figure 2.1. Classification Scheme.

and logic symbols [18].

The reliability graph, on the other hand, shows the functional relationship and indicates which element (s) must operate successfully from the system to accomplish its intended function. Note its difference with the system graph that simply depicts the physical relationship of the system elements. In reliability logic diagram, if two components must both function to obtain system success, then blocks corresponding to these components are shown in series, whereas two functionally redundant components are depicted by parallel blocks [18,20].

The reliability analysis of a CCN is based on modeling the network by a graph G in which nodes correspond to computer centres (hosts and IMPs) in the network, and edges to the communication links [21]. It uses the following assumptions--

i. All elements are always operating (no standby or switched redundancy).

ii. The states of all elements are statistically independent.

iii. Each element can be represented as a two terminal device.

iv. The state of each element and of the network is either good (operating) or bad (failed).

v. The network is free from directed cycles and self-loops, because the success or failure of a branch(es) of the directed cycles or self loops does not alter the terminal reliability.

2.4.2 Performance Index – To measure the performance index in a CCN four criteria are usually considered [22,23].They are –

a. a source node s communicating with a terminal node t for all node pairs (s,t) [terminal pair reliability],

b. all nodes communicating with a terminal node t [MST reliability],

c. one node s communicating with all operative nodes [SMT reliability], and

d. all operative nodes communicating [network reliability].

Criterion a, called as terminal pair reliability, is defined as the probability that a specified source node s communicates with another specified terminal node t. Methods for it are reviewed in section 2.4.3. Criterion b, herein after termed MST reliability, is useful for applications in which a central node exercises control over all communication nodes. Various algorithms are available to compute it [22, 24,25]. Criterion c is the source to multiple terminal reliability and is abbreviated as SMT reliability [23,26]. Recently, the development in computer communication networks has aroused interest in computational techniques for more global reliability measures. Criterion d, usually referred to as network (global) reliability [27 – 32], satisfies it. Network reliability is the probability that every vertex is communicating with all other remaining vertices of the CCN. Satyanarayana [23] has shown that under some restrictions to the graph SMT and network reliabilities are identical.

2.4.3 Classification of Reliability Evaluation Techniques- Present day reliability literature describes a multiplicity of methods for evaluating the terminal reliability of a general network. Figure 2.1 depicts a classification scheme for reliability evaluation techniques.

Approximate methods [3,7,18,33-41] are normally used for quick reliability analysis of large systems. Methods of bounds, star-delta and delta-star transformation, simulation method including Monte Corlo technique and network decomposition belong to this category. Some of these methods require apriori knowledge of minimal cutsets. Other global parameters viz. network reliability and SMT reliability have also been discussed using approximate methods [42 - 44].

In contrast to approximate methods symbolic or exact evaluation is important because of various reasons mentioned in [7,45]. Exact methods are divided into two broad categories as

a. two-step, and

b. one-step techniques

Two steps of a require

1. enumerating all or minimal path and cut set, and
2. computing exclusive and mutually disjoint (EMD) terms for the information obtained in step 1.

Subsections 2.4.3.1 and 2.4.3.2 describe steps 1 and 2 in detail.

2.4.3.1 Path and Cut set Enumeration – In a general network with k nodes and b branches there can be approximately 2^{k-2} cutsets and 2^{b-k+2} paths [39], which means that number of paths and minimal cutsets increases exponentially with an increase in network size. Most methods [17,24,25,46 –63] have considered a simplified way of obtaining the minimal path or cut– set starting from either connection matrix, [C], or with incidence matrix. Jamson [25] has given a different approach utilizing the idea of network decomposition for enumerating the simple paths between source and sink node and hence to reduce computational work. Decomposition is performed to partition the network into two subnetworks through a minimal cutset roughly in the centre of the network and it has been shown that only one partition is optimum. Lin et al [64] observed that if $m \geq 10$, the time required to obtain all paths or cutset is only a small fraction of time for reliability evaluation problem. Thus, it becomes imperative to enumerate both paths and cutset and then determine which approach to pursue. Several approaches obtain the minimal cutset from a minimal pathset and vice–versa by a process called inversion utilizing two step application of De Morgan's law [65– 69]. Jamson et al [24] and Jamson [25], have described a computationally efficient method to obtain the path and cutset of a general network. They consider basic minimal paths of the network and find out all the minimal cutset. In a network having 150 simple paths only 11 basic paths are needed to enumerate all minimal cutsets [24,25].

Another simple decision criterion regarding pursuing reliability or unreliability approach is reported in [37,70]. It requires only a knowledge of the structure of the graph. The criterion states: If $b \geq 2k$, the number of paths is larger than the number of cutset and, therefore, the unreliability computation using cutset is simpler.

2.4.3.2 EMD Term Generation - Several methods exist to obtain EMD terms if path and/or cutset are known. These methods generally utilize Boolean algebra, set theoretic approach and probability theory. Bennetts [20] has shown relational correspondence between Boolean and probabilistic approaches. Graph theory is also applied to compute EMD terms. Pruned tree approach reported in [71] is an improvement of tree algorithm described by Nakazawa [72] and uses graph theoretic concepts. Both schemes use Bayes' theorem to decompose the system state into two different disjoint states according to the up- and down-states of a keystone element. Pruned tree approach eliminates irrelevant states of tree approach. Here, the decomposition technique is applied on path matrix rather than system success function. However, this approach suffers from two problems: the number of iterations required grows exponentially with number of literals, and the execution time of algorithm depends heavily upon the selection of keystone element(s).

The probability theory approach obtains the reliability expression using the following result

$$R_G(K) = \sum_{i=1}^{m} P_r(P_i) - \sum_{i<j} P_r(P_iP_j) + \ldots + (-1)^m P_r(P_1P_2\ldots P_m) \qquad (2.1)$$

For m paths, equation (2.1) involves 2^m-1 terms [73,74]. Lin et al [64], using conditional probability, have generated a reliability expression requiring lesser number of terms than 2^m-1. Satyanarayana and Prabhakar [75] have given a topological formula for terminal reliability that generates only noncancelling terms of equation (2.1). This method has additional advantage that p- acyclic subgraphs are generated in sequence without the knowledge of paths or cutset. The timing analysis shows that for a 13 path ARPA network (figure 2.2) algorithm [75] needs 1.2 sec on PDP 11/35, whereas the method of Lin et al [64] requires 9 sec on CDC-6500. Recently, Hariri and Raghavendra [45] have described 'SYREL' which avoids applying the time consuming disjoint process at each iteration of the algorithm by the use of set theory and conditional probability. Reference [45] presents a comparative chart and reports the superiority of SYREL approach.

For a large network the probabilistic domain calculations are more lengthy [20,68,76]. Thus, it is advisable to consider the component reliability parameters to be Boolean variables and treat the whole problem in Boolean domain. This has advantage of allowing the use of powerful Boolean theorems to contain the size of problem [77 - 80]. Moreover, if paths are arranged in the order of their ascending cardinality the overall size of EMD terms is further reduced. Recently, Buzacott [60] has observed a similar effect with cutset and emphasized arranging cuts in accordance with node partition associated with each cut for simplifying unreliability expression.

Using Boolean algebra, path polynomial or system success function is defined as SOP expression of paths P_i's –

$$S_s = P_1 \cup P_2 \cup \ldots \cup P_m \tag{2.2}$$

The next step obtains a mutually exclusive and disjoint system success function as –

$$S_s(\text{disjoint}) = P_1 \cup \bar{P}_1 P_2 \cup \ldots \cup \bar{P}_1 \bar{P}_2 \ldots \bar{P}_{m-1} P_m \tag{2.3a}$$

Equation (2.3a) is although in terms of Boolean variables but has one to one correspondence with the corresponding probability variables [20]. Thus,

$$R_G(K) = P_r(P_1) + P_r(\bar{P}_1 P_2) + \ldots + P_r(\bar{P}_1 \bar{P}_2 \ldots \bar{P}_{m-1} P_m) \tag{2.3b}$$

For m paths, note that there are m terms in (2.3b). However, the time needed to generate each term may be exponential with m. Many methods [20,21,35,41,81 – 91] are given to obtain (2.3a). Abraham's algorithm [41] reduces the amount of computation needed to generate EMD events. Bennetts technique [20] is similar to Abraham's method. Pruned tree approach and Abraham's technique are compared

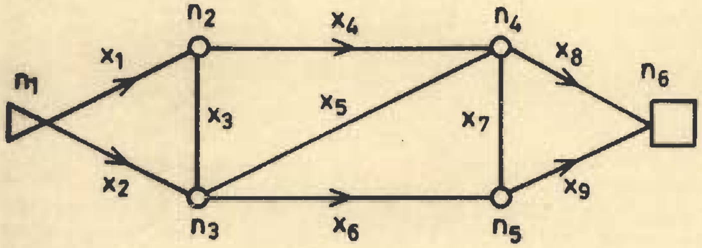


Figure 2.2. Modified ARPA Network.

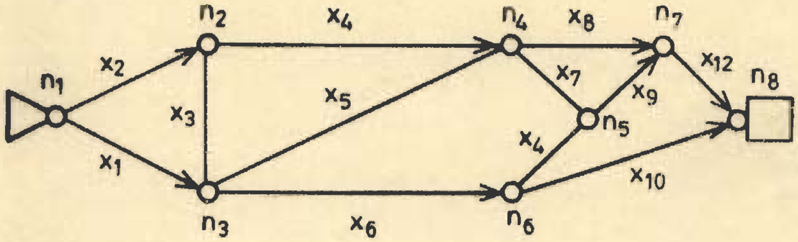


Figure 2.3. 8-Node 12-Link Network.

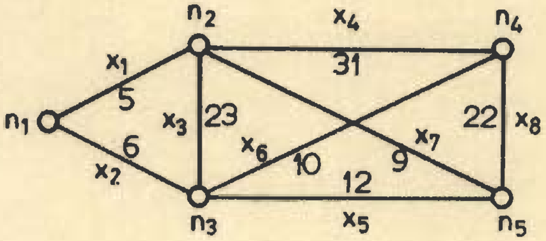


Figure 2.4. A Communication Network.

in [71] considering programs written in the same host language and run on same computer. For a network of figure 2.3 having 24 paths it has been found that pruned tree approach requires 0.33 sec while Abraham's method needs 2.88 sec. Rai and Aggarwal [21] utilize the concept of E – operator to obtain EMD terms. Recently, Schneeweiss [88] has considered Shannon's expansion theorem of Boolean algebra for generating disjoint products (DP). It is a simple algorithm and yields result with ease of documentation [88].

2.4.3.3 One Step Methods – As against two step approaches described earlier, one step approaches offer advantage in the sense that apriori knowledge of path or cutset is not required with them. A probability map method [92] similar to Boolean algebra truth table map has been presented for solving probability combinatorial problems. This technique is simple and straightforward regardless of whether or not the component reliabilities combine in purely series parrallel style. Herley's method [92] is unwieldy for large systems although the final reliability expression contains lesser number of terms. This idea has further been extended by Phibbs and Kuwamoto [93].

Methods [34, 78 – 80, 94,95] make use of Boolean algebra and need a truth table. Systematic procedure for generating a truth table which contains 2^b rows is available in [95].

Network decomposition is another approach that has been studied with several methods. To decompose a network a number of concepts are utilized. They are reported as under.

1. Decompose the network using a keystone element. It, thus implies Bayes' theorem for computing the terminal reliability. The network is decomposed into short- and open-removed subnetworks according to the keystone element. Decomposition can be around either a single keystone element or a number of elements. When more than one edges are considered it is advisable to restrict to conservative policy as against exhaustive one [36]. The rationale under conservative policy is to minimize the number of disjoint events in the analysis and also reduction of parallel links [26,36,96 -102].
2. Separate a node into two or more nodes without changing the network reliability. It simplifies the computation too [103].
3. Use an appropriate cutset to decompose the probabilistic graph of the system [37 -39,104]. A method of this category requires 55 multiplications and 24 sums for numerical computation of reliability as compared to 612 multiplications and 71 sums required in Abraham's method for figure 2.3 [38]. This means a reduction in total computation time by a factor of about 10.
4. Decompose the network into decomposition tree by

finding the triconnected components of the underlying graph [105,106].

Definition 2.1 – Graph G is connected if there do not exist two subgraphs such that the edges of these subgraphs from a partition of edges of G and vertices of these subgraphs form a partition of the vertices of G.

Definition 2.2 – Graph G is biconnected if it is connected and contains no vertices whose removal disconnects the graph.

Definition 2.3 – G is triconnected if it is biconnected and contains no pair of vertices whose removal disconnect the graph.

A graph is splitted into many parts by adding virtual edges and obtains decomposition tree of the network [105]. In order to compute the reliability, the algorithm starts from the leaf of the tree and reaches to the root of the tree using the concept of open – and short – removal of virtual edges. For triconnected components reliability is obtained using Ball's method [22].

2.4.4 Reliability and Capacity Integration – For the reliability analysis it is generally assumed that requisite amount of information is always transmitted from source to terminal whenever a path is available between them. In practice, a communication link has only limited channel capacity which is assigned according to some criterion [1]. Failure probability of links is, thus, implicitely neglected in capacity assignment problem. An effort to integrate link capacity with its probability of being good has been made by various researchers [11, 107 – 111]. Aggarwal's method [11] searches for 2^b states of the network and assigns normalized weight to each state before summation. It, thus, grows exponentially. Trestensky and Bowron [112] also define alternative index for reliability evaluation using the concept of [11]; however they deal with a global parameter rather than terminal reliability.

Chapter 3 considers this aspect of the problem and describes a method for integrating link capacity with the link reliability.

2.5 BRANCH AVAILABILITY COMPUTATION

Computation of link availability or blocking probability is an important step in estimating node to node grade of service for a circuit switched communication network. In what follows, we discuss a simple method described in [12]. It is based on following assumptions–

- call arrival is a poisson process,
- call holding time has a negative exponential distribution,
- link blocking probabilities are statistically independent,
- blocked calls are cleared and do not return,
- call set up time is negligible, and
- the network is in statistical equilibrium.

Using the expressions given in [12], the procedure mentioned here obtains link blocking probability.

Step 0. (Input) Information regarding network topology, routing table, call control rule and traffic matrix.

Step 1. (Initialize) Assume $\vec{Q}_o$, the initial link blocking probability vector and compute $\vec{P}_o$.

Step 2. Generate the path loss sequence (refer chapter 4) for each node pair using the information of step 0.

Step 3. For each link the offered load (a_i), the carried load (z_i) and the link blocking (q_i) are related by

$$a_i = \frac{z_i}{1-q_i} \tag{2.4}$$

Obtain a_i using the most recent iterate value q_i and z_i.Refer example 2.1 for its explanation.

Step 4. Considering Erlang's loss formula determine the link blocking probability q_i as

$$q_i = \frac{a_i^{C_i} / C_i}{\sum_{r=o}^{C_i} (a_i^r / r!)} \tag{2.5}$$

Step 5. Repeat steps 3 and 4 until the differences between all corresponding enteries in $\vec{Q}_{n+1}$ and $\vec{Q}_n$ are smaller than some prescribed error (Δe).

Example 2.1 – In order to compute the link blocking or branch availability, consider figure 2.4 along with its routing information given in table 2.1 [12]. Integers on links indicate number of channels. The busy hour traffic is listed in table 2.2.

Table 2.1 A Routing Table for figure 2.4 [12]

FROM ↓ TO →	n_1	n_2	n_3	n_4	n_5
n_1	–	n_2, n_3	n_3, n_2	n_3, n_2	n_3^*, n_2
n_2	n_1	–	n_3	n_4, n_5	n_5, n_4
n_3	n_1, n_2	n_2	–	n_4, n_2	n_5, n_4, n_2
n_4	n_3, n_2	n_2, n_3	n_3, n_2	–	n_5
n_5	n_2, n_3	n_2, n_4	n_3, n_2, n_4^*	n_4	–

Table 2.2 Busy Hour Traffic Matrix for figure 2.4 in CCS ($1\text{CCS} = \frac{1}{36}$ Erlang)

FROM ↓ TO →	n_1	n_2	n_3	n_4	n_5
n_1	–	30	40	10	20
n_2	40	–	375	350	200
n_3	50	200	–	150	250
n_4	20	300	125	–	325
n_5	20	60	200	60	–

Table 2.3 Reliability Expression for Nodes $n_5 - n_3$

P_i	Path	EMD term	Remark
P_1	x_5	X_5	$P_r(P_1$ used)
P_2	x_3x_7	$X_3X_7\bar{X}_5$	$P_r(P_2$ used)
P_3	x_6x_8	$X_6X_8\bar{X}_5\overline{X_3X_7}$	$P_r(P_3$ used)
P_4	$x_3x_4x_8$	$X_3X_4X_8\bar{X}_5\bar{X}_6\bar{X}_7$	$P_r(P_4$ used)

Table 2.4 Traffic Carried by Links for n_5 to n_3 Call [12]

Path	Load carried by links in CCS
$P_1(x_5)$	200 x 0.9 = 180
$P_2(x_3,x_7)$	200 x 0.081=16.2
$P_3(x_6,x_8)$	200 x 0.0154=3.08
$P_4(x_3,x_4,x_8)$	200 x 0.0007=0.14

Table 2.5 Traffic Carried by Link x_3[12]

Node to Node	Carried by Link x_3
$n_2 - n_3$	337.5
$n_4 - n_3$	10.12
$n_5 - n_3$	16.34
$n_1 - n_3$	3.24
$n_3 - n_2$	180.0
$n_1 - n_2$	2.43
$n_3 - n_4$	12.15
$n_3 - n_5$	3.88
$n_3 - n_1$	4.05
$n_1 - n_5$	0.277

A path loss sequence for each node pair under a specific call control rule may be determined using the method described later in chapter 4. As an example, the n_5 to n_3 call under OOC with spill scheme results into following path loss sequence

$P_1(x_5)$, $P_2(x_7x_3)$, $P_3(x_8x_6)$, $P_4(x_8x_4x_3)$, $L_1(x_8)$, $L_2(1)$

In order to calculate link blocking vector $\vec{Q}$, consider the initialization value $\vec{Q}_o$ as

$\vec{Q}_o = [\ 0.1,\ 0.1,\ \ldots,\ 0.1]$

which obtains the initial link reliability vector

$\vec{P}_o = [0.9,\ 0.9,\ \ldots,\ 0.9]$

Using the system reliability analysis technique [21] the EMD terms given in table 2.3 are obtained. Thus we have -

$P_r(P_1$ used) = 0.9

$P_r(P_2$ used) = 0.0810

$P_r(P_3$ used) = 0.0154

$P_r(P_4$ used) = 0.0007

Once these probabilities are computed and traffic for n_5 to n_3 call is also known, the traffic carried by each link is determined easily. Table 2.4 lists the link traffics. From table 2.4, it is obvious that total load carried by link x_3 is 16.34 for n_5 to n_3 calls. Similar calculations of loads are made for all other node pairs. Table 2.5 lists such data obtained for x_3 link. Summing up the last column, total traffic carried by link x_3 is 570 CCS (= 15.83 erlangs). Using link unavailability of 0.1 (assumed) load offered to

x_3 link is

$a_3 = 15.83/(1 - 0.1) = 17.59$ erlangs

From figure 2.4 it is observed that capacity of link x_3 is 23. Considering $a_3 = 17.59$ and $C_3 = 23$ the computed link blocking of x_3 becomes

$q_3 = 0.0377$

Blocking probabilities of all other links are determined similarly. The resultant vector is called $\vec{Q_1}$. This completes the first iteration. Assuming this vector as input obtain a another vector $\vec{Q_2}$. The iteration is carried out until a negligible error Δe is detected between two consecutive $\vec{Q}$ vectors. Here, with an error criterion of 0.001, the procedure terminates after six iterations and results obtained are given in table 2.6.

Table 2.6 Link Blocking Probabilities [12]

Link	Blocking probability
x_1	0.1094
x_2	0.1039
x_3	0.0784
x_4	0.0138
x_5	0.2434
x_6	0.2556
x_7	0.2530
x_8	0.0151

2.6 CONCLUSION

This chapter reviews some design aspects of a CCN. We have typically considered topology, capacity assignment, routing, reliability etc to highlight the issue. Some of these are carried out in detail in the chapters to follow.

CHAPTER 3

CAPACITY CONSIDERATION IN PERFORMANCE MEASURE OF A CCN

This chapter discusses the concept of integrating link capacity with its reliability of success in section 3.1. It also reviews methods described in the literature for combining the two parameters. A technique proposed in section 3.2 solves this problem. It utilizes the concept of terminal reliability evaluation. The method applies for large networks too. Lastly, section 3.3 considers the capacity assignment problem in a network using the procedure discussed in section 3.2. Examples illustrate these points.

3.1 CAPACITY CONSIDERATION – A REVIEW

System reliability analysis often assumes that the system modeled as a probabilistic linear graph is functioning if there exists a path from source node to sink node. In this way, it is equivalent to a problem of connectivity analysis and primarily concerns with the enumeration of paths or cuts in the graph. But in many physical systems, a number corresponding to flow/ capacity is associated with branch. Consequently, reliability of a network is not necessarily characterized by only connectivity [109]. Moreover, the reliability analysis implies that the requisite amount of information, which is known in advance, is always transmitted from source(s) to terminal(t) whenever a path is available between them. It means that every link is capable of handling the required s-t flow. This implication is neither valid nor economically justifiable in the design of a CCN [11].

Methods [11,107 – 110] are suggested to consider the link capacity, besides its reliability, as an important constraint for the successfull operation of flow network. Here, the definition of terminal reliability is modified as " the probability of successfully transmitting the required amount of information from s to t". Lee [109] introduces a failure state as a state in which network carries only a part of requisite traffic even if a path is available. His method hinges on the lexicography property and hence suitable for directed graphs only. Aggarwal [11] defines performance index (PI) where channel capacity of a link is integrated with its reliability of success. To obtain PI, his method searches for 2^b states of truth table. It, thus, grows exponentially and is not useful even for a moderately large network.

3.2 TECHNIQUE FOR PERFORMANCE INDEX COMPUTATION

3.2.1 Assumptions –

a) The capacity of each link is known and is finite.

b) Nodes are perfect. If the nodes are not perfect, the method can be modified using [113].

c) Each link (and also the network) has only two states good or failed.

d) No information or flow can pass through a failed link.

e) The network is free from self loops and directed cycles.

f) The states of all elements are statistically independent.

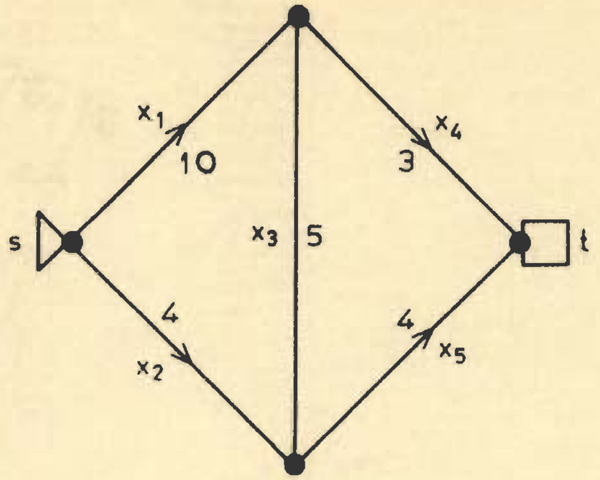


Figure 3.1. A Bridge Network.

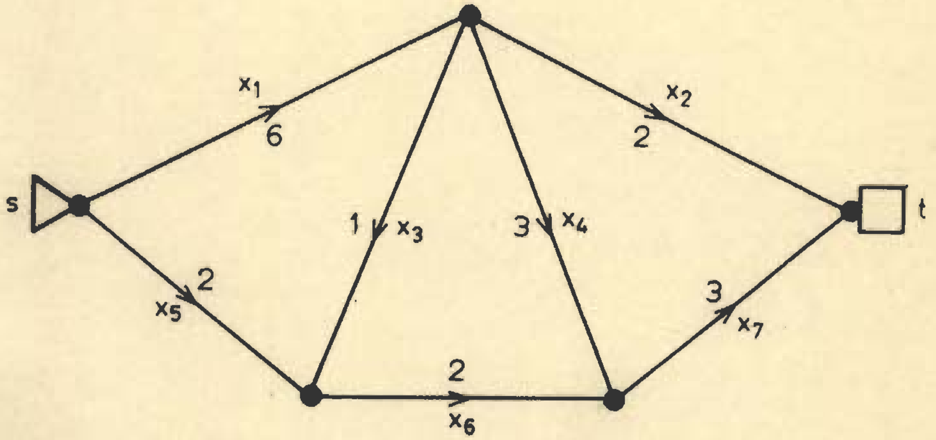


Figure 3.2. A Flow Network.

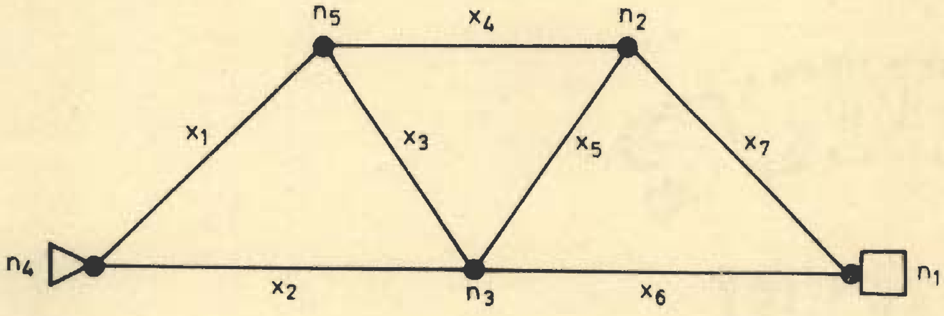


Figure 3.3. 5-Node 7-Link Network.

3.2.2 Procedure Development –

Definition 3.1 – A branch is said to be 1(0) branch if it is good (failed). A 1(0) branch is denoted as X_j ($\bar{X}_j$).

Definition 3.2 – The weighted reliability measure called the performance index [11] is defined as

$$PI = \sum_i w_i \, PS_i$$

where PS_i represents the probability of network being in state S_i($S_i \in S$; S being the set of all success states). In the conventional sense, terminal reliability is

$$R_{st} = \sum_{S_i} PS_i$$

Definition 3.3 – Let the capacity of the subnetwork formed by the 1 – branches in state S_i be CS_i. The normalized weight w_i is, then, defined as [11].

$$w_i = CS_i / C_{max}$$

where C_{max} is the capacity of network with all branches assumed to be '1 – branch'.

Definition 3.4 – The capacity of a path P_i is obtained from the capacities of 1 – branches contained in it and is

$$CP_i = \min_{x_j \in \{P_i\}} [C_j]$$

Note that P_i is one of the success states for the network provided $CP_i \leq C_{max}$. Thus $S_i = P_i$ and hence CS_i is obtainable using definition 3.4.

Definition 3.5 – The value of a cutset CQ_r, indicated by $V[CQ_r]$, is the sum of link capacities of all links in CQ_r. Consider a set $\{CQ\ (s,t)\}$ of all cutsets with separate vertices s and t. Note that for any cutset $CQ_r \in \{CQ\ (s,t)\}$, $V[CQ_r]$ can not be smaller than the maximum flow C_{max}[48]. Thus, C_{max} is obtained easily using the max– flow min– cut theorem as –

$$C_{max} = \min \{V[CQ_r]\} \ ; \ CQ_r \in \{CQ(s,t)\}$$

Example 3.1 – The cutset which separates s and t in figure 3.1 is $\{(x_1,x_2), (x_1,x_3,x_5), (x_2,x_3,x_4), (x_4,x_5)\}$. Hence,

$$\begin{aligned} C_{max} &= \min [V(x_1,x_2), V(x_1,x_3,x_5), V(x_2,x_3,x_4), V(x_4,x_5)] \\ &= \min [14,19,12,7] \\ &= 7 \end{aligned}$$

Example 3.2 – In order to illustrate above definitions, consider the bridge network of figure 3.1. The normalized weight w_i for a path x_1x_4 is 3/7, where

$$\begin{aligned} CP_1 &= \min (C_1,C_4) \\ &= 3 \end{aligned}$$

Here x_1 and x_4 are representing the 1 –branches for the network. Thus, PI for path $x_1\ x_4$ is $\frac{3}{7}\, p_1p_4$.

3.2.3 Method – The proposed method utilizes following three major steps.

Step 1. Enumeration of all minimal paths between specified source and terminal node.

Step 2. Computation of capacity for each path of step 1.

Obtain C_{max} too.

Step 3a. Manipulation of terms of Step 2 into disjoint products ignoring the effect of capacities.

b. Inclusion of capacity coefficient(s) into step 3a and obtaining performance index (PI) of the network.

Note that steps 1 and 3a are usually employed to compute terminal reliability. These may, therefore, be adopted as such from the reliability literature. For convenience, we are using the technique of E – operator [21]. Step 2 is covered by definitions 3.4 and 3.5. Max –flow min –cut theorem obtains C_{max} which, in turn, is used to normalise CP_i. We have, thus, to concentrate on step 3b. The discussion that follows highlights this aspect of the procedure.

Algorithm A – It obtains terminal reliability using E – operator [21].

A1. Write down the system success function as

$$SF = \sum_{i=1}^{m} \left(\frac{CP_i}{C_{max}} \right) P_i \qquad (3.1)$$

where P_i represents minimal path i of the network and is obtained using method given later in section 4.2.In equation (3.1) each path is multiplied with a capacity term which is equivalent to traffic carried through it. Finally a normalization factor of C_{max} is used to normalize all such terms.

A2. Ignoring the effect of capacities in (3.1), obtain a disjoint sum of minimal path such that

$P_i \wedge P_j \equiv 0$

For this purpose any standard Boolean technique can be utilized. Here, we have used E – operator [21]. For each term P_i, $1 < i \leq m$, F_i is defined to be the union of all predecessor terms $P_1, P_2, \ldots, P_{i-1}$ in which any literal that is present in both P_i and any of the predecessor terms is deleted from those predecessor terms. Thus,

$$F_i = P_1 + P_2 + \ldots + P_{i-1} \Big|_{\text{Each term of } P_i \rightarrow 1}$$

F_i is, then, simplified using standard Boolean reduction identities [77]. Using E – operator, the symbolic expression for terminal reliability is

$$R_{st} = [P_1 + \sum_i P_i \, E(F_i)] \Big|_{X_i(\bar{X}_i) \rightarrow p_i\ (q_i)} \qquad (3.2)$$

A3. Compute performance index PI using the steps of Algorithm B.

Algorithm B – It mainly determines the coefficient of a term of terminal reliability obtained in equation (3.2). The various substeps are described as under.

B1. Define jth term of path P_i in (3.2) as P_{ij} and obtain its S_{ij}, which is the sum of 1 – and 0 – branches present with P_{ij}.

B2a. [Initialize] $i \leftarrow 1$ and $PI \leftarrow 0$.

b. $j \leftarrow 1$

c. If $S_{ij} = b$; CP_i is not modified and go to step B9a.

B3. Assume $Y_1, Y_2, \ldots, Y_r$ be the absent branches in P_{ij}. Use exhaustive policy to obtain 2^r disjoint events $f_e\,(Y_1, \ldots, Y_r)$'s. Thus, 2^r terms are generated out of P_{ij}. Denote a $P_{ij} f_e(Y_1, \ldots, Y_r)$ term by $P_{ij}(e)$; $1 \leq e \leq 2^r$.

B4a. $e \leftarrow 1$

b. Check whether 1– branch terms of $P_{ij}(e)$ contains more than one path . It can easily be verified by using Boolean identity.

$A + AB \equiv A$

where A represents a path in the path list and AB is the 1 – branch terms of $P_{ij}(e)$.

c. If $P_{ij}(e)$ contains only one path, compute $CP_i(e)$ using definition 3.4 as

$$CP_i(e) = \min_{x_h \in P_{ij}(e)} (C_h)$$

and go to step B7. Otherwise, proceed as follow.

B5. Use algorithm given in [65] to complement the paths contained in $P_{ij}(e)$.

B6a. Define τ_z as the sum of edge capacities of z–th term of B5.

b. $CP_i(e)$ associated with $P_{ij}(e)$ is given as

$$CP_i(e) = \min \; \forall_z(\tau_z)$$

B7. Compute the following

$$PI \leftarrow PI + \left(\frac{CP_i(e)}{C_{max}} \right) P_{ij}(e) \Big|_{X_h(\bar{X}_h) \to p_h(q_h)}$$

B8a. $e \leftarrow e+1$ and go to step B4b if $e < 2^r+1$. A simplification rule saves (2^r-1) terms in the expression for PI. It states; For e, $1 \leq e \leq 2^r$ if $CP_i(e) = CP_i$ there is no need to consider repeatedly 2^r terms generated out of P_{ij}. In that event it suffices to include only $(\frac{CP_i}{C_{max}})P_{ij}$ with PI.

b. Go to step B9b; otherwise.

B9a. Evaluate the following

$$PI \leftarrow PI + \left(\frac{CP_i}{C_{max}}\right) P_{ij}\Big|_{X_h(\bar{X}_h) \rightarrow p_h(q_h)}$$

b. $j \leftarrow j+1$ and go to step B2c.

B10. $i \leftarrow i+1$ and check whether $i<(m+1)$. If not go to B2b.

B11. [Terminate].

Example 3.3 – For the bridge network of figure 3.1 the 4 minimal paths and their respective capacities are

$P_1 = x_1 x_4$ $\qquad CP_1 = 3$

$P_2 = x_2 x_5$ $\qquad CP_2 = 4$

$P_3 = x_1 x_3 x_5$ $\qquad CP_3 = 4$

$P_4 = x_2 x_3 x_4$ $\qquad CP_4 = 3$

The system success function SF as defined in equation (3.1) is given as under

$$SF = \frac{3}{7}P_1 + \frac{4}{7}P_2 + \frac{4}{7}P_3 + \frac{3}{7}P_4$$

where $C_{max} = 7$ is obtained in example 3.1. The symbolic expression for terminal reliability obtained using E – operator is

$$R_{st} = [\, X_1X_4 + X_2X_5\,(\bar{X}_1+X_1\bar{X}_4) + X_1X_3X_5(\bar{X}_2\bar{X}_4) + X_2X_3X_4(\bar{X}_1\bar{X}_5)]\Big|_{X_i(\bar{X}_i) \rightarrow p_i(q_i)}$$

Note that terms like $X_1X_3X_5(\bar{X}_2\bar{X}_4)$ and $X_2X_3X_4(\bar{X}_1\bar{X}_5)$ contain total number of five 1 –and 0 – branches in each. Their coefficients are, therefore, not modified and remain same as $\frac{4}{7}$ and $\frac{3}{7}$ respectively. For remaining terms the expansion around absent branches are given in table 3.1. Using algorithm B, it is verified that the coefficient of $X_2X_5(\bar{X}_1+X_1\bar{X}_4)$ is also not modified. However, the capacity for various terms of X_1X_4 are modified and is shown in table 3.2. Thus the expression for performance index PI is given as

$$\begin{aligned} PI = {} & p_1q_2p_3p_4p_5 + p_1p_2p_3p_4p_5 + p_1p_2q_3p_4p_5 + \frac{3}{7}\,[p_1q_2q_3p_4q_5 \\ & + p_1q_2q_3p_4p_5 + p_1q_2p_3p_4q_5 + p_1p_2q_3p_4q_5 + p_1p_2p_3p_4q_5 \\ & + q_1p_2p_3p_4q_5\,] + \frac{4}{7}\,[q_1p_2p_5 + p_1p_2q_4p_5 + p_1q_2p_3q_4p_5] \end{aligned}$$

Example 3.4 – Consider the problem of a flow network considered by Lee [109; p 25; figure 1]. The network is redrawn in figure 3.2 to improve the readability of the text. It is assumed that [109] the network is good i.f.f. it transmits at least 3 units of flow from the input node to the output node.

Four paths for figure 3.2 are $x_1x_4x_7$, $x_5x_6x_7$, x_1x_2 and $x_1x_3x_6x_7$. The symbolic expression for network reliability is obtained using E – operator [21]. It is given as

Table 3.1 Use of Exhaustive policy for absent branches

Term	Absent branch(es)	Exhaustive expansion
X_1X_4	X_2,X_3,X_5	$\bar{X}_2\bar{X}_3\bar{X}_5$, $\bar{X}_2\bar{X}_3X_5$, $\bar{X}_2X_3\bar{X}_5$ $\bar{X}_2X_3X_5$, $X_2\bar{X}_3\bar{X}_5$, $X_2\bar{X}_3X_5$ $X_2X_3\bar{X}_5$, $X_2X_3X_5$
$X_2X_5\bar{X}_1$	X_3,X_4	$\bar{X}_3\bar{X}_4$, $X_3\bar{X}_4$, $\bar{X}_3X_4$, X_3X_4
$X_2X_5X_1\bar{X}_4$	X_3	$\bar{X}_3$, X_3

Table 3.2 Modification in Coefficient of X_1X_4

Term	Coefficient	Term	Coefficient
$X_1X_4\bar{X}_2\bar{X}_3\bar{X}_5$	3/7	$X_1X_4X_2\bar{X}_3\bar{X}_5$	3/7
$X_1X_4\bar{X}_2\bar{X}_3X_5$	3/7	$X_1X_4X_2\bar{X}_3X_5$	1
$X_1X_4\bar{X}_2X_3\bar{X}_5$	3/7	$X_1X_4X_2X_3\bar{X}_5$	3/7
$X_1X_4\bar{X}_2X_3X_5$	1	$X_1X_4X_2X_3X_5$	1

$$R_{st} = [X_1X_4X_7 + X_5X_6X_7(\bar{X}_1 + X_1\bar{X}_4) + X_1X_2(\bar{X}_7 + X_7\bar{X}_4\bar{X}_5 + X_7X_5\bar{X}_4\bar{X}_6) + X_1\bar{X}_2X_3\bar{X}_4\bar{X}_5X_6X_7]\Big|_{X_i(\bar{X}_i) \rightarrow p_i(q_i)} \qquad (3.3)$$

Using steps of algorithm B, each term of equation (3.3) is tested for a flow of 3 or more units. It is found that only following 4 terms satisfy this criterion

$X_1X_4X_7$, $X_1X_2X_3\bar{X}_4\bar{X}_5X_6X_7$, $X_1X_2\bar{X}_3\bar{X}_4X_5X_6X_7$, $X_1X_2X_3\bar{X}_4X_5X_6X_7$

and, hence, they taken together form an expression for reliability of the flow network. The result matches with that given in [109].

Thus, the technique given above is general and applies for computing i) performance index of a CCN and ii) reliability of a flow network. Moreover, the method is not restrictive for large networks, an important drawback of [11].

3.3 CAPACITY ASSIGNMENT

One of the major design problems in a CCN is the allocation of link capacity in bps. In this section, we study this problem and augment it with the concept of network reliability. To start with, assume the information regarding following is known.

1. topology of the network,
2. traffic matrix $[\gamma_{ij}]$
3. route matrix and
4. half and full duplex working of links,
5. total cost to be invested.

Consider that the choice of a link capacity C_i requires a cost b_iC_i; b_i representing the constant of proportionality. This assumption is in no way restrictive to the method suggested here. Now, choose C_i to minimize.

$$T^{(h)} = \left[\sum_{i=1}^{b} \frac{\lambda_i}{\gamma} (T_i)^h \right]^{\frac{1}{h}} \qquad (3.4)$$

where $T^{(h)}$ represents delay time and is discussed below. Various criteria exist for choosing the link capacity C_i[1]. Table 3.3 lists four of them and provides relations for C_i. It also gives average network delay T_{avg} under different situations described below.

Case (i). For $h = 1$ in (3.4) we have the average time delay criterion. It is also named as the square root assignment.

Case (ii). For $h \rightarrow \infty$ in (3.4) we have Chebyshev or min-max criterion. Under this rule maximum time delay on any link is minimized, keeping total cost fixed. The time delay on each link turns out to be equal.

From table 3.3, it is obvious that square root assignment rule gives minimum average time delay and is disadvantageous to the light link users. Note that the delay on light link traffic is more as compared to that on high link traffic. The min-max criterion of case (ii) eliminates time delay penalties incurred by light users altogether, with link time delays equalized throughout the network. This, however slightly increases the total average network delay. Thus, min-max criterion comes out to be user oriented approach to

Table 3.3 Link capacity and network delay under different criterion.

Sr. No.	Criterion name	C_i	T_{avg}	Remark
1.	Square root	$\frac{\lambda_i}{\mu}[1+ \frac{1}{r_1\sqrt{\lambda_i}}]$	$\frac{n_{avg}r_1}{\lambda} \sum_i \sqrt{\lambda_i}$	$T_i = \frac{1}{\mu C_i - \lambda_i}$
2.	Min-max	$\frac{\lambda_i}{\mu}[1+ \frac{1}{r_2\lambda_i}]$	$n_{avg}r_2$	$n_{avg} = \lambda/\gamma$ $\lambda = \Sigma_i \lambda_i$
3.	Equal assignment	$\frac{C}{b}$	$\frac{n_{avg}}{\lambda} \sum_i \frac{\lambda_i}{(\frac{\mu C}{b}) - \lambda_i}$	$\gamma = \Sigma_i \Sigma_j \gamma_{ij}/2$ $r_o = \mu C - \lambda$
4.	Proportional assignment	$C \frac{\lambda_i}{\lambda}$	$n_{avg}\ r_2$	$r_1 = \Sigma_i \sqrt{\lambda_i}/r_o$ $r_2 = \frac{b}{r_o}$

Table 3.4 Traffic matrix for figure 3.3

FROM ↓ TO →	n_1	n_2	n_3	n_4	n_5
n_1	–	9.34	0.935	2.940	0.610
n_2	9.34	–	0.820	2.40	0.628
n_3	0.935	0.820	–	0.608	0.131
n_4	2.94	2.40	0.608	–	0.753
n_5	0.610	0.628	0.131	0.735	–

design a network [1].

Case (iii). An equal assignment strategy uses a simple rule in which the total cost/ capacity C is equally divided amongst all the links, independent of traffic on it (refer table 3.3).

Case (iv). A proportional assignment strategy obtains C_i in proportional to the traffic deman λ_i.

Out of four cases discussed above, the question is how to select a particular criterion for assigning C_i. Using equation (3.4), a choice is generally made in the favour of one which offers minimum average network delay. Note that no consideration is being taken for network reliability because a few methods [11, 107 – 110] incorporate the link capacity and – reliability together. The problem gets simplified by utilizing the technique outlined in section 3.2. Thus, augmenting network reliability besides average network delay for judiciously selecting a method to obtain C_i considers both the link probability of success and link delay.

Algorithm C – The steps of the algorithm are as follows.

Step C1. Compute C_i under different criterion.

Step C2. Use algorithm A and B of section 3.2 to obtain PI.

Select a set of C_i's that offers better value of PI.

Algorithm C is illustrated in example 3.5.

Example 3.5 – Consider a network shown in figure 3.3 [1]. Each link works both ways simultaneous (full – duplex) and is represented by an undirected branch. The link capacity in either direction is, thus, same. Assume the symmetrical traffic matrix as given in table 3.4 for simplicity of computation. Assymetrical traffic flows can also be handled in a similar manner. In order to compute traffic flow on each link assume.

1. traffic queueing up for transmission over any link is statistically independent of traffic operating any where else in the network and,
2. routing of messages takes the shortest geographical route as mentioned in table 3.5.

Table 3.5 Shortest route traffic for figure 3.3

TO→ FROM ↓	n_1	n_2
n_4	n_3	n_5
n_5	n_2	—

Using the informations given in table 3.4 and 3.5 obtain average message rate λ_i's and μC_i (table 3.6). The value of λ is 25.12. It assumes that the total cost of network is fixed (192 message/sec) and $\frac{1}{\mu}$ message length in bits is also constant. Tables 3.7 and 3.8 show the time delay on each link and average network time delay respectively. It is obvious from these tables that square – root assignment strategy offers minimum average time delay at the cost of increase in delay of light traffic link (s) such as x_3 and x5

Table 3.6 Link capacity (μC_i) assignment for various criterion.

Link	λ_i	Criterion (gives one-way values)			
		Square root	Proportional	Equal	Min-Max
x_1	3.15	28	24	27.4	27
x_2	3.55	30	27.5	27.4	27.4
x_3	0.13	5	1	27.4	24
x_4	3.64	30	28	27.4	27.5
x_5	0.82	13.5	6.3	27.4	24.6
x_6	3.88	31.5	30	27.4	27.7
x_7	9.95	54	76.5	27.4	33.8

Table 3.7 Average time delay (T_i) in m.sec on each link under various criterion.

Link	λ_i	Criterion (gives one - way values)			
		Square root	Proportional	Equal	Min - Max
x_1	3.15	40.4	48	41.3	42
x_2	3.55	37.8	41.8	41.9	42
x_3	0.13	206	1149	36.6	42
x_4	3.64	38	41.1	42.1	42
x_5	0.82	78.8	182.5	37.6	42
x_6	3.88	36.2	38.3	42.5	42
x_7	9.95	22.6	15	57.3	42

Table 3.8 Average time delay (T_{avg}) in m.sec under various criterion.

Square root criterion	Proportional criterion	Equal criterion	Min- Max criterion
44.32	54.5	62.26	54.53

Table 3.9 Expression for PI under various design criterions

Criterion	PI Expression assuming equal values for link reliability
Square root	$\frac{1}{58}[58p^6 + 35p^6q + 132p^5q + 32p^5q^2 + 109p^4q^2 + 49p^3q^2 + 5p^4q^3 + 28p^3q + 30p^2q]$
Proportional	$\frac{1}{52}[52p^6 + 111p^5q + 59p^4q^2 + p^4q^2 + 29p^4q + 24p^3q + 35p^3q^2 + 28p^2q]$
Equal	$\frac{1}{110}[110p^6 + 164p^5q + 110p^6q + 55p^5q^2 + 220p^4q^2 + 55p^4q^3 + 110p^3q^2 + 110p^3q + 55p^2q]$
Min - Max	$\frac{1}{54}[54p^6 + 51p^6q + 116p^5q + 28p^5q^2 + 130p^4q^2 + 24p^4q^3 + 75p^3q^2 + 27p^3q + 27p^2q]$
Reliability without considering capacities	$p^4q^3 + 2p^4q^2 + p^3q^2 + 2p^3q + p^2$

as compared to heavy traffic link x_7. Thus, user of former type is penalized in this case. Average network delay is almost equal with proportional and min - max criterion. In min - max criterion delay in each link is equal whereas in proportional assignment strategy light traffic links are having more delays even as compared to square root assignment strategy. Thus in min - max criterion each link has equal preference.

To select a particular criterion, reliability analysis is made. PI expressions are obtained using the method described in section 3.2. Assume $n_4(n_1)$ is source (sink) node. Table 3.9 lists these expressions. In these expressions it is assumed that each link has an equal probability of failure. PI figure is then, computed by considering reliability of each link to be .8,.9,.95 and .99. It is evident from the table 3.10 that PI for min - max criterion is more as

Table 3.10 PI for various design criterions

Link reliability	Square root	Proportional	Equal	Min-max	Reliability (without integrating capacity)
.80	.6142	.5963	.6407	.6437	.9013
.90	.7967	.7837	.8102	.8147	.9769
.95	.8961	.8882	.9014	.9058	.9889
.99	.9789	.9770	.9799	.9814	.9998

compared to any other criteria. Thus, min -- max criterion performs better under failure condition of links and may be selected as the capacity assignment rule.

3.4 CONCLUSION

In this chapter the problem of integrating the link capacity with its reliability of success is considered in the performance measure of a CCN. It has utilized the concept of terminal reliability evaluation. The method is simple and advantageous as it applies to large networks too. Above performance measure is used to select a particular capacity assignment rule if reliability is considered as another constraint besides the cost.

CHAPTER 4

ROUTE SEQUENCE ENUMERATION

Enumeration of some or all paths in any general network is an important step from the point of view of reliability evaluation. Methods [21,51,52] use a connection matrix an analytical correspondence of the system graph, and obtain $[C]^r$ to define a r-size path, where $r = 1,\ldots,k-1$. The problem of terminal pair reliability can, alternatively, be translated into NNGOS (node to node grade of service) or EEB (end to end blocking or congestion), an important measure of service quality of a circuit switched network. The NNGOS for each node pair depends on the node to node routing table and a call control rule [12,114]. The concept of route (path loss)sequence is, then, employed in system reliability analysis to determine the probability of each route being used to complete a call. Thus enumeration of route sequence containing the complete information about the call routing becomes an important problem as it helps compute EEB. This chapter, first, envisages augmented route tree approach [12] to derive route sequences for completing calls from a source node to a destination node under different call control rules. Secondly, an algebraic method is discussed. The proposed method enumerates not only all paths but also the route sequence. It is straightforward and requires only simple algebraic concepts. Appendix 1 list a computer program for it. It has successfully been run on DEC 20 system.

4.1 AUGMENTED ROUTE TREE APPROACH

Definition 4.1 – A path between two nodes n_i and n_j is an ordered set of links connecting n_i to n_j. For a connected graph there must exist at least one path between every origin/destination node pair.

Definition 4.2 – All paths originating from source node do not terminate at destination node. Those which terminate at destination are called completion routes and the rest are called loss routes. The route (path loss) sequence, thus, contains the full information about the call routing.

Definition 4.3 – A tandem node is any intermediate node of a route between the originating node and the terminating node.

Definition 4.4 – A routing plan or stretegy specifies, for every origin/ destination node pair, a first choice (or primary route) and a number of alternate choice routes. A routing stretegy is completely described by a routing table and a call control rule. A typical routing table for the network of figure 4.1 is given in table 2.1.

Definition 4.5 – The routing table is a k x k size matrix with the diagonal blocks unused. For $n_i \neq n_j$, the (n_i,n_j) entry denotes the node(s) which is (are) employed as the tandem node in the alternate route. In table 2.1, the n_3 to n_5 block has three enteries in the order n_5,n_4,n_2. This

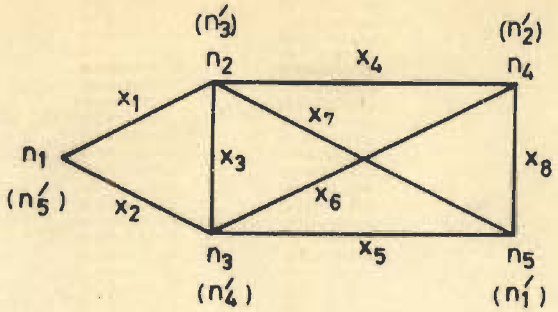


Figure 4.1. A Communication Network.

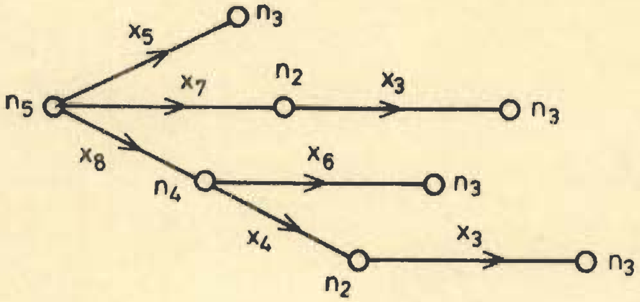


Figure 4.2. Route Tree for n_5-n_3 Calls.

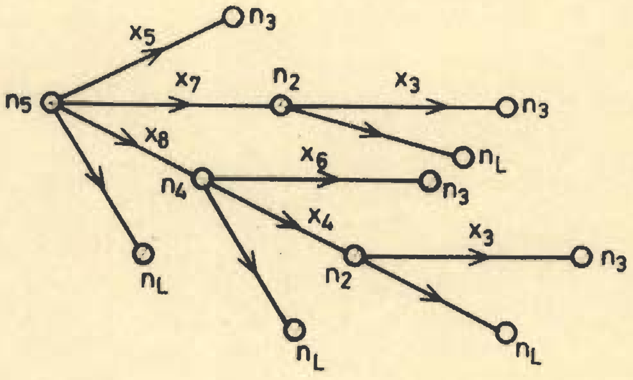


Figure 4.3. Augmented Route Tree for SOC.

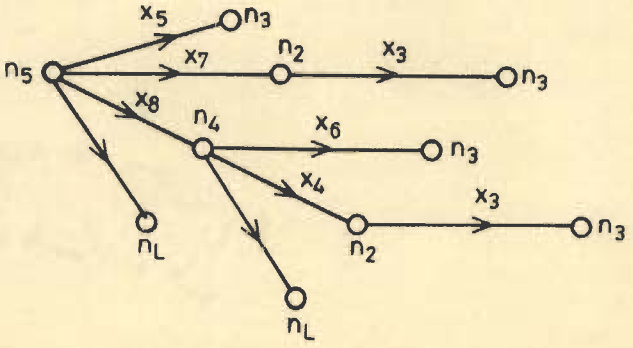


Figure 4.4. Augmented Route Tree Under OOC with Spill.

means that a call reaching node n_3 (regardless of the origin of the call) and destined for node n_5 can only be routed through the links n_3-n_5, n_3-n_4, n_3-n_2. If all of these three choices are permitted, then the preferred order is, first choice n_3-n_5, second choice n_3-n_4, last choice n_3-n_2 [12].

A number of call control rules are presently used in various communication networks. Some typical examples include SOC, OOC with spill forward [12]. SOC is also termed as sequential or progressive routing. In what follows, an algorithm is given to construct a route tree for n_i-n_j calls. Note that the structure of a route tree bears a relationship with the routing table and call control rules. The steps are as follows -

1. The root of the tree has the values n_1.
2. The call from a node always first attempts to use the primary route.
3. If the call control rules specify the alternate choice routes, use them only when primary route is not available.
4. To ensure that the call progress in the network is cycle-free, care be taken that no traffic parcel is switched twice through the same node. For this, necessary precautions must be exercised in constructing the routing table.

Lin et al [12] have shown that the route tree obtained as above is not always sufficient for describing the network operation. Two different call control rules may produce exactly the same route tree and the same sequence of paths. And yet the actual network operation may be different. Consider figure 4.2 where this situation is depicted.

Augmented route tree approach [12] overcomes this problem by introducing the concept of path loss sequence. In it, a fictitious loss node n_L is added into the route tree using the following addition rules [12].

Rule 4.1 [SOC strategy] – Add one directed branch to every nonpendant node of a route tree, below all existing branches leaving that node and label the terminating node n_L.

Rule 4.2 [OOC and OOC with spill stretegy] – Add one directed branch to each node of route tree which either is originating node, or assumes the role of originating node through spill forward action, below all existing branches leaving that node and label the terminating node n_L.

Example 4.1 – The augmented route trees for n_5 – n_3 calls (refer figure 4.1 and table 2.1) constructed according to above rules are shown in figures 4.3 and 4.4 for SOC and OOC with spill respectively. Note that the two augmented trees for n_5 – n_3 calls are different, despite the fact that they are derived from the same route – tree of figure 4.2 and same routing table 2.1.

In the augmented route tree, a pendant node is labeled either with destination node n_k or with loss – node n_L. There is a unique path from root of tree to every pendant (or tip) node. If the tip is $n_k(n_L)$, corresponding node sequence is designated by $P_j(L_h)$ and is termed as path (loss) sequence. The node sequence of P_j shows the way the call is routed. On the other hand, the sequence in L_h depicts exactly how far the call has reached before getting blocked. The interlaced sequence of P_j and L_h, obtained using depth – first search process is the path loss sequence for a given source – destination node pair. As an example, figure 4.4 shows an augmented route tree for $n_5 - n_3$ calls under OOC with spill at tandem node n_4. Performing the depth – first search on figure 4.4, we obtain the following route sequence.

$P_1(x_5)$, $P_2(x_7x_3)$, $P_3(x_8x_6)$, $P_4(x_8x_4x_3)$, $L_1(x_8)$, $L_2(1)$

4.2 ALGEBRAIC METHOD

4.2.1 Path sequence Enumeration – The following definitions and theorems are useful to help explain the procedure that will be discussed later in this section. Consider the reliability graph G in which branches go **from** (to) the source (sink) node and no branch goes to (from) it.

Definition 4.6 – A path of size one designated as a_{ij} is said to exist if there is directed branch connecting node n_i to n_j. An undirected branch is resolved as two directed branches in anti parallel.

Definition 4.7 – Any pair of nodes which are connected by an edge in a graph is termed as adjacent nodes. Nodes n_i and n_j not adjacent to each other are characterized by $a_{ij} = 0$.

Definition 4.8 – Connectivity of a node n_j, designated as A_j, is an ordered set of a_{ij}'s for $i = 1, \ldots, h$;

$A_j = \langle a_{1j}, a_{2j}, \ldots, a_{hj} \rangle$ where h represents the maximum size of A_j and is equal to $(k-1)$. For example, A_3 for figure 4.5 is given as $\langle a_{13}, a_{23}, a_{33} \rangle$.

Theorem 4.1 – $a_{ip}\, a_{iq} = 0$

Proof. Note that $a_{ip}\, a_{iq}$ is union of two branches a_{ip} and a_{iq}. Alternatively, it reflects travelling simultaneously from source node n_i towards two adjacent nodes n_p and n_q. It is an undesirable event from path enumeration point of view.

$\square$

Corollary 4.1 – All paths between n_1 and n_k are given by the cartesian products of A_j's defined as under

$$PT_{1k} = \underset{j=2}{\overset{k}{\times}} A_j$$

$\square$

Corollary 4.2 – The paths between any source node n_i and sink node n_k, such that $i < k$, are

$$PT_{ik} = A_1 \times \ldots \times A_{i-1} \times A_{i+1} \times \ldots \times A_k$$

$\square$

Corollary 4.3 – All paths between a particular source node n_1 to any other node n_t, such that $t < k$, are

$$PT_{1t} = \underset{j=2}{\overset{k}{\times}} A'_j$$

where A'_j is derived from A_j. For this, an entry a_{tj} in A_j is replaced by a_{kj}, while all other enteries of A_j are kept unchanged.

$\square$

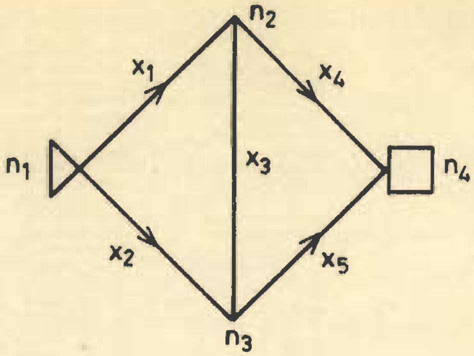
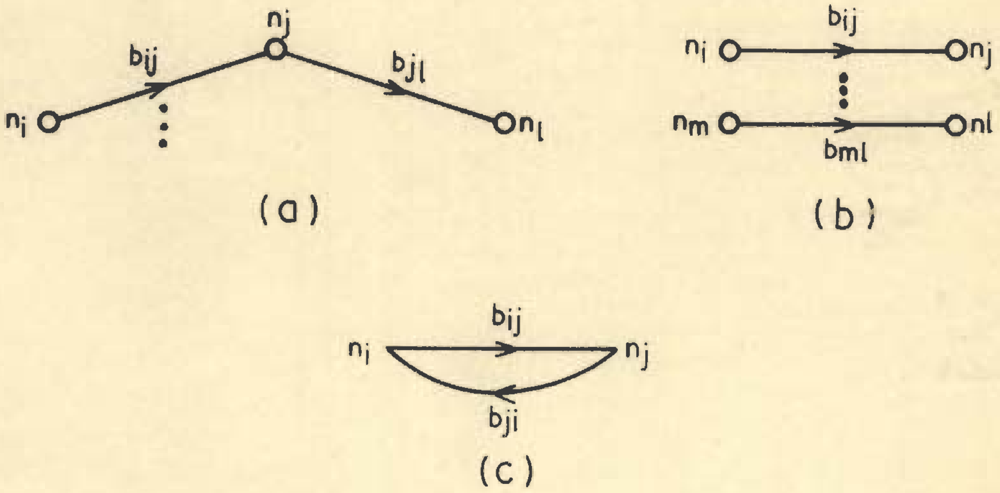


Figure 4.5. A Bridge Network.



(a)

(b)

(c)

Figure 4.6. Illustrating Concatenation Operation Rules.

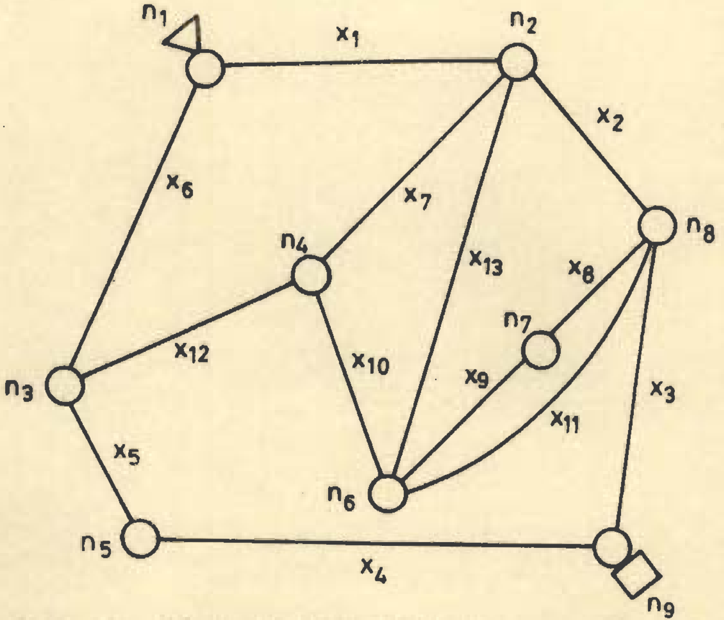


Figure 4.7. A Simple Long Distance Telephone Network.

Corollary 4.4 – The paths between any source node n_i and sink node n_t are

$$PT_{it} = A'_1 \text{ X } \ldots \text{ X } A'_{i-1} \text{ X } A'_{i+1} \text{ X } \ldots \text{ X } A'_k$$

where A'_j is same as defined in corrolary 4.3 above. $\square$

The proofs of corollaries 4.1 through 4.4 follow directly from the connection matrix representation of the reliability graph. For figure 4.5 the connection matrix is defined as

$$[C] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \qquad (4.1)$$

Note that $a_{21}, a_{31}, a_{41}, a_{42}, a_{43}$ and a_{14} are zero for figure 4.5. Delete a column i (row j) corresponding to node $n_i(n_j)$ assigned to source (sink) node. The column j enteries of reduced (k – 1) x (k – 1) size connection matrix define A_j. To be specific, if n_2 is source node and n_3 is the sink node, then the reduced connection matrix is obtained by deleting column 2 and row 3 from (4.1) and is given as

$$\begin{bmatrix} a_{11} & a_{13} & a_{14} \\ a_{21} & a_{23} & a_{24} \\ a_{41} & a_{43} & a_{44} \end{bmatrix} \qquad (4.2)$$

A'_1, A'_3 and A'_4 are then defined as

$$A'_1 = \langle a_{11}, a_{21}, a_{41} \rangle$$

$A'_3 = \langle a_{13}, a_{23}, a_{43} \rangle$

$A'_4 = \langle a_{14}, a_{24}, a_{44} \rangle$

Note that the same result can be obtained using definition 4.8 and corollary 4.3.

Theorem 4.2 – For a complete polygon with k nodes and all bidirectional branches the upper bound on the number of paths are

$$m = (k - 1)!$$

Proof. To prove this, apply induction. For $k = 3$, $A_2 \times A_3$ will result into $m = 2$. If $k = 4$, $A_2 \times A_3 \times A_4$ will give rise to 6 paths. Similarly it can be proved that $m = 24$, for $k = 5$ and hence the theorem follows.

□

However, by recognizing the group of terms mentioned in theorem 4.1, one can further reduce the limit value to

$$m = (k - 2) \sum_{i=o}^{k-2} 1/i! \qquad (4.3)$$

Equation 4.3 has also been reported in [19].

Algorithm . Let the total in – degree of node n_i be d_i. It means A_i will have the cardinality as d_i. Consider an ordered set D whose elements are various d_i's. Thus,

$D = \langle d_1, d_2, \ldots, d_k \rangle$

Similarly, define J where

$J = \langle j_1, j_2, \ldots, j_k \rangle$

j_i being the current element of ordered set A_i. In order to effectively use the concept of theorem 4.1, consider an operation FIRST $\{x_{ij}\}$ such that

FIRST $\{x_{ij}\} = i$

Thus, it picks up the first variable of the argument x_{ij}. In what follows, the MAIN procedure for path enumeration is described. It uses an Algol like language. Appendix 1, however, lists the computer program for it.

```
procedure MAIN
          call GEN (1,k)
end       MAIN
```

Note that the algorithm requires GEN(i,k) procedure. To understand the steps use a notation where $\beta[i,j]$ denotes the jth element of β_i. Inputs $\beta_1, \beta_2, \ldots, \beta_{k-1}$ are derived from A_j's by replacing first A_j term with β_1, second by β_2 and so on till all ($k-1$) terms have been considered. This substitution generalizes the following GEN procedure. It utilizes again Algol like language for its description.

```
procedure  GEN(i,k)
    for     $J(i) \leftarrow 1$ to $D(i)$ do
            $r \leftarrow 1$
            for $j \leftarrow 1$ to $i-1$ do
              if FIRST$\{\beta[i,J(i)]\}$ = FIRST$\{\beta[j,J(j)]\}$ then
                                          [$r \leftarrow 0$; exit]
            end
              if $r \neq 0$ then
                 [if $i = k-1$ then
                  for $j \leftarrow 1$ to $k-1$ do print $\beta[j,J(j)]$ end
                  else call GEN(i+1,k)
    end
end        GEN
```

Example 4.2 -- For figure 4.5, let n_2 be source and n_3 be sink node. As defined earlier, various terms of A'_i s are

$$A'_1 = < a_{11}, a_{21}, a_{41} >$$
$$A'_3 = < a_{13}, a_{23}, a_{43} >$$
$$A'_4 = < a_{14}, a_{24}, a_{44} >$$

Note that a_{21}, a_{41}, a_{43} and a_{14} are all zeros because the nodes connected to them are not adjacent nodes. The modified connectivity set is

$$A'_1 = < a_{11} >$$
$$A'_3 = < a_{13}, a_{23} >$$
$$A'_4 = < a_{24}, a_{44} >$$

Thus, the path list between n_2 and n_3 is obtained using

$$A'_1 \text{ X } A'_3 \text{ X } A'_4$$

which results into $a_{11}\ a_{23}\ a_{44}$. Replacing all a_{ii}'s terms by 1, a single path obtained in this case is a_{23}.

4.2.2 Computing Route Sequence -- Path enumeration method considered above is modified to generate the route (or path loss) sequence in a CCN for a given routing table and call control rule. First modification necessiates the use of terminal numbering convention, TNC [115]. It helps avoid a traffic parcel routing through the same node for more than once. In this, $n_1 (n_k)$ is used to designate source (sink) node. Intermediate (or tandem) nodes are numbered such that

i. each node is assigned a different number, and

ii. the number assigned to a higher level node follows lexicographic ordering.

Secondly, connectivity of a node n_j, described in definition 4.8, is modified. It is now considered as an ordered set of b_{ji}'s for $i = 1,\ldots,k$ as $B_j = \langle b_{j1}, b_{j2},\ldots,b_{jk} \rangle$. The set B_j, thus, contains the information regarding alternate choice routes available at node n_j. For example, B_2 in the figure 4.5 is given as $\langle b_{21}, b_{22}, b_{23}, b_{24} \rangle$.

Definition 4.9 – The concatenation operation, $\textcircled{X}$, on two paths of unit distance is defined by following rules.

$$b_{ij} \textcircled{X} b_{jl} = b_{ij}\, b_{jl} \qquad (4.4a)$$

$$b_{ij} \textcircled{X} b_{ml} = b_{ij} \qquad (4.4b)$$

$$b_{ij} \textcircled{X} b_{ji} = 0 \qquad (4.4c)$$

The implication of rules (4.4a) through (4.4c) is obvious from figures 4.6a to 4.6c respectively.

In order to enumerate path loss sequence in different call control strategies, B_j's needs restructuring. For it, following construction rules are helpful.

1. For the tandem node n_j not assuming the role of originating office, consider b_{ji} (the first choice route between n_j and n_i) only in the definition of B_j.

2a. A path of size 1, b_{jj}, is used in the definition of B_j iff the node n_j assumes the role of an originating office ie., alternative choices of outgoing links for a call are permitted at node n_j.

b. In this event, the enteries in B_j should reflect the permitted choice routes with primary route appearing first and so on. The last entry is b_{jj}.

These substitutions correspond to the descriptions given by Lin et al [12] for SOC, OOC and arbitrary routing plan. At this stage, the difference between B_j and A_j is obvious; the connectivity set B_j providing much more information in comparison with A_j.

Example 4.3 – Consider a bridge network shown in figure 4.5. Here $n_1(n_4)$ represents source (sink) node. Various B_j's are

$B_1 = < b_{11}, b_{12}, b_{13}, b_{14} >$

$B_2 = < b_{21}, b_{22}, b_{23}, b_{24} >$

$B_3 = < b_{31}, b_{32}, b_{33}, b_{34} >$

Note that b_{14}, b_{21} and b_{31} are all zeros because the nodes connected to them are not adjacent nodes. Assuming call flow from n_2 to n_3, b_{32} is also zero. The simplified B_j's after incorporating the alternate choice routes will be

$B_1 = < b_{12}, b_{13}, b_{11} >$

$B_2 = < b_{23}, b_{24}, b_{22} >$

$B_3 = < b_{34}, b_{33} >$

Enteries b_{22} and b_{33} retained with B_2 and B_3 respectively emphasize that tandem nodes n_2 and n_3 assume the role of originating office.

Theorem 4.3 – Path loss sequence between source node n_1

and sink node n_k is given by concatenation operation of B_j's as

$$PL_{1k} = B_1 \otimes B_2 \otimes \ldots \otimes B_{k-1}$$

Proof. Obvious

$\square$

Algorithm – Based on theorem 4.3 the steps of the algorithm to compute PL sequence between a node pair are as follows:

1. Obtain B_j's. Make use of construction rules to incorporate route strategy and call control rules.
2. Consider definition 4.9 and theorem 4.3 to derive path loss sequence between a source/destination node pair.

The algorithm presented above gives all the route sequences in any CCN. Note here that terms with coefficient as b_{ii} represents loss sequence and is equivalent to a loss route of augmented route terminating in n_L via node n_i as shown in figure 4.4.

Example 4.4 – Consider a network of figure 4.1 for which augmented route tree approach has been applied to enumerate a path loss sequence between node n_5 to n_3 and the results are given in [12].

Under TNC convention node n_5 is relabeled as node n_1' and node n_3 as n_4'. Other tandem nodes follow lexicographic ordering. It is shown in figure 4.1 under brackets. In SOC scheme each tandem node assumes the role of originating office and hence every B_j contains size 1 path b_{jj}. The various B_j's after incorporating alternate choice routes are

$B_1 = \langle b_{14}, b_{13}, b_{12}, b_{11} \rangle$

$B_2 = \langle b_{24}, b_{23}, b_{22} \rangle$

$B_3 = \langle b_{34}, b_{33} \rangle$

Thus, path loss sequence between nodes n'_1 and n'_4 call is obtained as under

$$\begin{aligned} PL_{14} &= B_1 \otimes B_2 \otimes B_3 \\ &= b_{14}, b_{13}b_{34}, b_{13}b_{33}, b_{12}b_{24}, b_{12}b_{23}b_{34}, b_{12}b_{23}b_{33}, \\ &\quad b_{12}b_{22}, b_{11} \end{aligned}$$

These are same as that mentioned in [12] and shown in figure 4.3.

Under OOC with spill, assume node n'_2 is acting as orginating office. This effect is mentioned in table 2.1 by asterisks. Here various B_j's terms are

$B_1 = \langle b_{14}, b_{13}, b_{12}, b_{11} \rangle$

$B_2 = \langle b_{24}, b_{23}, b_{22} \rangle$

$B_3 = \langle b_{34} \rangle$

Note that b_{33} is not present in B_3 as node n'_3 do not assume the role of originating office. Thus, path loss sequence for OOC with spill is

$$\begin{aligned} PL_{14} &= B_1 \otimes B_2 \otimes B_3 \\ &= b_{14}, b_{13}b_{34}, b_{12}b_{24}, b_{12}b_{23}b_{34}, b_{12}b_{22}, b_{11} \end{aligned}$$

The results are again same as shown in figure 4.4.

Example 4.5 – In this example, a simple long distance telephone network shown in figure 4.7 is considered [116]. For $n_1(n_9)$ representing source (sink) node, various B_j's are

$B_1 = < b_{11}, b_{12}, b_{13}, b_{14}, b_{15}, b_{16}, b_{17}, b_{18}, b_{19} >$
$B_2 = < b_{21}, b_{22}, b_{23}, b_{24}, b_{25}, b_{26}, b_{27}, b_{28}, b_{29} >$
$B_3 = < b_{31}, b_{32}, b_{33}, b_{34}, b_{35}, b_{36}, b_{37}, b_{38}, b_{39} >$
$B_4 = < b_{41}, b_{42}, b_{43}, b_{44}, b_{45}, b_{46}, b_{47}, b_{48}, b_{49} >$
$B_5 = < b_{51}, b_{52}, b_{53}, b_{54}, b_{55}, b_{56}, b_{57}, b_{58}, b_{59} >$
$B_6 = < b_{61}, b_{62}, b_{63}, b_{64}, b_{65}, b_{66}, b_{67}, b_{68}, b_{69} >$
$B_7 = < b_{71}, b_{72}, b_{73}, b_{74}, b_{75}, b_{76}, b_{77}, b_{78}, b_{79} >$
$B_8 = < b_{81}, b_{82}, b_{83}, b_{84}, b_{85}, b_{86}, b_{87}, b_{88}, b_{89} >$

Here, 51 b_{ij}'s are zeros as various n_i and n_j's are not adjacent nodes. Table 4.1 shows a partial routing table between nodes n_1 and n_9.

Table 4.1 Partial routing table between nodes n_1 and n_9 for figure 4.7 (Nodes marked with asterisk act as originating office in OOC with spill strategy)

FROM → TO ↓	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9
n_9	$\overset{*}{n_2}, \overset{*}{n_3}$	n_8, n_6, n_4	n_5, n_4	n_6	n_9	n_8, n_7	n_8	n_9	–

Table 4.2 Various B_j's for figure 4.7

Call control rules	B_j's
OOC	$B_1 = < b_{12}, b_{13}, b_{11} >$ $B_2 = < b_{28} >$ $B_3 = < b_{35} >$ $B_4 = < b_{46} >$ $B_5 = < b_{59} >$ $B_6 = < b_{68} >$ $B_7 = < b_{78} >$ $B_8 = < b_{89} >$
SOC	$B_1 = < b_{12}, b_{13}, b_{11} >$ $B_2 = < b_{28}, b_{26}, b_{24}, b_{22} >$ $B_3 = < b_{35}, b_{34}, b_{33} >$ $B_4 = < b_{46}, b_{44}, >$ $B_5 = < b_{59}, b_{55} >$ $B_6 = < b_{68}, b_{67}, b_{66} >$ $B_7 = < b_{78}, b_{77} >$ $B_8 = < b_{89}, b_{88} >$
OOC with spill at node n_2 and n_3 (ie., arbitrary routing plan)	$B_1 = < b_{12}, b_{13}, b_{11} >$ $B_2 = < b_{28}, b_{26}, b_{24}, b_{22} >$ $B_3 = < b_{35}, b_{34}, b_{33} >$ $B_4 = < b_{46} >$ $B_5 = < b_{59} >$ $B_6 = < b_{68} >$ $B_7 = < b_{78} >$ $B_8 = < b_{89} >$

Table 4.3 Routing of source/ destination call(figure 4.7)

Call control rules	Number of paths	Modes in paths
SOC	8	$P_1(x_1x_2x_3)$, $L_1(x_1x_2)$,$P_2(x_1x_{13}x_{11}x_3)$, $L_2(x_1x_{13}x_{11})$,$P_3(x_1x_{13}x_9x_8x_3)$, $L_3(x_1x_{13}x_9x_8)$,$L_4(x_1x_{13}x_9)$,$L_5(x_1x_{13})$, $P_4(x_1x_7x_{10}x_{11}x_3)$,$L_6(x_1x_7x_{10}x_{11})$, $P_5(x_1x_7x_{10}x_9x_8x_3)$,$L_7(x_1x_7x_{10}x_9x_8)$, $L_8(x_1x_7x_{10}x_9)$,$L_9(x_1x_7x_{10})$,$L_{10}(x_1x_7)$, $L_{11}(x_1)$,$P_6(x_6x_5x_4)$,$L_{12}(x_6x_5)$, $P_7(x_6x_{12}x_{10}x_{11}x_3)$,$L_{13}(x_6x_{12}x_{10}x_{11})$, $P_8(x_6x_{12}x_{10}x_9x_8x_3)$,$L_{14}(x_6x_{12}x_{10}x_9x_8)$, $L_{15}(x_6x_{12}x_{10}x_9)$,$L_{16}(x_6x_{12}x_{10})$, $L_{17}(x_6x_{12})$,$L_{18}(x_6)$,$L_{19}(1)$
OOC	2	$P_1(x_1x_2x_3)$, $P_2(x_6x_5x_4)$ $L_1(1)$
OOC with spill at node n_2 and n_3(ie.,arbitrary routing plan)	5	$P_1(x_1x_2x_3)$, $P_2(x_1x_{13}x_{11}x_3)$, $P_3(x_1x_7x_{10}x_{11}x_3)$,$L_1(x_1)$,$P_4(x_6x_5x_4)$, $P_5(x_6x_{12}x_{10}x_{11}x_3)$,$L_2(x_6)$,$L_3(1)$.

Using information given in table 4.1, various B_j's obtained are shown in table 4.2 under different call control rules. The number of paths along with path loss sequence between n_1 and n_9 are listed in table 4.3.

4.3 CONCLUSION

In this chapter the problem of enumerating path loss sequence between a source/ destination node pair is considered. Once enumerated there is no need to further refer to routing table and call control rules. As discussed in chapter 2 this information is vital in obtaining NNGOS of a circuit switched network for which the detailed discussion is presented in chapter 5.

CHAPTER 5

POINT TO POINT CONGESTION COMPUTATION

As discussed earlier, two problems namely EEB and terminal pair reliability are generally considered for the performance analysis of a circuit switched network. This chapter describes methods to compute EEB in the telecommunication network. While discussing the EEB problem, an integrated approach is adopted. Many methods exist in the literature for terminal pair reliability evaluation. Chapter 4 provides a method for route sequence. Section 5.1 uses this sequence and the techniques in system reliability analysis to compute point to point congestion. The subsequent sections present recursive algorithms for determining EEB probabilities in the network under various call control rules discussed in chapter 2. Algorithms use decomposition approach in which a succession of keystone element is chosen. Note that, the recursive approach leads to concise descriptions and also languages like PL/1 and Pascal which support recursion can easily be utilized to program it. Methods, discussed here, are illustrated with examples.

5.1 USING RELIABILITY EVALUATION METHOD

Chapter 2 overviews the reliability evaluation methods and considers them into two basic groups: 2 - step and 1 -step approaches. Two step methods require apriori knowledge of either path or cutset between a source/destination node pair. To obtain the reliability/unreliability of a general network, this information is then transformed into EMD set by using a suitable technique. Lin et al [12] have shown that these techniques in system reliability analysis can be made to compute EEB provided the path or cutset information is suitably replaced by path loss sequence to suit a routing strategy and call control rule. A four step procedure, first described in [12], is given as under.

Step 1. From the given network configuration, the call control rule and the routing table, generate the pathloss sequence for each node pair.

Step 2. Find the link blocking probabilities under the given node to node traffic demands.

Step 3. From the link blocking probabilities found in step 2, find the probability of each path in the path loss sequence being used to complete a call.

Step 4. Using the results of step 3, determine EEB (NNGOS) for each node pair by

$$B_G(K) = 1 - \sum_j P_r(P_j \text{ used}) \tag{5.1}$$

$$= \sum_h P_r\,(L_h \text{ used}) \tag{5.2}$$

Equations (5.1) or (5.2) determine NNGOS between nodes n_i and n_j as the probability of each route being used to complete or lose a call. Step 1 is considered in chapter 4. The proposed algebraic method not only obtains the path loss sequence but also arranges them in order of preference of choice routes. The order of completion and loss routes in route sequence is, however, immaterial in single loss route strategies (viz OOC, predictive routing, reliability analysis approach) because

i) EEB in this situation is not changed by reordering its completion (path) routes, and

ii) for single loss route, the route sequence contains only one loss route which is always the last route of the route sequence.

Nontheless, the order of path loss sequence plays an important role in multiple loss route strategies (viz SOC and arbitrary routing) where EEB computed does depend on it.

For step 2, methods [12,117] exist in the literature and any one of them may be used to compute link availabilities. The availability of a link depends on its size (ie.,number of transmission circuits) and total offered traffic (ie., end to end traffic plus traffic distributed from other links due to alternate routing). An example considered in section 2.5 discusses an approach to determine link availabilities under various assumptions.

Methods of reliability analysis are applied for steps 3 and 4. For it, we consider E – operator technique [21] because the computer program for it is available. Moreover, it is computationally efficient as compared to [26]. The substeps are already described in chapter 3.

Example 5.1 – Consider the network of figure 4.1 for which path loss sequence have been enumerated in example 4.4 for n_5 to n_3 calls. For OOC with spill forward call control strategy the route sequence is

$$P_1(x_5),\ P_2(x_3x_7),\ P_3(x_6x_8),\ P_4(x_3x_4x_8),\ L_1(x_8),\ L_2(1)$$

The path loss polynomial PL is obtained

$$PL = X_5 \cup X_3X_7 \cup X_6X_8 \cup X_3X_4X_8 \cup X_8 \cup (1)$$

F_i's and $E(F_i)$'s for $i = 2,3,\ldots,6$ are obtained as shown in table 5.1. Using these terms, PL (disjoint) is computed as –

$$\begin{aligned} PL(\text{disjoint}) = {} & X_5 \cup X_3X_7(\overline{X}_5) \cup X_6X_8\ (\overline{X}_5\overline{X}_7 \cup \overline{X}_5\overline{X}_3X_7) \\ & \cup X_3X_4X_8(\overline{X}_5\overline{X}_6\overline{X}_7) \cup X_8(\overline{X}_3\overline{X}_5\overline{X}_6 \cup X_3\overline{X}_4\overline{X}_5\overline{X}_6\overline{X}_7) \\ & \cup (\overline{X}_5\overline{X}_7\overline{X}_8 \cup \overline{X}_3\overline{X}_5X_7\overline{X}_8) \end{aligned}$$

Thus, we have –

$P_r(P_1 \text{ used}) = p_5$

$P_r(P_2 \text{ used}) = p_3p_7q_5$

$P_r(P_3 \text{ used}) = p_6p_8(q_5q_7 + q_3q_5p_7)$

$P_r(P_4 \text{ used}) = p_3p_4p_8(q_5q_6q_7)$

$P_r(L_1 \text{ used}) = p_8(q_3q_5q_6 + p_3q_4q_5q_6q_7)$

$P_r(L_2 \text{ used}) = q_5q_7q_8 + q_3q_5p_7q_8$

From (5.2) the point to point congestion expression becomes

$$B_G(K) = p_8(q_3q_5q_6 + p_3q_4q_5q_6q_7) + q_5q_7q_8 + q_3q_5p_7q_8 \quad (5.3)$$

The equation (5.3) is same as that obtained in [12]. The values of link blocking probabilities when substituted in

Table 5.1 Terms for Example 5.1. α_i represents either P_j or L_h.

F_i	$E(F_i)$	$\alpha_i E(F_i)$
$F_2=X_5$	$\bar{X}_5$	$X_3X_7(\bar{X}_5)$
$F_3=X_5 \cup X_7X_3$	$\bar{X}_5\bar{X}_7 \cup \bar{X}_5\bar{X}_3X_7$	$X_6X_8(\bar{X}_5\bar{X}_7 \cup \bar{X}_5\bar{X}_3X_7)$
$F_4=X_5 \cup X_7 \cup X_6$	$\bar{X}_5\bar{X}_6\bar{X}_7$	$X_3X_4X_8(\bar{X}_5\bar{X}_6\bar{X}_7)$
$F_5=X_5 \cup X_3X_7 \cup X_6$ $\cup X_3X_4$	$\bar{X}_3\bar{X}_5\bar{X}_6 \cup X_3\bar{X}_4\bar{X}_5\bar{X}_6\bar{X}_7$	$X_8(\bar{X}_3\bar{X}_5\bar{X}_6 \cup X_3\bar{X}_4\bar{X}_5\bar{X}_6\bar{X}_7)$
$F_6=X_3X_7 \cup X_5 \cup X_8$	$\bar{X}_5\bar{X}_7\bar{X}_8 \cup \bar{X}_5\bar{X}_3X_7\bar{X}_8$	$(\bar{X}_5\bar{X}_7\bar{X}_8 \cup \bar{X}_3\bar{X}_5X_7\bar{X}_8)$

Table 5.2 EEB for figure 4.1

Call control rule	Expression for EEB
SOC	$q_2p_4q_6 + p_1p_3q_2q_4q_5q_6 + p_1q_2q_3q_4q_5$ $+ q_1q_2q_4$
OOC	$q_3q_5q_6q_8 + p_3q_5q_6q_7q_8 + q_5q_7q_8 + q_3q_5p_7q_8$
OOC with spill forward	$q_3q_5q_6p_8 + p_3q_4q_5q_6q_7p_8 + q_5q_7q_8$ $+ q_3q_5p_7q_8$

this symbolic expression gives EEB for figure 4.1. For the same network symbolic expressions under SOC and OOC schemes are also obtained and table 5.2 shows them.

5.2 RECURSIVE TECHNIQUE

5.2.1 Graphical Approach – The following assumptions made in chapter 2 are rewritten here to complete the discussion.

1. All elements are always operating (no stand – by or switched redundancy). Each element, and the network, is either good or bad.
2. The network is free from directed cycles and self loops. It excludes such possibilities as packet switching twice through the same node.
3. The states of all elements are mutually statistically independent.

The following definitions are useful to help explain the procedure that will be discussed later in this section. Note that, for the sake of simplicity and ease of explanation, jth node has been expressed as j instead of n_j in this section.

Definition 5.1 – A feasible (s,t) cutset of a probabilistic graph G that can have both directed and undirected branches is a minimal set of unreliable elements, denoted by CQ_r, such that every (s,t) path has at least one element common with CQ_r. For sake of brevity, a single word cutset in this chapter means a feasible (s,t) cutset.

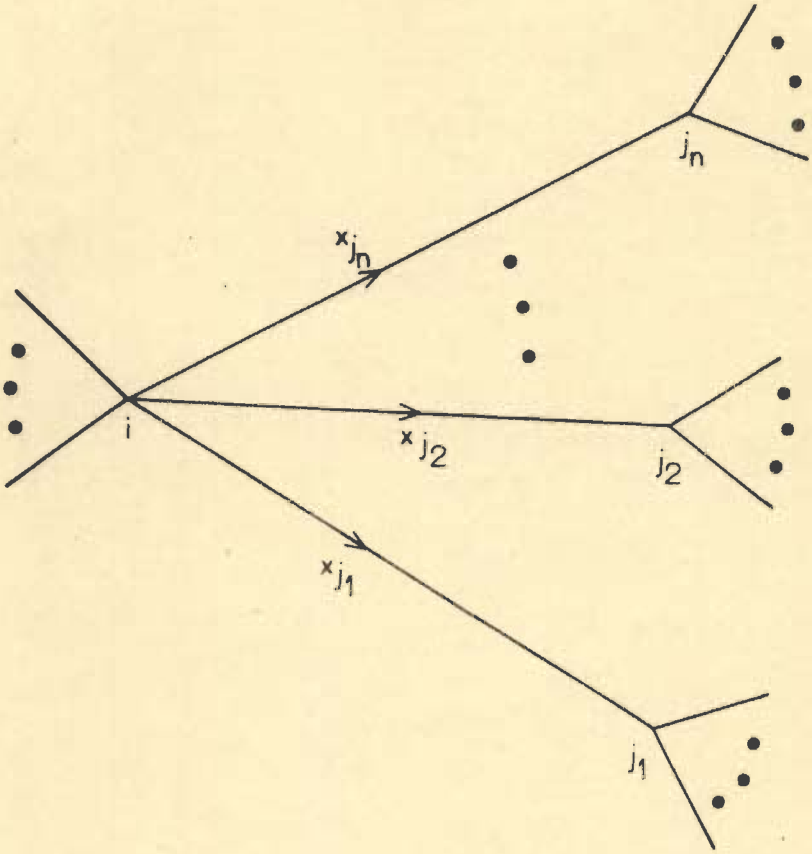


Figure 5.1. Illustration for Theorem 5.1.

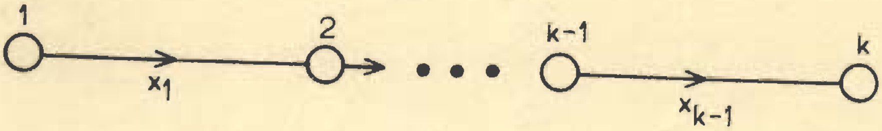


Figure 5.2. A Series Connection.

Definition 5.2 — The number of edges incident out of a node i is called the outdegree of i.

Definition 5.3 — The concept of blocking refers to the fact that a call encounters an "all equipment busy condition" on a given link.

Definition 5.4—A final link is one from which a blocked call is cleared (lost) from the network, ie., it does not overflow to another link. Figure 5.1 illustrates the concept of final link. A call leaving node i has $x_{j_1}, \ldots, x_{j_n}$ alternate choice routes. If x_{j_1} is blocked, x_{j_2} can be selected. This alternate routing eventually leads to x_{j_n}. If final link x_{j_n} is blocked, then the call is lost because no other link is available at node i for the call to overflow.

Definition 5.5 — A pair of nodes i and j are fused (merged, coalesced [64]) if the two nodes are replaced by a single new node such that all branches that were incident on either i or j or both are incident on the new node. Figures 5.4a — c explain the concept of merging of nodes.

Definition 5.6 — Terminal unreliability (node to node network blocking) is the totality of conditions under which a call at source node is not allowed to reach terminal node.

5.2.1.1 Preliminaries — The recursive technique for computing system unreliability uses the following propositions P0 — P3 for its development.

P0 (concept of $S_i(.)$) — $S_i(.)$ represents call (signal) at node

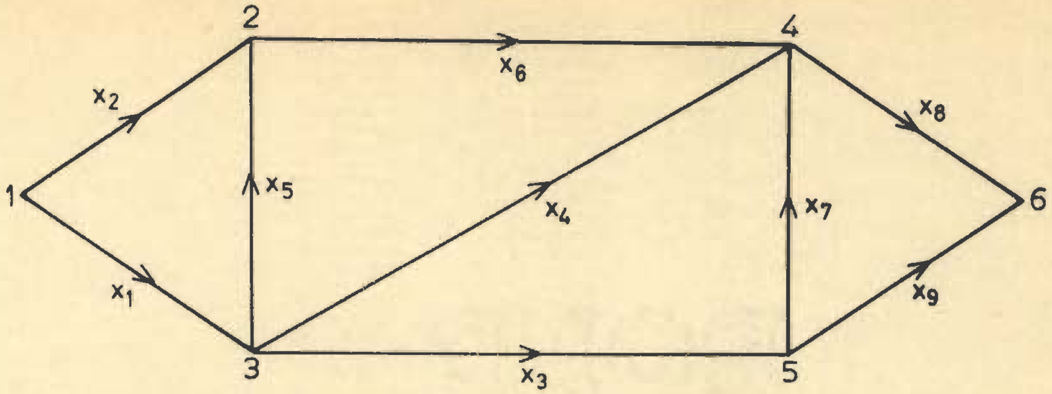


Figure 5.3. Modified ARPA Network.

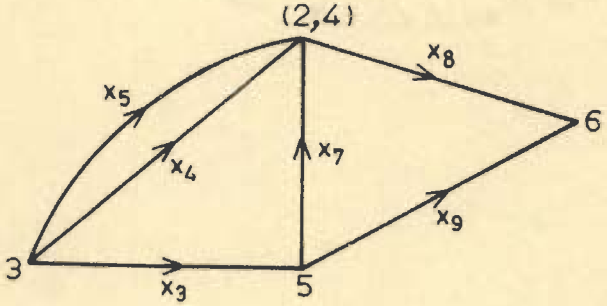


Figure 5.4 (a). Coalescing Nodes 2 & 4.

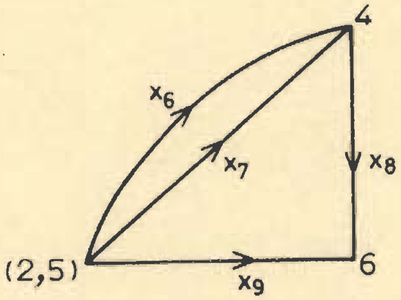


Figure 5.4 (b). Coalescing Nodes 2 & 5.

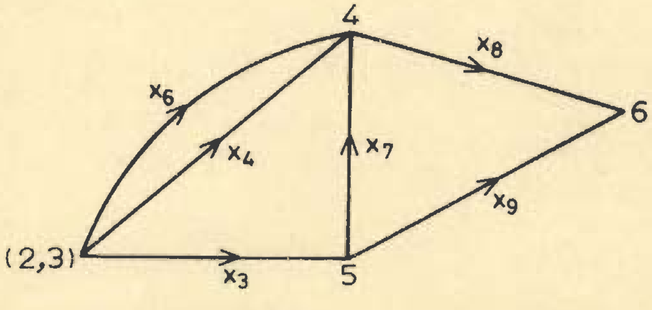


Figure 5.4 (c). Coalescing Nodes 2 & 3.

i overflowing onto various available alternate links of i. The terms in the argument (.) describe.

i. the outgoing branches from node i and

ii. the ordering of sequence of routes for the overflowing call.

For figure 5.1, $S_i(x_{j_1},\dots,x_{j_n})$ conveys:A signal can leave node i onto n alternate routes; x_{j_1} being the first choice and so on until final link x_{j_n} is reached.

P1 (call flowing methodology) – Let node i be connected with node j_1 through x_{j_1}.

If x_{j_1} is busy, then the call remains at i. The call, then, overflows onto remaining available alternate links of node i. This situation is described by

$$\bar{X}_{j_1}\, S_i(x_{j_2},\dots,x_{j_n}) \tag{5.4}$$

Else, the call overflows onto various alternate links of both nodes i and j_1. This situation is characterized as follows –

$$X_{j_1}\, S_i(x_{j_2},\dots,x_{j_n})\, S_{j_1}(.) \tag{5.5}$$

The totality of conditions for P1 is, then, given as Boolean sum of relations (5.4) and (5.5). Thus,

$$S_i(x_{j_1},\dots,x_{j_n}) = \bar{X}_{j_1} S_i(x_{j_2},\dots,x_{j_n}) + X_{j_1} S_i(x_{j_2},\dots,x_{j_n}) S_{j_1}(.) \tag{5.6}$$

The $S_i(x_{j_2},\dots,x_{j_n})$ in equation (5.6) is obtainable from $S_i(x_{j_1},\dots,x_{j_n})$ by deleting branch x_{j_1} from the argument list of later.

P2 (call flowing constraint) – $S_i(\emptyset)$ represents call overflowing from the final link of node i. In this event no alternate links exists at i for calls to spill. Thus

$$S_i(\emptyset) = \begin{cases} 0, \text{if } i \text{ is the terminal node} \\ 1, \text{otherwise} \end{cases}$$

P3 (node merging and $S_i(.)$) – The AND operation on $S_i(.)$'s is conceptually same is if different nodes are fused together. The links internally connecting the nodes before fusion form selfloops and are deleted.

$$S_j(.)\, S_p(.) \ldots S_r(.) \Rightarrow S_{jp\ldots r}(.)$$

Nodes j, p, ..., r have been fused to form a single node jp...r. For the fusion of any number of nodes with terminal node t, we have

$$S_{\ldots t}(.) = 0$$

Theorem 5.1 follows from P0 through P3.

Theorem 5.1 – Let the outdegree of node i be n, then

$$\begin{aligned} S_i(x_{j_1}, \ldots, x_{j_n}) = {} & \bar{X}_{j_1}\, S_i(x_{j_2}, \ldots, x_{j_n}) + \ldots \\ & + X_{j_1} \ldots \bar{X}_{j_r} S_i(x_{j_{r+1}}, \ldots, x_{j_n}) S_{j_1 \ldots j_{r-1}}(.) \\ & + X_{j_1} \ldots \bar{X}_{j_n} S_{j_1 j_2 \ldots j_{n-1}}(.) + X_{j_1} \ldots X_{j_n} S_{j_1 \ldots j_n}(.) \end{aligned} \tag{5.7}$$

where $j_1, \ldots, j_n$ are the respective nodes connected by $x_{j_1}, \ldots, x_{j_n}$ as shown in figure 5.1.

Proof. Consider only two branches x_{j_1}, x_{j_2} leaving node i. Use the result of P1 repeatedly for i, we have –

$$S_i(x_{j_1},x_{j_2}) = \bar{X}_{j_1} S_i(x_{j_2}) + X_{j_1} S_{j_1}(.)S_i(x_{j_2}) \qquad (5.8)$$

$$S_i(x_{j_2}) = \bar{X}_{j_2} S_i(\emptyset) + X_{j_2} S_{j_2}(.)S_i(\emptyset) \qquad (5.9)$$

Note that $S_i(\emptyset) = 1$. Substitute (5.9) in (5.8) and apply P3. The resultant expression is :

$$S_i(x_{j_1},x_{j_2}) = \bar{X}_{j_1} S_i(x_{j_2}) + X_{j_1}\bar{X}_{j_2} S_{j_1}(.) + X_{j_1}X_{j_2}S_{j_1 j_2}(.) \qquad (5.10)$$

Equation (5.10) and result of P1 can, then, be used to prove the theorem.

□

Equation (5.7) can be read as follows: Blocking occurs if: 1) link x_{j_1} is busy, or if 2) link x_{j_1} is not busy and x_{j_2} is busy, or 3)...,so on.

Each factor in (5.7) is composed of:

a. the status of various links and is defined by X's

b. the associated other term representing the graph modifier and defined by S's.

The terms in theorem 5.1 of type a are expanded as:

$$\bar{X}_{j_1},\ X_{j_1}\bar{X}_{j_2},\ldots,X_{j_1}\ldots\bar{X}_{j_r},\ldots,X_{j_1}\ldots\bar{X}_{j_n},\ X_{j_1}\ldots X_{j_n}$$

with Boolean values of

0

10

⋮

11...0 (5.11)

⋮

111..1

Fratta and Montanari [36] have termed this decomposition technique: Negative Conservative Policy. For negative type, the logical expansion contains exactly one 0 and $(i-1)$ 1's for all $i \leq n$. In the event, $i = n+1$, the expansion results into 111...1 combination. Note that only $(n+1)$ EMD (exclusive and mutually disjoint) events are generated in this policy and hence the name conservative. The exhaustive policy, on the other hand, obtains 2^n EMD terms starting from all 0's to all 1's. The method given in [115] utilizes exhaustive policy as the basis to compute network reliability. Obviously, the proposed algorithm that makes use of conservative policy is economical in comparison to [115].

Each Boolean pattern given in (5.11) is, in turn used to obtain the associated coefficient of type b by using the Graph Reduction Algorithm (GRA) described below.

Algorithm (GRA) –

GRA – 0. [Initialize] Consider input pattern as 11...0 (where $r = |11...0|$ ie., the cardinality of input pattern).

GRA – 1. Branches not falling in the category of 0 or 1 branches are variable branches. They have the weights assigned to them as defined by their name variable. A branch x_{j_r} is termed as 0(1)-branch if $x_{j_r} = 0(1)$.

GRA – 2a. From amongst the list of n branches leaving node i delete 0 – and 1 – branches. Deletion of these branches defines a term $S_i(x_{j_{r+1}},\ldots,x_{j_n})$.

b. Coalesce all (r –1) nodes $j_1,\ldots,j_{r-1}$ connected by 1 – branches and generate a term like $S_{j_1\ldots j_{r-1}}(.)$.

c. Logical multiplication relates the two terms obtained in a and b above.

Example 5.2 – Substitute $S_i(x_{j_1},x_{j_2},x_{j_3},x_{j_4})$ for $X_{j_1}X_{j_2}\bar{X}_{j_3}$ (110). As mentioned in GRA – 1 x_{j_1},x_{j_2} are 1 –branches, x_{j_3} 0 –branch and x_{j_4} is a variable branch. Use GRA –2; we have:

a. $S_i(x_{j_4})$ is obtained by deleting all 0 – and 1 –branches from $S_i(x_{j_1},x_{j_2},x_{j_3},x_{j_4})$,

b. fusion of 1 –branches, this generates a term $S_{j_1j_2}(.)$, the actual arguments are obtainable from the graph.

The graph modifier coefficient associated with the 110 bit pattern is $S_i(x_{j_4})S_{j_1j_2}(.)$. Using theorem 5.1, one can derive following result for n = 4.

$$
\begin{aligned}
S_i(x_{j_1},x_{j_2},x_{j_3},x_{j_4}) = {} & \bar{X}_{j_1} S_i(x_{j_2},x_{j_3},x_{j_4}) \\
& + X_{j_1}\bar{X}_{j_2} S_{j_1}(.) S_i(x_{j_3},x_{j_4}) \\
& + X_{j_1}X_{j_2}\bar{X}_{j_3} S_{j_1 j_2}(.) S_i(x_{j_4}) \\
& + X_{j_1}X_{j_2}X_{j_3}\bar{X}_{j_4} S_{j_1 j_2 j_3}(.) \\
& + X_{j_1}X_{j_2}X_{j_3}X_{j_4} S_{j_1 j_2 j_3 j_4}(.).
\end{aligned}
$$

Theorem 5.2 – For a series connection (see figure 5.2), the logical expression for node to node network blocking is

$$S_1(x_1) = \bar{X}_1 + X_1\bar{X}_2 + \ldots + X_1X_2\ldots\bar{X}_{k-1}$$

Proof. For k=2, branches x_1 and x_2 constitute the series connection (see figure 5.2). Using P1, we have –

$$S_1(x_1) = \bar{X}_1\ S_1(\emptyset) + X_1\ S_2(.),$$

$$S_2(.) = \bar{X}_2\ S_2(\emptyset) + X_2\ S_3(\emptyset).$$

From P2, $S_1(\emptyset) = S_2(\emptyset) = 1$ while $S_3(\emptyset) = 0$. Thus –

$$B_G(K) = \bar{X}_1 + X_1\ \bar{X}_2$$

By iteration, the theorem is proved.

□

5.2.1.2. Algorithm –

A1a. Start from the source node.

b. The system unreliability $B_G(K)$ is given as

$$B_G(K) = S_1(x_1,x_2,\ldots)\Big|_{X_i(\bar{X}_i)\,\to\,p_i(q_i)}$$

A2. Expand $S_1(x_1,x_2,\ldots)$ using theorem 5.1. If the out-degree of node 1 is n_1, the (n_1+1) factors are generated at the first instance.

A3. Apply theorem 5.1 again on factors obtained and while doing this successively proceed towards the terminal node. Repeat A3 until the terminal node is reached.

A4. Substitute –

$$S_{\ldots t}(\cdot) = 0, \text{and}$$

$$S_i(\emptyset) = \begin{cases} 0, \text{if } i = t \\ 1, \text{otherwise} \end{cases}$$

where...t represents the fusion of any number of nodes with terminal node t. These results have been defined in P2 and P3.

Example 5.3 – The algorithm is used for deriving the unreliability expression for figure 5.3, which is the representation of modified ARPA network [21]. In this graph all the nodes are assumed to be perfect. Applying A1 for the source node 1, logical expression for node to node blocking is

$$S_1(x_1,x_2) = \bar{X}_1\underbrace{S_1(x_2)}_{\text{factor \# 1}} + X_1\bar{X}_2\underbrace{S_3(\cdot)}_{\text{factor \# 2}} + X_1X_2\underbrace{S_{23}(\cdot)}_{\text{factor \# 3}} \qquad (5.12)$$

Table 5.3a through 5.3c show the final expression for various factors. Note that, the generation of these expressions is following recursion.

Substitute the expressions in table 5.3a through c in equation (5.12) and apply step Alb. of the algorithm, the node to node blocking is

$$\begin{aligned} B_G(K) = {} & q_1 q_2 + q_1 p_2 q_6 + q_1 p_2 p_6 q_8 + p_1 q_2 q_3 q_4 q_5 + p_1 p_2 q_3 q_4 q_6 \\ & + p_1 q_2 q_3 q_4 p_5 q_6 + p_1 q_2 q_3 p_4 q_5 q_8 + p_1 q_2 q_3 p_4 p_5 q_8 + p_1 p_2 q_3 q_4 p_6 q_8 \\ & + p_1 p_2 q_3 p_4 q_6 q_8 + p_1 p_2 q_3 p_4 p_6 q_8 + p_1 q_2 q_3 q_4 p_5 p_6 q_8 \\ & + p_1 q_2 p_3 q_4 q_5 q_7 q_9 + p_1 p_2 p_3 q_4 q_6 q_7 q_9 + p_1 p_2 p_3 q_4 p_6 q_8 q_9 \\ & + p_1 p_2 p_3 p_4 q_6 q_8 q_9 + p_1 q_2 p_3 p_4 q_5 q_8 q_9 + p_1 q_2 p_3 p_4 p_5 q_8 q_9 \\ & + p_1 q_2 p_3 q_4 p_5 q_6 q_7 q_9 + p_1 p_2 p_3 p_4 p_6 q_8 q_9 + p_1 q_2 p_3 q_4 q_5 p_7 q_8 q_9 \\ & + p_1 p_2 p_3 q_4 q_6 p_7 q_8 q_9 + p_1 q_2 p_3 q_4 p_5 p_6 q_7 q_8 q_9 + p_1 q_2 p_3 q_4 p_5 p_6 p_7 q_8 q_9 \\ & + p_1 q_2 p_3 q_4 p_5 q_6 p_7 q_8 q_9 \end{aligned}$$

5.2.1.3 Discussion –

a. The minimal cutset in the network is obtained by writing Boolean expression consisting of only complemented variables from each of the term of $B_G(K)$ and simplifying it using Boolean relation $Y+YZ \equiv Y$, where Y and Z are Boolean functions. The following derivation is suggested

$$\begin{aligned} & \overline{X}_1\overline{X}_2 + \overline{X}_1\overline{X}_6 + \overline{X}_1\overline{X}_8 + \overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_5 + \overline{X}_3\overline{X}_4\overline{X}_6 + \cancel{\overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_6} + \cancel{\overline{X}_2\overline{X}_3\overline{X}_5\overline{X}_8} + \\ & \cancel{\overline{X}_2\overline{X}_3\overline{X}_8} + \cancel{\overline{X}_3\overline{X}_4\overline{X}_8} + \cancel{\overline{X}_3\overline{X}_6\overline{X}_8} + \overline{X}_3\overline{X}_8 + \cancel{\overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_8} + \overline{X}_2\overline{X}_4\overline{X}_5\overline{X}_7\overline{X}_9 + \\ & \overline{X}_4\overline{X}_6\overline{X}_7\overline{X}_9 + \cancel{\overline{X}_4\overline{X}_8\overline{X}_9} + \cancel{\overline{X}_6\overline{X}_8\overline{X}_9} + \cancel{\overline{X}_2\overline{X}_5\overline{X}_8\overline{X}_9} + \cancel{\overline{X}_2\overline{X}_8\overline{X}_9} + \cancel{\overline{X}_2\overline{X}_4\overline{X}_6\overline{X}_7\overline{X}_9} + \\ & \overline{X}_8\overline{X}_9 + \cancel{\overline{X}_2\overline{X}_4\overline{X}_5\overline{X}_8\overline{X}_9} + \cancel{\overline{X}_4\overline{X}_6\overline{X}_7\overline{X}_8\overline{X}_9} + \cancel{\overline{X}_2\overline{X}_4\overline{X}_7\overline{X}_8\overline{X}_9} + \cancel{\overline{X}_2\overline{X}_4\overline{X}_8\overline{X}_9} + \cancel{\overline{X}_2\overline{X}_4\overline{X}_6\overline{X}_8\overline{X}_9} \end{aligned}$$

Table 5.3a Generation of expression for factor # 1

Expression(s)		Remarks
$S_1(x_2)$	$= \bar{X}_2S_1 + X_2S_2(1)$	$S_1=S_2=S_4=1$
$S_2(.)$	$= \bar{X}_6S_2 + X_6S_4(.)$	and
$S_4(.)$	$= \bar{X}_8S_4 + X_8S_6(.)$	$S_6(.)=0$
$S_1(x_2)$	$= \bar{X}_2 + X_2\bar{X}_6 + X_2X_6\bar{X}_8$	

Table 5.3b Generation of expression for factor # 2

Expression(s)		Remarks
$S_3(.)$	$= \bar{X}_3S_3(x_4,x_5)+X_3\bar{X}_4X_5(.)S_3(x_5)$	$S_3=S_5=S_{24}=1$
	$+ X_3X_4\bar{X}_5S_{45}(.) + X_3X_4X_5S_{245}(.)$	
$S_3(x_4,x_5)$	$= \bar{X}_4S_3(x_5)+X_4\bar{X}_5S_4(.)+X_4X_5S_{24}(.)$	
$S_3(x_5)$	$= \bar{X}_5S_3+X_5 S_2(.)$	
$S_{24}(.)$	$= \bar{X}_8S_{24}+X_8S_6(.)$	$S_{45}=S_{25}=1$ and
$S_5(.)S_3(x_5)$	$= S_5(.)[\bar{X}_5S_3+X_5S_2(.)]$	$S_6(.)=S_{46}(.)=0$
	$= \bar{X}_5S_5(.)+X_5S_{25}(.)$	
$S_5(.)$	$= \bar{X}_9S_5(x_7)+X_9\bar{X}_7S_6(.)+X_9X_7S_{46}(.)$	
$S_5(x_7)$	$= \bar{X}_7S_5+X_7S_4(.)$	
$S_{25}(.)$	$= \bar{X}_9S_{25}(x_6,x_7)+X_9\bar{X}_7S_6(.)S_{25}(x_6)$	
	$+ X_9X_7\bar{X}_6S_{46}(.)+X_9X_7X_6S_{46}(.)$	
$S_{25}(x_6,x_7)$	$= \bar{X}_7S_{25}(x_6)+X_7\bar{X}_6S_4(.)+X_7X_6S_4(.)$	
$S_{25}(x_6)$	$=\bar{X}_6S_{25}+X_6S_4(.)$	
$S_{45}(.)$	$=\bar{X}_9S_{45}(x_8)+X_9S_6(.)$	
$S_{45}(x_8)$	$=\bar{X}_8S_{45}+X_8S_6(.)$	
$S_{245}(.)$	$=\bar{X}_9S_{45}(x_8) + X_9S_6(.)$	

Table 5.3b contd.

Expression(s)	
$S_3(.)$	$= \bar{X}_3\bar{X}_4\bar{X}_5 + \bar{X}_3\bar{X}_4X_5\bar{X}_6 + \bar{X}_3\bar{X}_4X_5X_6\bar{X}_8 + \bar{X}_3X_4\bar{X}_5\bar{X}_8$
	$+ \bar{X}_3X_4X_5\bar{X}_8 + X_3\bar{X}_4\bar{X}_5\bar{X}_7\bar{X}_9 + X_3\bar{X}_4\bar{X}_5X_7\bar{X}_8\bar{X}_9 + X_3\bar{X}_4X_5\bar{X}_6\bar{X}_7\bar{X}_9$
	$+ X_3\bar{X}_4X_5X_6\bar{X}_7\bar{X}_8\bar{X}_9 + X_3\bar{X}_4X_5\bar{X}_6X_7\bar{X}_8\bar{X}_9 + X_3\bar{X}_4X_5X_6X_7\bar{X}_8\bar{X}_9$
	$+ X_3X_4\bar{X}_5\bar{X}_8\bar{X}_9 + X_3X_4X_5\bar{X}_8\bar{X}_9$

Table 5.3c Generation of expression for factor # 3

Expression(s)		Remarks
$S_{23}(.)$	$= \bar{X}_3S_{23}(x_4,x_6) + X_3\bar{X}_4S_5(.)S_{23}(x_6)$	$S_{23}=1$
	$+ X_3X_4\bar{X}_6S_{45}(.) + X_3X_4X_6S_{45}(.)$	
$S_{23}(x_4,x_6)$	$= \bar{X}_4S_{23}(x_6) \quad + X_4\bar{X}_6S_4(.) + X_4X_6S_4(.)$	
$S_{23}(x_6)$	$= \bar{X}_6S_{23} + X_6S_4(.)$	
$S_5(.)S_{23}(x_6)$	$= \bar{X}_6S_5(.) + X_6S_{45}(.)$	
$S_{23}(.)$	$= \bar{X}_3\bar{X}_4\bar{X}_6 + \bar{X}_3\bar{X}_4X_6\bar{X}_8 + \bar{X}_3X_4\bar{X}_6\bar{X}_8 + \bar{X}_3X_4X_6\bar{X}_8$	
	$+ X_3\bar{X}_4\bar{X}_6\bar{X}_7\bar{X}_9 + X_3\bar{X}_4\bar{X}_6X_7\bar{X}_8\bar{X}_9 + X_3\bar{X}_4X_6\bar{X}_8\bar{X}_9$	
	$+ X_3X_4\bar{X}_6\bar{X}_8\bar{X}_9 + X_3X_4X_6\bar{X}_8\bar{X}_9$	

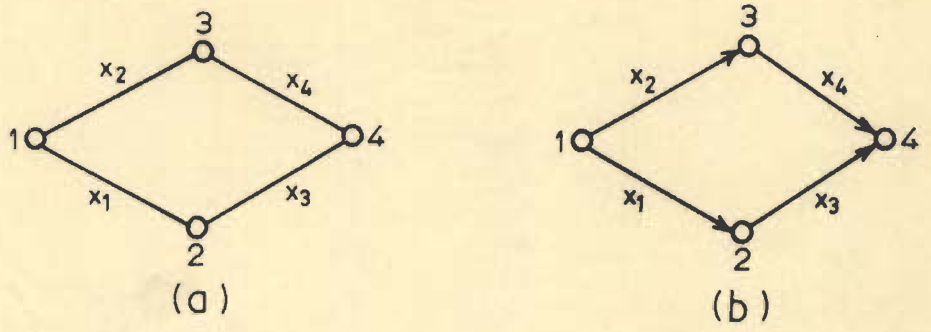


Figure 5.5. Illustrating (a) HMW and (b) Proposed Method.

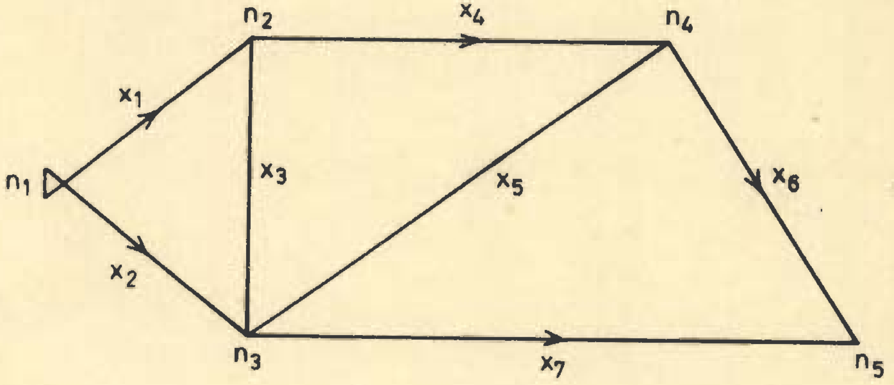


Figure 5.6. An Example.

After cancelling terms based on the Boolean expression, and interchanging X_i to x_i, the minimal cutsets are $x_1\ x_2$, $x_1\ x_6$ $x_1\ x_8$, $x_2\ x_3\ x_4\ x_5$, $x_3x_4x_6$, $x_3\ x_8$, $x_2\ x_4\ x_5\ x_7\ x_9$, $x_4\ x_6\ x_7\ x_9$ $x_8\ x_9$.

b. Blocking functions of a network have been expressed assuming nodes to be perfect. This is, however, not limitation of method and if nodes are not perfect these relations can be modified using modified form by replacing X_i by $N_i\ X_i$ and $\overline{X}_i$ by $\overline{N}_i + N_i\ \overline{\overline{X}}_i$ if branch x_i is incident from node i [115].

c. The algorithm also applies to mixed graphs having undirected branches. In that event, the undirected branch should be replaced by two directed branches in anti-parallel. The application of the algorithm to mixed graphs having undirected branches is described in section 5.2.2.

5.2.1.4 Comparison With HMW Method –

Example 5.4 – In this example, we compare the HMW method [70] with the proposed technique described in section 5.2.1.2. An example is given first. Figure 5.5a applies to the HMW method whereas figure 5.5b applies to the proposed method. Node 1 is the source and node 4 is the terminal. The two figures should yield the same reliability results. A discussion follows the example.

HMW method – Apply the steps given in [70] for figure 5.5a and also use the same notations mentioned therein. We have –

Step 1 $N = \{2, 3, 4\}$

$C = \{x_3, x_4\}$

$F_1 = \emptyset$

$S_1 = \{x_1, x_2\}$

$B_1 = 11$

$i = 1$

Step 2 $T_1 = \{x_1, x_2\}$

$M_1 = \{2, 3\}$

$N = \{4\}$

$F_2 = \{x_3, x_4\}$

$S_2 = \emptyset$

$C = \emptyset$

Step 3 $T_2 = \emptyset$

$CS = x_3x_4\bar{x}_1\bar{x}_2$ (result)

Step 4 $C = \{x_3, x_4\}$

$N = \{2, 3, 4\}$

$B_1 = 10$

Step 2 $T_1 = \{x_1\}$

$M_1 = \{2\}$

$N = \{3, 4\}$

$F_2 = \{x_3\}$

$S_2 = \emptyset$

$C = \{x_4\}$

Step 3 $T_2 = \emptyset$

$CS = \bar{x}_1x_2x_3$ (result)

Step 4 $C = \{x_3, x_4\}$

$N = \{2, 3, 4\}$

$B_1 = 01$

Step 2 $T_1 = \{x_2\}$

$M_1 = \{3\}$

$N = \{2, 4\}$

$F_2 = \{x_4\}$

$S_2 = \emptyset$

$C = \{x_3\}$

Step 3 $T_2 = \emptyset$

$CS = \bar{x}_2 \; x_1 \; x_4$ (result)

Step 4 $C = \{x_3, x_4\}$

$N = \{2, 3, 4\}$

$B_1 = 00$

Step 2 $T_1 = \emptyset$

$M_1 = \emptyset$

$N = \{2, 3, 4\}$

$F_2 = \emptyset$

$S_2 = \emptyset$

$C = \{x_3, x_4\}$

Step 3 $T_2 = \emptyset$

$CS = x_1 \; x_2$ (result)

Step 4 $C = \{x_3, x_4\}$

$N = \{2, 3, 4\}$

$B_1 = -1$

Step 5 $i = 0$

terminated

Hansler, McAuliffe and Wilkov [70] use the bar to denote success whereas the proposed method uses the bar to denote failure. Therefore, to be consistent with the proposed method, the above results are converted to

$X_1 X_2 \bar{X}_3 \bar{X}_4$, $X_1 \bar{X}_2 \bar{X}_3$, $\bar{X}_1 X_2 \bar{X}_4$, $\bar{X}_1 \bar{X}_2$

Using the proposed method — For figure 5.5b, we have —

$$S_1(x_1,x_2) = \bar{X}_1 S_1(x_2) + X_1 \bar{X}_2 S_2(.) + X_1 X_2 S_{23}(.)$$

$$S_1(x_2) = \bar{X}_2 S_1 + X_2 S_3(.)$$

$$S_3(.) = \bar{X}_4 S_3 + X_4 S_4(.)$$

$$S_2(.) = \bar{X}_3 S_2 + X_3 S_4(.)$$

$$S_{23}(.) = \bar{X}_3 S_{23}(x_4) + X_3 \bar{X}_4 S_4(.) + X_3 X_4 S_4(.)$$

$$S_{23}(x_4) = \bar{X}_4 S_3 + X_4 S_4(.)$$

$$S_4(.) = 0$$

$$S_1 = S_2 = S_3 = 1$$

Therefore $S_1(x_1,x_2)$ simplifies to

$\bar{X}_1(\bar{X}_2 + X_2 \bar{X}_4) + X_1\bar{X}_2\bar{X}_3 + X_1X_2\bar{X}_3\bar{X}_4$; and finally to

$\bar{X}_1\bar{X}_2 + \bar{X}_1X_2\bar{X}_4 + X_1\bar{X}_2\bar{X}_3 + X_1X_2X_3\bar{X}_4$

Thus, the results are

$\bar{X}_1\bar{X}_2$, $\bar{X}_1X_2\bar{X}_4$, $X_1\bar{X}_2\bar{X}_3$, $X_1X_2\bar{X}_3\bar{X}_4$

Discussion –

1) The two methods give identical results and identical terms in the example.

2) The HMW method is for undirected graphs whereas the proposed method is for directed graphs (Potential exists for modifying the HMW method for directed graphs).

3) The HMW method uses the concept of level (i) and uses binary subtraction (B_i) to obtain the various combinations of branch failure/ success at each level. The proposed method uses the concept of keystone branch and 'Negative Conservative Policy'.

4) The proposed method uses the concept of coalescence. A variant of this concept is implicitly used in the HMW method, although not stated as coalescence. It consists of the formation of the set S_{i+1} which with minor exception can be viewed as the set of branches leaving the coalesced node of the nodes reached in the previous level.

5) The proposed method is quite simple.

5.2.2 Algebraic Approach – The technique presented in this section is quite useful for mixed graphs ie., graphs having both directed and undirected branches.

5.2.2.1 Procedure Development – The following discussion helps understand the algorithm presented later in this section.

System success determinant – This is obtained from knowledge of the connection matrix [C] for the logic diagram of the network. To define |S|, use the following steps [21]:

a1. Use a terminal numbering convention to name the nodes and branches of the network. Note that n_1(n_k) represents the source (sink) node.

 2. Write [C] for the logic graph of the network.

b. Add a diagonal unity matrix [U] of size k x k to the connection matrix [C].

c1. Remove the column corresponding to source node and the row corresponding to sink node in the matrix generated at step b.

 2. Take remaining rows and columns and define the system success determinant |S| of size (k – 1). In this sub-step all the algebraic variables (x_i's) are changed to the corresponding Boolean variables (X_i's).

The system success determinant |S|, if expanded using Boolean sum and product operations, results into path set for the network [21].

Example 5.5 – For the modified ARPA network shown in figure 2.2, the connection matrix is

$$[C] = \begin{array}{c} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \end{array} \begin{bmatrix} 0 & x_1 & x_2 & 0 & 0 & 0 \\ 0 & 0 & x_3 & x_4 & 0 & 0 \\ 0 & x_3 & 0 & x_5 & x_6 & 0 \\ 0 & 0 & x_5 & 0 & x_7 & x_8 \\ 0 & 0 & 0 & x_7 & 0 & x_9 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Using steps a through c, $|S|$ is given as

$$|S| = \left|\begin{array}{ccccc} X_1 & X_2 & 0 & 0 & 0 \\ 1 & X_3 & X_4 & 0 & 0 \\ X_3 & 1 & X_5 & X_6 & 0 \\ 0 & X_5 & 1 & X_7 & X_8 \\ 0 & 0 & X_7 & 1 & X_9 \end{array}\right| \begin{array}{l} \text{row 1} \\ \text{row 2} \\ \text{row 3} \\ \text{row 4} \\ \text{row 5} \end{array} \qquad (5.13)$$

For the case where node n_1 (node n_6) is a source (sink) node, the 2-D array A is same as $|S|$. In the event node $i(i \neq 1)$ is assumed a source node, A is derived from $|S|$ by simply interchanging row i with row 1. To avoid this interchange, we have used terminal numbering convention in step a1.

JNR operation – It is used to join any number of rows in the 2 – D array A. The following procedure performs JNR operation. Assume '+' operation representing Boolean ORing.

Input. Source node and node(s) r participating in JNR operation.

```
JNR.  i ← source node, j ← r ;
      row (i) ← row (i) + row (j);
      delete row (j);
      accept another j, if exists; Repeat the procedure
      and.
```

Example 5.6 – On performing JNR operation over row-1, row-2 and row-3 of (5.13) the result is

$$\begin{pmatrix} 1 & 1 & X_4+X_5 & X_6 & 0 \\ 0 & X_5 & 1 & X_7 & X_8 \\ 0 & 0 & X_7 & 1 & X_9 \end{pmatrix} \qquad (5.14)$$

Here, Boolean identity 1+Anything $\equiv$ 1 is used to simplify the elements in positions (columns) 1 and 2. Note that the size of A is reduced when A is subjected to JNR operation.

0 – SUB Operation – This takes care of a 0 –branch for which a 0 is substituted in A. As an example, consider $X_1 = 0$ in 5.13. Thus x_1 represents a 0 –branch and 0 –SUBstitute operation modifies (5.13) as under

$$\begin{pmatrix} 0 & X_2 & 0 & 0 & 0 \\ 1 & X_3 & X_4 & 0 & 0 \\ X_3 & 1 & X_5 & X_6 & 0 \\ 0 & X_5 & 1 & X_7 & X_8 \\ 0 & 0 & X_7 & 1 & X_9 \end{pmatrix}$$

Note that 0 –SUB operation is a unary operation and is equivalent to an edge being removed from a graph [118].

1 –SUB Operation – It is performed for 1–branches and the 2 –D array A is modified using following rules.

a) Replace 1–branch (es) by zero

b) In the column (s) corresponding to 1–branch (es) obtain row(s) containing an entry 1. Use JNR operation for row 1 with that (those) obtained above.

c) Remove column(s) corresponding to 1–branch(es).

As an example assume x_1 and x_2 to be 1–branches. Replace X_1 and X_2 by zero. Two rows namely row 2 and row 3 are obtained as they contain an entry 1 in the columns corresponding to X_1 and X_2. As stated earlier, JNR operation results into equation (5.14). After deleting the columns corresponding to 1-branches the modified 2-D array A_M will be as follows

$$\begin{pmatrix} X_4+X_5 & X_6 & 0 \\ 1 & X_7 & X_8 \\ X_7 & 1 & X_9 \end{pmatrix}$$

The 1-SUB operation, discussed above, applies directly for OOC and system reliability analysis. Under SOC scheme, rule a) is modified as under.

a1. Replace 1-branch(es) by zero.

 2. All other variable branches in the row corresponding to 1-branch are also replaced by zero.

Rule b) and c), however, remain unchanged. In arbitrary routing plan (OOC with spill), the branch(es) incident on node(s) assuming the originating office control is(are) also subjected to modified version of rule a) for its (their) 1-SUB operation, while the remaining branches being treated similarly to OOC case. Thus, there exists variation according to call routing procedure only in 1-Sub operation. Such variation is not allowed in 0-SUB operation.

Note that 1-SUB operation is an unary operation and reduces the size of the array. For p number of 1-branches the 1-SUB operation generates a modified array of size $(n-p)x(n-p)$ where n x n be the initial array size. An equivalent to 1-SUB operation is 'contraction' and is reported in [118].

5.2.2.2 Algorithm EBRA [End to end blocking and reliability analysis] –

Step 0. [Input] connection matrix, alternate choice links for a given routing strategy (optional for reliability analysis).

Step 1. Start with modified connection matrix and obtain the 2 –D array A as explained earlier.

Step 2. (a) Set level $i \leftarrow 0$

(b) Let $(X_1, X_2 \ldots X_n)$ be the nonzero choice route enteries in the first row.

(c) Use conservative logical expansion 1,01,001,...,000...1,00...0 [16]. The first term corresponds to $X_1=1$ followed by $X_1=0$, $X_2=1$ and so on. Note that there are in all $(n+1)$ terms.

(d) For any rth term $(1 \leq r < n+1)$ there are exactly $(r-1)$ 0 – branches (one 1–branch) for which a branch variable in that term is assigned a value 0(1). In the event $r = n+1$, all the n –branches are assigned a logical value 0.

(e) Consider an rth term and perform 0 –SUB and appropriate 1–SUB operations to obtain the modified 2 –D array. In case an 1 does not exist in any of the row(s) beneath the 1–branch column(s), the resultant is a null set ($\emptyset$).

(f) Repeat (d) and (e) for all r.

Step 3. Set level $i \leftarrow i+1$ and return to step 2(b).

Repeat the above procedure until one obtains pendant vertex of the type (i) 1(ii) 0(iii) a branch variable or sum of branch variables and (iv) null set (ϕ).

Step 4. (a) Traversing the tree from root to pendant vertex of types (ii) and (iii) in step 3, we obtain a logical expression for end to end blocking or unreliability $B_G(K)$ of the network G.

(b) The term of types (ii) and (iii) at pendant vertex is complemented.

(c) To determine the symbolic expression of end to end network blocking and also unreliability replace all Boolean term with corresponding probabilities as

$$X_i\ (\overline{X}_i) \rightarrow p_i(q_i)$$

The algorithm, if modified, generates the terminal reliability of the network. The necessary modification is proposed as under.

1. Use a logical expansion of the type 1,01,001,...,000..01 only for step 2(c). Note that there are in all n terms in it.

2a. Traversing the tree from root to pendant vertex of type i) and iii) in step 3, we obtain a logical expression for terminal reliability.

b. The sub-step 4(b) is not needed and therefore be deleted.

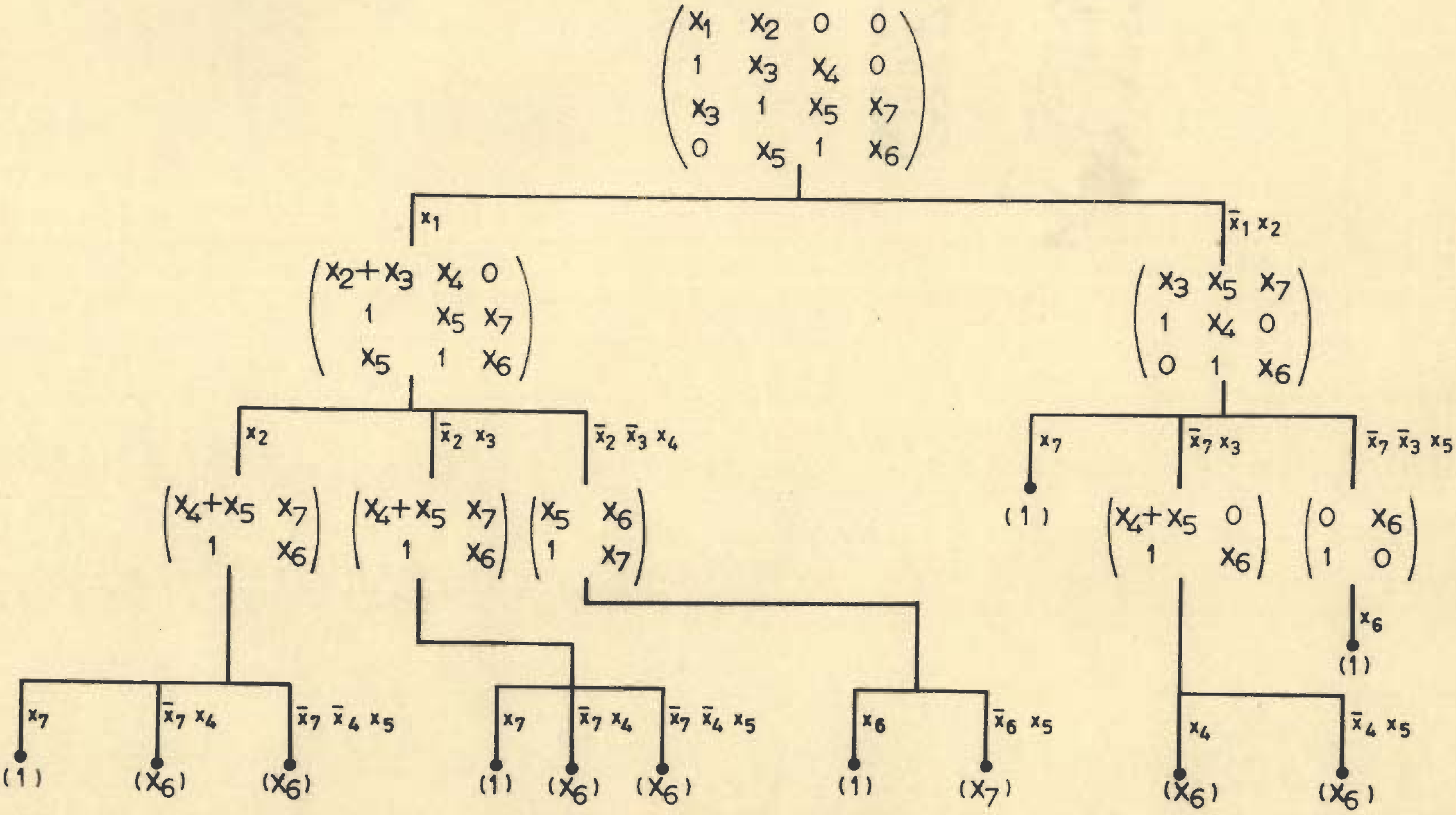


Figure 5.7. Steps for Terminal Reliability.

Example 5.7 – Figure 5.7 illustrates the modified algorithm for computing terminal reliability for figure 5.6. Note that there are (k –2) levels in the tree thus generated. A branch x_j(in row –1) is preferred for expansion if the column corresponding to it does not contain an entry of 1. In that event the pendant vertex will have a value of 1(refer detailed steps shown in figure 5.7).

Example 5.8 – For figure 5.6 let n_1 be source node and n_5 be the sink node. In this network modified connection matrix is given by

$$\begin{bmatrix} 1 & x_1 & x_2 & 0 & 0 \\ 0 & 1 & x_3 & x_4 & 0 \\ 0 & x_3 & 1 & x_5 & x_7 \\ 0 & 0 & x_5 & 1 & x_6 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 1. Using principles outlined above, A is obtained as

$$\begin{pmatrix} X_1 & X_2 & 0 & 0 \\ 1 & X_3 & X_4 & 0 \\ X_3 & 1 & X_5 & X_7 \\ 0 & X_5 & 1 & X_6 \end{pmatrix}$$

Step 2. The first row contains two non –zero entries as X_1, X_2. Set $i \leftarrow 0$, and expand x_1 and x_2 using conservative policy as shown in figure 5.8. Perform 0 –SUB and 1 –SUB operations over A.

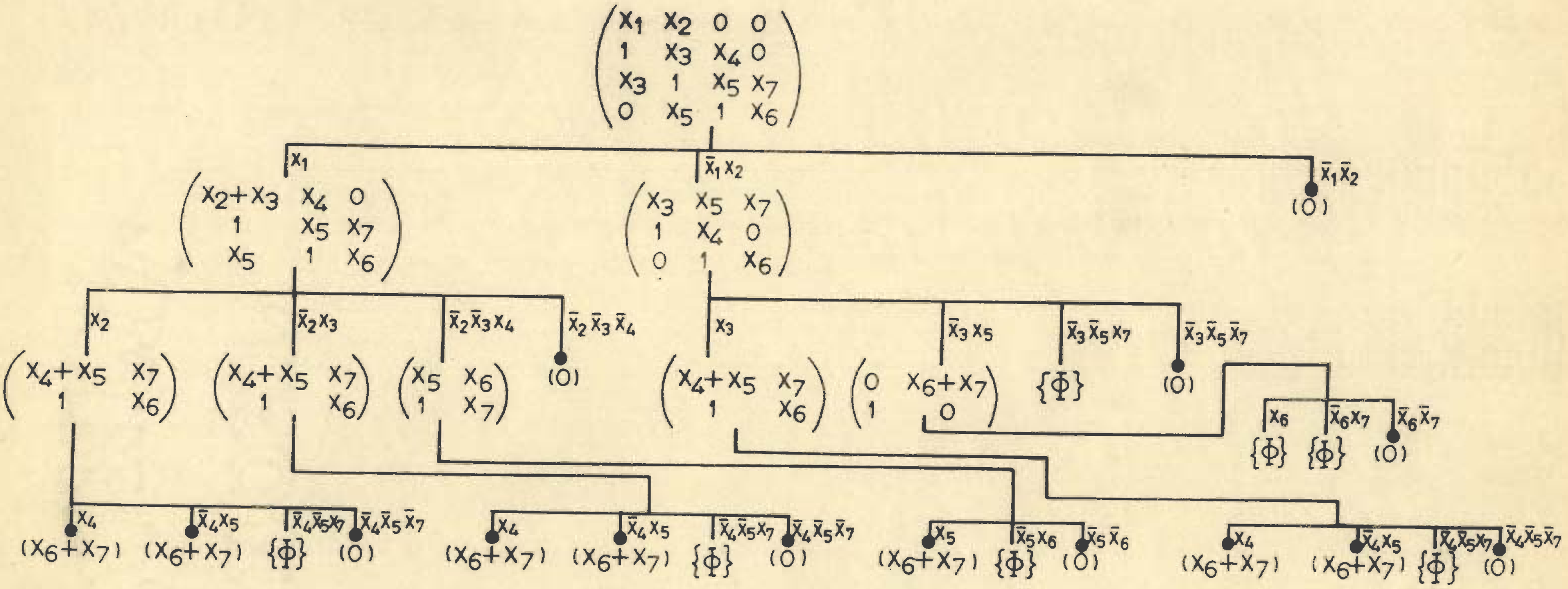


Figure 5.8. Details Steps of the Example 5.8.

Step 3. Set level $i \leftarrow 1$ and step 2 is being repeated until a pendant vertex of type mentioned in the algorithm is obtained. Figure 5.8 illustrates the application of step 2 and 3 of the algorithm for the network of figure 5.6.

Step 4. Traversing the tree from root to circle marked pendant vertex in figure 5.8 and replacing Boolean variables with corresponding probabilities, we get expression for terminal unreliability (UR) as

$$\begin{aligned} B_G(K) = {} & q_1 q_2 + p_1 q_2 q_3 q_4 + q_1 p_2 q_3 q_5 q_7 + p_1 p_2 q_4 q_5 q_7 \\ & + q_1 p_2 q_3 p_5 q_6 q_7 + q_1 p_2 p_3 q_4 q_5 q_7 + p_1 q_2 p_3 p_4 q_6 q_7 \\ & + p_1 q_2 q_3 p_4 q_5 q_6 + p_1 q_2 p_3 q_4 q_5 q_7 + p_1 p_2 p_4 q_6 q_7 \\ & + p_1 p_2 q_4 p_5 q_6 q_7 + q_1 p_2 p_3 q_4 p_5 q_6 q_7 + p_1 q_2 q_3 p_4 p_5 q_6 q_7 \\ & + p_1 q_2 p_3 q_4 p_5 q_6 q_7 + q_1 p_2 p_3 p_4 q_6 q_7 \end{aligned}$$

The minimal cutset in the network is obtained as usual by writing Boolean expression (F) consisting of only complemented variables from each of the term of $B_G(K)$ and simplifying it using Boolean relation $Y + YZ \equiv Y$. Minimal cutset expression thus obtained for figure 5.6 collecting only complemented variables is as under

$$F = \bar{X}_1\bar{X}_2 + \bar{X}_2\bar{X}_3\bar{X}_4 + \bar{X}_1\bar{X}_3\bar{X}_5\bar{X}_7 + \bar{X}_4\bar{X}_5\bar{X}_7 + \bar{X}_6\bar{X}_7 + \bar{X}_2\bar{X}_3\bar{X}_5\bar{X}_6$$

Hence minimal cutsets are $x_1\, x_2$, $x_2\, x_3\, x_4$, $x_1\, x_3\, x_5\, x_7$, $x_6\, x_7$, $x_4\, x_5\, x_7$, $x_2\, x_3\, x_5\, x_6$.

Example 5.9 – Consider a network of figure 5.6 for which NNGOS is computed between source node n_1 and sink node n_5

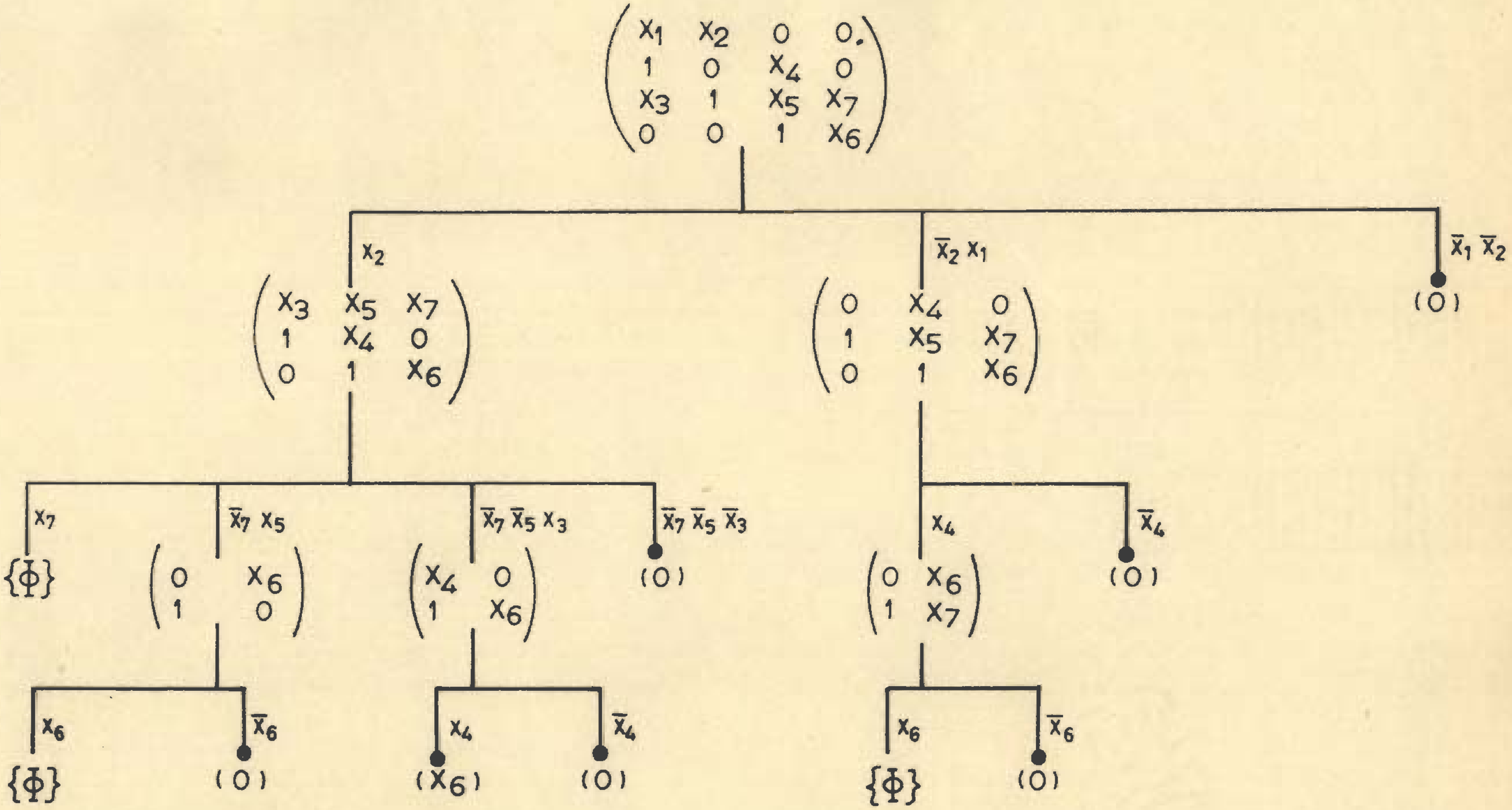


Figure 5.9(a). Details Steps of Example 5.9 Under SOC.

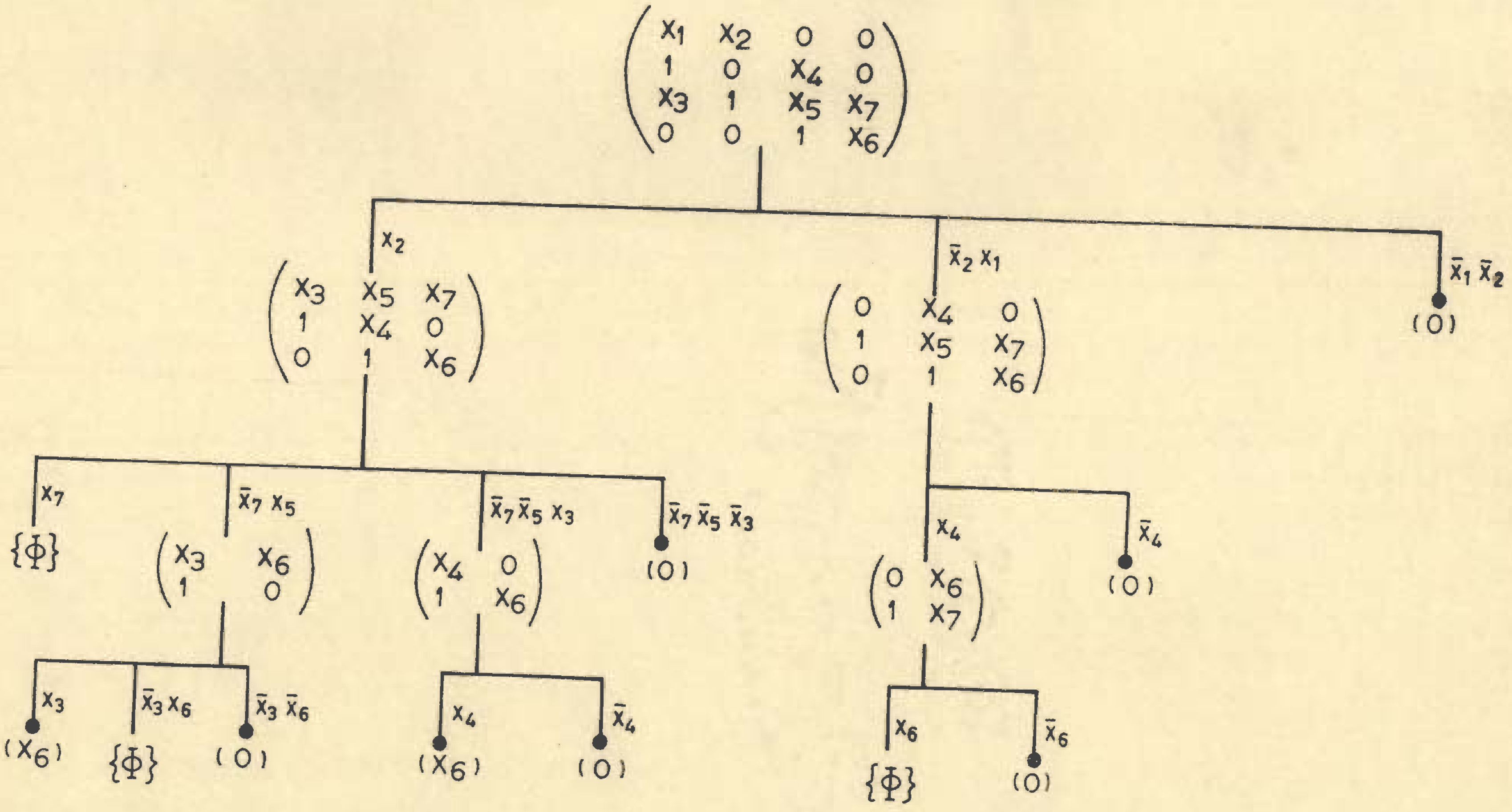


Figure 5.9 (b). Details Steps of Example 5.9 Under OOC.

under various call control stretegies. Table 5.4 shows the partial routing between n_1 and n_5 for the network of figure 5.6.

Table 5.4 Partial Routing Table between Nodes n_1 to n_5 for figure 5.6 (Asterisk has the same meaning defined with table 4.1).

FROM → TO ↓	n_1	n_2	n_3	n_4	n_5
n_5	$\overset{*}{n_3}, n_2$	n_4	n_5, n_4, n_2	n_5	–

Using the information given in table 5.4, NNGOS is computed. The detailed steps of algorithm is shown in figures 5.9a and 5.9b for SOC and OOC with spill forward call control strategies respectively. Note that expansion is being performed in accordance with the information given in table 5.4. For OOC with spill forward the branch x_2 incident on node n_3 (assuming the role of originating office) is treated similarly for 1-SUB operation as that considered in SOC. However 1-SUB operation for remaining entries is performed in accordance with OOC or reliability analysis.

5.2.2.3 Discussion –

a. As considered earlier, conservative policy [36] is utilized to obtain EMD events. For p-type (n-type), the logical expansion contains exactly one 1(0) and (i −1) 0's (1's) for all $i \leq n$. In the event, $i = n+1$, the expansion results into 000...0(111...1) combination. Note that only (n+1) terms are generated in this policy and hence the name conservative. The exhaustive policy, on the other hand, obtains 2^n terms starting from all 0's to all 1's.

Definition 5.7 – Mixed policy is one that makes use of
i) exhaustive policy for all $n \leq 2$, and
ii) conservative policy; otherwise.

The algorithm, presented above, utilizes the conservative p –type policy for its step 2(c). As such any one of the policies considered here may be used for single loss route strategies. An analysis of three networks shown in figures 4.5, 5.3 and 5.6 is made taking this fact into account. The result in terms of number of levels and factors being computed in tree along with total number of terms in final unreliability expression are given in table 5.5 and 5.6 respectively. It is obvious that out of various alternatives, conservative p –type policy is by far the best. It provides advantage at following stages:

i) (k −2) levels in tree need only be generated, and
ii) nearly 50 percent of the factors computed are present with final expression of unreliability of the network.

b. Blocking functions of a network have been expressed assuming switching machines (nodes) to be prefect or having

Table 5.5 Comparison of various expansion policies for (UR)

Description	Figure	Exhaustive Policy	Conservative policy		Mixed policy	
			p– type	n–type	p– type	n– type
Number of levels in tree	4.5	2	2	4	2	2
	5.3	4	4	5	4	5
	5.6	3	3	5	3	4
Number of factors computed in tree	4.5	12	10	15	12	12
	5.3	60	50	53	50	62
	5.6	36	29	45	32	38
Total terms with final expression of UR	4.5	6	6	6	6	6
	5.3	25	22	23	22	24
	5.6	16	15	16	15	16

Table 5.6 Comparison of various expansion policies for OOC

Description	Figure	Exhaustive policy	Conservative policy		Mixed policy	
			p–type	n– type	p– type	n –type
Number of levels in tree	4.5	2	2	3	2	2
	5.3	3	4	4	3	3
	5.6	3	3	3	3	3
Number of factors computed in tree	4.5	8	8	9	8	8
	5.3	20	20	22	20	20
	5.6	14	13	16	14	14
Total terms with final expression of NNGOS	4.5	4	4	4	4	4
	5.3	9	9	9	9	9
	5.6	6	6	6	6	6

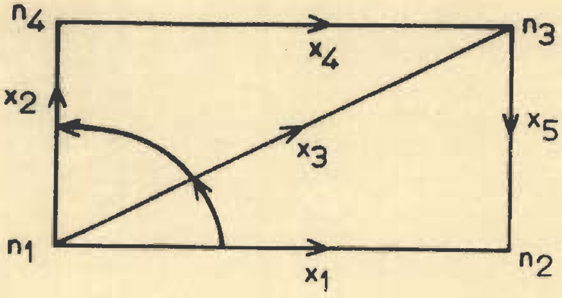


Figure 5.10. A Small Communication Network.

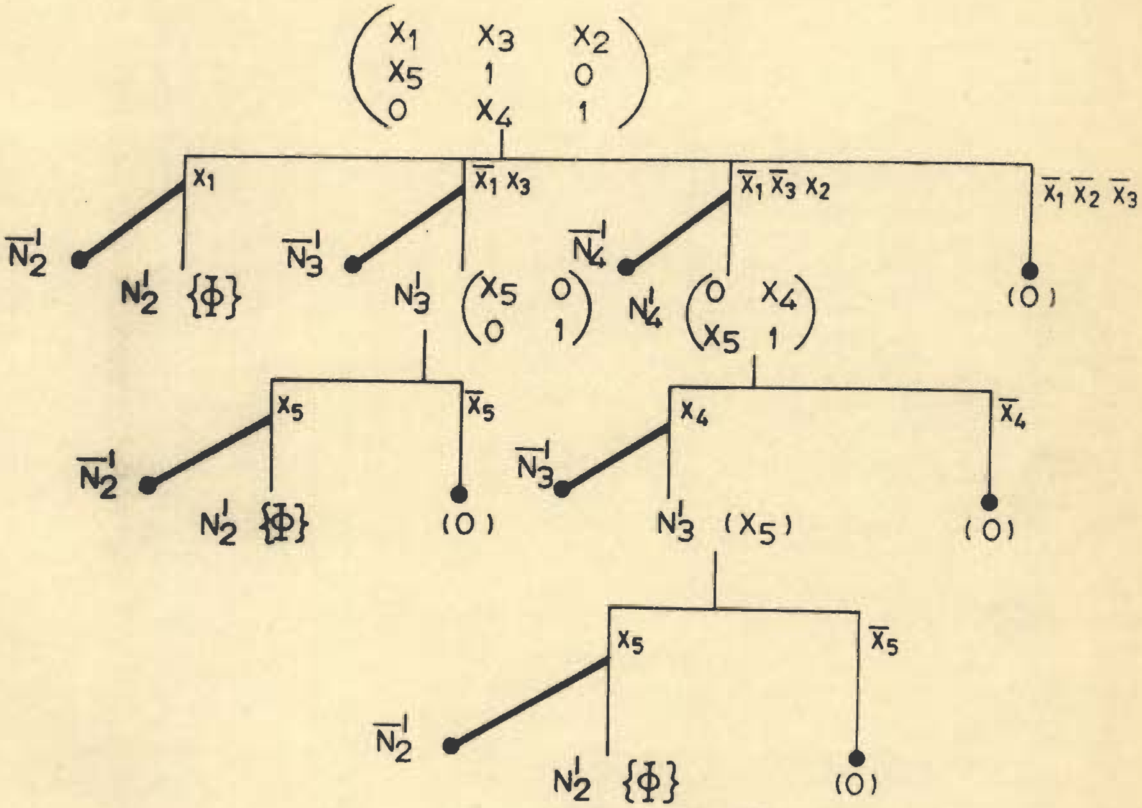


Figure 5.11. Detail Steps of Example 5.10.

infinite capacity. Let us consider two different blocking probabilities namely the incoming, $\overline{N}'_i$, and outgoing, $\overline{N}^o_i$, blocking probabilities for the switch n_i. The following assumptions are made:

i) If an outgoing call is blocked within the switch as it attempts to seize a given outgoing link, it overflows to the next alternative link, if one exists; otherwise it is cleared.

ii) If an incoming call is blocked, it is cleared.

For figure 5.1, link x_{j_1} connects node i and j_1. As discussed in subsection 5.2.1, P1 obtains the totality of condition under which the effect of link x_{j_1} being busy or not busy is illustrated. In the event N^o_i and N'_{j_1} are also to be considered, the result of P1 need following modification –

$$\overline{X}_{j_1} \rightarrow \overline{N}^o_i + N^o_i\, \overline{X}_{j_1} \qquad (5.15a)$$

$$X_{j_1} \rightarrow N^{\bullet}_i\, X_{j_1} \qquad (5.15b)$$

$$S_{j_1}(.) \rightarrow \overline{N}'_{j_1} + N'_{j_1}\, S_{j_1}(.) \qquad (5.15c)$$

Here, equations (5.15a) and (5.15b) identify the effect of assumption i) made above while (5.15c) that for ii).

Example 5.10– Consider the network shown in figure 5.10. For it, NNGOS between node n_1 to node n_4 assuming each node having incoming and outgoing blocking probability is given in equation (5.16). Steps are illustrated in figure 5.11.

Note that heavy lineas shown in it correspond to equation (5.15c). Each good(bad) branch variable is further modified by using equation (5.15a)(5.15b).

$$
\begin{aligned}
B_G(K) = {} & q^o_{n_1} + p^o_{n_1} q_1 q_2 q_3 + p^o_{n_1} q'_{n_2} p_1 + p^o_{n_1} q'_{n_3} q_1 p_3 \\
& + p^o_{n_1} q'_{n_4} q_1 p_2 q_3 + p^o_{n_1} p'_{n_3} q_1 p_3 [q^o_{n_3} + p^o_{n_3} q_5 + p^o_{n_3} q'_{n_2} p_5] \\
& + p^o_{n_1} p'_{n_4} q_1 p_2 q_3 [q^o_{n_4} + p^o_{n_4} q_4 + p^o_{n_4} q'_{n_3} p_4 + p^o_{n_4} p'_{n_3} p_4 (q^o_{n_3} \\
& + p^o_{n_3} q_5) + p^o_{n_4} p'_{n_3} p^o_{n_3} q'_{n_2} p_4 p_5] \qquad (5.16)
\end{aligned}
$$

c. A computer program in Pascal has been written for the proposed method. Its computer run time and memory requirement for several problems considered in [119] show the advantage of this method over [41,45].

5.3 CONCLUSION -

In this chapter the problem of computing NNGOS between a source/ destination node pair is considered under various call control strategies (viz SOC, OOC, OOC with spill forward). First, path loss sequences enumerated in chapter 4 are considered as an input and utilizing E – operator technique of reliability evaluation NNGOS is obtained.

Section 5.2, describes another method which does not require apeiori knowledge of path loss sequences and proposed method is recursive. Algebraic method, described in section 5.2.2., uses connection matrix as an input and does not require the knowledge of graph once the connection matrix has been obtained. The advantage with proposed method is that it requires lesser number of steps as compared to existing methods for computing NNGOS. Node failure probabilities are also considered. Examples are taken to illustrate various ideas.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE OF THE WORK

Section 6.1 summarizes the dissertation and discusses several conclusions drawn in it. Lastly, section 6.2 presents a list of problems that may be studied in future as an extension to the present work.

6.1 SUMMARY AND CONCLUSION

In this dissertation, some design problems of a CCN have been studied. First, a CCN is defined for its various relevant aspects. CCN classification in accordance with topology and message transport methodology has also been considered. Several design problems are described in chapter 2. These problems included topological optimization, capacity assignment to links, routing, network reliability, node to node call blocking and flow control. But for topological optimization and flow control, all other issues are considered in this dissertation.

The problem of link capacity assignment has been implicitely attempted in chapter 3. First, the integration of link capacity with its reliability is adopted as a new performance measure of a CCN. A method, utilizing the concept of terminal reliability evaluation has been given to solve this problem. The method is advantageous as it applies to large networks too. Second incorporating reliability as another constraint besides the cost, the performance measure outlined above has been made to select a particular capacity assignment rule. The study of reliability evaluation techniques revealed that most approaches needed an apriori knowledge of either all simple paths or cuts. Enumeration of paths, thus, became a necessary step in computing symbolic reliability expression. An algorithm, based on simple concepts of algebra, has been proposed to obtain paths between any source - destination node pair. The method is straightforward and computationally efficient. A computer program written in Pascal and run over DEC 20 system is given in appendix 1.

The problem of terminal pair reliability may, alternatively, be translated into NNGOS provided the path or cutset information is replaced by path loss (route) sequence. Note, the route sequence need a routing strategy and call control rule for its enumeration. Once such sequence is known there is no need to further refer to routing table and call control rules for computing NNGOS. A method, proposed for the path loss sequence, has been applied successfully to the CCN under various call control rules namely SOC, OOC and arbitrary routing plans.

Methods are also described for obtaining NNGOS for a circuit switched network. First, a Boolean method using E - operator has been proposed; the path loss sequence being considered as input to the method. A second method, not requiring apriori knowledge of path loss sequences has also been given. This method is recursive. Recursive approach generally leads to concise description of an algorithm. The proposed method, being graphical, is not suitable for computerization. Another recursive technique which is algebraic has been given to overcome this problem. The method, described in section 5.2.2, used connection and route matrix along with call control rule as the input. The computer program developed in [119] show the advantages of our method over [41,45] in terms of run time and memory requirement.

As nodes in the reliability graph of a CCN represent sophisticated and complex digital computers, a finite nonzero node incoming and outgoing probabilities have been assumed. Extension of the suggested method for NNGOS under above node blocking probabilities has also been considered.

6.2 FUTURE EXTENSION OF THE WORK

Based on the contribution of the dissertation the following suggestions are made for future extention.

1. Most reliability problems are, in worst case, NP -hard [120]. Network reliability problems are difficult than any standard combinatorial optimization problems. That is for a combinatorial problem given a solution, its correctness can be determined in polynomial time. However, for a reliability problem, given a solution, it can not even be checked without computing the reliability

of the network from a beginning. These problems provide a rich area for further investigation.

2. An efficient graph partitioning technique for reliability evaluation of large, highly interconnected network need be found.
3. Research on reliability analysis should proceed in the field of graph theoretic directions and concentrate on improving the efficiency of data processing and computation for very large and complex systems [120].
4. At present there is no comparable theoretical basis for analysing the computational complexity of logic (or fault) trees; structures quite useful for large systems. Research is needed in this field too.
5. The developments in CCN has aroused interest in computational techniques for more global reliability measures. It necessiates to obtain method for enumerating path and/or computing α source to β terminal reliability.
6. The method of computing performance index needs further investigations where topological consideration may also be incorporated.

REFERENCES

[1] M.Schwartz, Computer communication network design and analysis, Prentice Hall, Englewood Cliffs, New Jersey, 1977.

[2] N.Abramson and F.F.Kuo, Computer communication networks, Prentice Hall, Englewood Cliffs, New Jersey, 1973.

[3] A.S.Tanenbaum, Computer networks, Prentice Hall, Englewood Cliffs, New Jersey, 1981.

[4] V.Ahuja, Design and analysis of computer communication networks, McGraw Hill Book Company, New York,1982.

[5] L.Kleinrock, Queueing systems: computer applications, John Wiley, New York, 1976.

[6] W.H.Greene and U.W.Pooch, A review of classification schemes for computer communication networks, Computer, vol 10 Nov 1977, pp 12 -20.

[7] S.Rai, Some aspects of reliability of computers and communication networks, Ph.D dissertation, Kurukshetra University, Kurukshetra, 1979.

[8] C.S.Raghavendra and J.A.Silvester, Double loop network architecture - a performance study, IEEE Transactions Communications, vol COM -33, Feb 1985, pp 185 -187.

[9] W.Stallings, Local networks: an introduction, Macmillan Publishing Company, New York, 1984.

[10] R.R.Boorstyn and H.Frank, Large scale network topological optimization, IEEE Transactions Communications, vol COM -25, Jan 1977, pp 29 -47.

[11] K.K.Aggarwal, Integration of reliability and capacity in performance measure of a telecommunication network, IEEE Transactions Reliability, vol R -34, June 1985, pp 184 -186.

[12] P.M.Lin, B.J.Leon, and T.C.Huang, Analysis of circuit switched networks employing originated office control with spill forward, IEEE Transactions Communications, vol COM -26, June 1978, pp 754 -765.

[13] K.K.Aggarwal, Y.C.Chopra, and J.S.Bajwa, Topological layout of lines for optimizing the s-t reliability in a computer communication system, Microelectronics and Reliability, vol 22, 1982, pp 341 -345.

[14] ________, Topological lay -out of links for optimizing the over all reliability in a computer communication system, Microelectronics and Reliability, vol 22, 1982, pp 347 -351.

[15] M.Schwartz and T.E.Stern, Routing techniques used in computer communication networks, IEEE Transactions Communications, vol COM -28, April 1980, pp 539 -551.

[16] M.Gerla, Controlling routes, traffic rates and buffer allocation in packet networks, IEEE Communications Magzine, vol 22, Nov 1984, pp 11 -23.

[17] R.Petrovic and S.Jovanovic, Two algorithms for determining the most reliable path, IEEE Transactions Reliability, vol R –28, June 1979, pp 115–120.

[18] R.Billinton and R.N.Allan, Reliability evaluation of engineering systems: concepts and techniques, Planum Press, New York, 1983.

[19] A.P.Camarda et al, An effecient simple algorithm for fault tree automatic synthesis from the reliability graph, IEEE Transactions, Reliability, vol R–27, aug 1978, pp 215 –227.

[20] R.G.Bennetts, Analysis of reliability block diagrams by Boolean technique, IEEE Transactions Reliability, vol R– 31, June 1982, pp 159 –165.

[21] S.Rai and K.K.Aggarwal, An effecient method for reliability evaluation of a general network, IEEE Transactions Reliability, vol R –27, Aug 1978, pp 205 –211.

[22] M.O.Ball, Computing network reliability, Operations Research, vol 27, July –Aug 1979, pp 823 –837.

[23] A.Satyanarayana and J.N.Hagstrom, A new algorithm for the reliability analysis of multiterminal networks, IEEE Transactions Reliability, vol R –30, Oct 1981, pp 325 –334.

[24] G.B.Jamson and O.S.Kai, A new technique in minimal path and cutsets evaluation, IEEE Transactions Reliability, vol R –34, June 1985, pp 136 –141.

[25] G.B.Jamson, Cut -set analysis of networks using basic minimal paths and network decomposition, IEEE Transactions Reliability, vol. R -34, Oct 1985, pp. 303 -307.

[26] A.Satyanarayana and J.N.Hagstrom, A unified formula for analysis of some network reliability problems, IEEE Transactions Reliability, vol R -31, April 1982, pp 23 -32.

[27] K.K.Aggarwal and S.Rai, Reliability evaluation in computer communication networks, IEEE Transactions Reliability, vol R -30, April 1981, pp 32 -35.

[28] S.Rai, A cut - set approach to reliability evaluation in communication networks, IEEE Transactions Reliability, vol R -31, Dec 1982, pp 428 -431.

[29] K.D.Heidtman, Inverting paths and cuts of 2 -state systems, IEEE Transactions Reliability, vol R -32, Dec 1983, pp 469 -471.

[30] K.K.Aggarwal, J.S.Gupta and K.B.Mishra, Reliability evaluation of communication systems using flow graphs, Microelectronics and Reliability,vol 13, 1974, p 535.

[31] J.M.Kontoleon, An algorithm for evaluating overall reliability of special tree - networks, IEEE Transactions Reliability, vol R -29, Oct 1980, p 314.

[32] Z.Khalil, On the reliability of biconnected networks, Microelectronics and Reliability, vol 23, 1983, pp 71- 78.

[33] G.D.M.Pearson, Computer program for approximating the reliability characteristics of a cyclic directed graph, IEEE Transactions Reliability, vol R –26, April 1977, pp 32 – 38.

[34] M.L.Shooman, Probabilistic reliability: an engineering approach, McGraw Hill, New York, 1968.

[35] L.Fratta and U.G.Montanari, A Boolean algebra method for computing the terminal reliability in communication networks, IEEE Transactions Circuit Theory, vol CT –20, May 1973, pp 203 –211.

[36] ________, A recursive method based on case analysis for computing network terminal reliability, IEEE Transactions, communications, vol COM –26, Aug 1978, pp 1166 –1177.

[37] J.De Mercado, N.Spyratos, and B.A.Brown, A method for calculation of network reliability, IEEE Transactions Reliability, vol R –25, June 1976, pp 71 –76.

[38] A.M.Rushdi, On reliability evaluation by network decomposition, IEEE Transactions Reliability, vol R –33, Dec 1984, pp 379 –384.

[39] K.K.Aggarwal, Y.C.Chopra, and J.S.Bajwa, Reliability evaluation by network decomposition, IEEE Transactions Reliability, vol R –31, Oct 1982, pp 355 –358.

[40] M.O.Locks, Recursive disjoint products inclusion – exclusion and min – cut approximations, IEEE Transactions Reliability, vol. R –29, Dec 1980, pp 368 –371.

[41] J.A.Abraham, An improved method for network reliability, IEEE Transactions Reliability, vol R -28, April 1979, pp 58 -61.

[42] I.M.Soi and K.K.Aggarwal, Over all reliability evaluation for large computer communication networks: an MHC approach, Microelectronics and Reliability, vol 25, 1985, pp 215 -222.

[43] ________, Reliability indices for topological design of computer communication networks, IEEE Transactions Reliability, vol R -30, Dec 1981, pp 438 -443.

[44] S.Srivastava and I.M.Soi, Empirical prediction of overall reliability in computer communication network, Microelectronics and Reliability, vol 23,1983,pp 137-139.

[45] S.Harriri and C.S.Raghavendra, A symbolic reliability algorithm based on path set and cut set method, Report No. 1184 -5, University of Southern California, Los Angles, 1984, pp 1 -31.

[46] H.Ariyoshi, Cutset graph and systematic generation of separating sets, IEEE Transactions Circuit Theory, vol CT -19, May 1972, pp 233 -240.

[47] J.E.Biegel, Determination of tie -sets and cut -sets for a system without feedback, IEEE Transactions Reliability, vol R -26, April 1977, pp 39 -42.

[48] W.Mayeda, Graph theory, John Wiley and Sons, New York, 1972.

[49] S.Arun Kumar, and S.H.Lee, Enumeration of all minimal cut –sets for a node pair in a graph, IEEE Transactions Reliability, vol R –28, April 1979, pp 51 –55.

[50] U.Abel and R.Bicker, Determination of all minimal cut –sets between a vertex pair in a undirected graph, IEEE Transactions Reliability, vol R –31, June 1982, pp 167 –171.

[51] E.V.Krishnamurty and G.Komissar, Computer aided reliability analysis of complicated networks, IEEE Transactions Reliability, vol R –21, May 1972, pp 86 –89.

[52] F.Rudin, Enumerating all simple paths in a graph, IEEE Transactions Circuits and Systems, vol CAS –25, Aug 1978, pp 641 –642.

[53] S.Rai and K.K.Aggarwal, Path enumeration using flow graphs, Microelectronics and Reliability, vol 19,1979, pp 391 –393.

[54] G.S.Hura, On determination of all tie –sets and minimal cut sets between any two nodes of a graph through Petri nets, Microelectronics and Reliability, vol 23, 1983, pp 471 –475.

[55] G.S.Hura, Enumeration of all simple paths in a directed graph using Petri nets, Microelectronics and Reliability, vol 23, 1983, pp 157 –159.

[56] A.Rosenthal, Approaches to comparing cutset enumeration algorithms, IEEE Transactions Reliability, vol R –28, April 1979, pp 62 –66.

[57] R.B.Mishra and K.B.Mishra, Enumeration of all simple paths in a communication network, Microelectronics and Reliability, vol 20, 1980, pp 419 –426.

[58] J.L.Roco, Algebraical algorithm for computing complex system reliability, Microelectronics and Reliability, vol 24, 1984, pp 901 –903.

[59] H.C.Chen et al, A new algorithm for network system reliability, Microelectronics and Reliability, vol 25, 1985, pp 35 –40.

[60] J.A.Buzacott, The ordering of terms in cut –based recursive disjoint – products, IEEE Transactions Reliability, vol R –32, Dec 1983, pp 472 –474.

[61] S.Davamanoharan, A note on determination of tiesets and cutsets for system without feedback, IEEE Transactions Reliability, vol R –28, April 1979, pp 67 –69.

[62] U.Abel and R.Bicker, A counter example for algorithm of Arun Kumar and Lee, IEEE Trans. Reliability, vol R –30, Dec 1981, p 415.

[63] J.Vilar, Minimal cut upto third order in planar graph analysis, IEEE Transaction Reliability, vol R –33, Aug 1984, pp 250 –256.

[64] P.M.Lin, B.J.Leon, and T.C.Huang, A new algorithm for symbolic system reliability analysis, IEEE Transactions Reliability, vol R –25, April 1976, pp 2 –15.

[65] M.O.Locks, Inverting and minimalising path sets and cut sets, IEEE Transactions Reliability, vol R –27, June 1978, pp 107 –109.

[66] Suresh Rai, K.K.Aggarwal, On complementation of path sets and cut sets, IEEE Transactions Reliability, vol R –29, June 1980, pp 139 –140.

[67] M.O.Locks, Inverting and minimising Boolean functions minimal paths and cuts non –coherent system analysis, IEEE Transactions Reliability, vol R –28, Dec 1979, pp 373 – 375.

[68] T.B.Boffey and R.J.M.Waters, Calculation of system reliability by algebraic manipulation of probability expression, IEEE Transactions Reliability, vol R –28, Dec 1979, pp 358 – 363.

[69] D.R.Shier and D.E.Whited, Algorithms for generating minimal cut –sets by inversion, IEEE Transactions Reliability, vol R –34, Oct 1985, pp 314 –319.

[70] E.Hansler, G.K.McAuliffe and R.S.Wilkov, Exact calculation of computer network reliability, Networks 4,1974, pp 95 –112.

[71] J.Torrey, A pruned tree approach to reliability computation, IEEE Transactions Reliability, vol R –32 June 1983, pp 170 –174.

[72] H.Nakazawa, A decomposition method for computing system reliability by Boolean expression, IEEE Transactions Reliability, vol R –26, Oct 1977, pp 250 –252.

[73] W.Feller, An introduction of probability theory and its application, John Wiley and Sons, New York,1968.

[74] A.Agrawal and R.E.Berlow, A survey of network reliability and domination theory, Operations Research, vol 32, May -June, 1984, pp 478 -492.

[75] A.Satyanarayana and A.Prabhakar, New topological formula and rapid algorithm for reliability analysis of complex networks, IEEE Transactions Reliability, vol R -27, June 1978, pp 82 -100.

[76] A.M.Rushdi, Symbolic reliability analysis with the aid of variable entered karanaugh -maps, IEEE Transactions Reliability, vol R -32, June 1983, pp 134 -139.

[77] H.T.Nagle Jr. et al, An introduction to computer logic, Prentice Hall, New Jersey, 1975.

[78] T.Case, A reduction techniques for obtaining a simplified reliability expression, IEEE Transactions Reliabilit vol R -26, Oct 1977, pp 248 - 249.

[79] K.Gopal and S.Rai, Discussion on : a reduction technique for obtaining simplified reliability expression, IEEE Transactions Reliability, vol R -28, April 1979, pp 66 -6

[80] P.P.Gupta and S.C.Aggarwal, A Boolean algebra method for reliability calculations, Microelectronics and Reliability, vol 23, 1983, pp 863 -865.

[81] K.K.Aggarwal, J.S.Gupta and K.B.Mishra, A new method for system reliability evaluation, Microelectronics and Reliability, vol 12, 1973, pp 435 -440.

[82] K.K.Aggarwal, K.B.Mishra and J.S.Gupta, Reliability evaluation: a comparative study of different techniques, Microelectronics and Reliability, vol 14, Feb 1975, pp 49 –56.

[83] ______, A fast algorithm for reliability evaluation, IEEE Transactions Reliability, vol R –24, April 1975, pp 83 –85.

[84] O.D'antona and M.Dipietro, Boolean algorithms in terminal reliability evaluation, Dipartimento Di Informatica E. Sistemistica, Ridis –1 –83, 1983, pp 1 –37.

[85] A.Girard and Y.Ouimet, End to end blocking for circuit switched networks –polynomial algorithms for some special cases, IEEE Transactions Communications, vol Com –31, Dec 1983, pp 1269 –1273.

[86] M.O.Locks, Recursive disjoint products: a review of three algorithms, IEEE Transactions Reliability, vol R –31, April 1982, pp 33 –35.

[87] R.K.Tiwari and M.Verma, An algebraic technique for reliability evaluation, IEEE Transactions Reliability, vol R –29, Oct 1980, pp 311 – 313.

[88] W.G.Schenneweiss, Disjoint Boolean products via Shannon's expansion, IEEE Transactions Reliability, vol R –33, Oct 1984, pp 329 –332.

[89] R.S.Wilkov, Analysis and design of reliable computer networks, IEEE Transactions Communications, vol COM –20, June 1972, pp 660 –678.

[90] K.K.Lee, A compilation technique for exact system reliability, IEEE Transactions Reliability, vol R –30, Aug 1981, pp 284 –288.

[91] A.Laviron, A Carmino and B.Koen, Comments on a comparasion technique for exact system reliability, IEEE Transactions Reliability, vol R –31, April 1982, p 51.

[92] R.B.Hureley , Probability maps, IEEE Transactions Reliability, vol R –12, Sept 1963, pp 39 –44.

[93] E.Phibbs and S.H.Kuwamamota, An efficient map method for processing multistate logic trees, IEEE Transactions Reliability, vol R –21, May 1972, pp 93 –98.

[94] S.K.Benerjee and S.D.Bhinde, A technique to simplify reliability expression, IEEE Transactions Reliability, vol R –30,Aug 1981, pp 298 –299.

[95] D.B.Brown, A computerized algorithm for determining the reliability of redundant configurations, IEEE Transactions Reliability, vol R –20, Aug 1971, pp 121 –124.

[96] H.Nakazawa, Bayesion decomposition method for computing the reliability of an oriented network, IEEE Transactions Reliability, vol R –25, June 1976, pp 77 –80.

[97] ________,A decomposition method for computing system reliability by a matrix expression, IEEE Transactions Reliability, vol R –27, Dec 1978, pp 342 –344.

[98] _____, Decomposition method for computing the reliability of complex networks, IEEE Transactions Reliability, vol R-30, June 1981, pp 289-292.

[99] A.M.Rushdi and A.S.Goda, Symbolic reliability analysis via Shannon's expansion, Microelectronics and Reliability, vol 25, 1985, pp 1041-1053.

[100] S.H.Ahmad, A simple technique for computing network reliability, IEEE Transactions Reliability, vol R-31, April 1982, pp 41-44.

[101] R.Johnson, The Wheatstone bridge reduction in network reliability computation, IEEE Transactions Reliability, vol R-32, Oct 1983, pp 374-378.

[102] H.Nakazawa, Equivalence of a non-oriented line and a pair of oriented lines in a network, IEEE Transactions Reliability, vol R-28, Dec 1979, pp 364-367.

[103] _____, Reliability equivalent separation of a node, IEEE Transactions Reliability, vol R-34, June 1985, pp 131-135.

[104] J.A.Buzacott and J.S.K.Chang, Cut-set intersections and node partitions, IEEE Transactions Reliability, vol R-33, Dec 1984, pp 385-389.

[105] J.N.Hagstorm, Using the decomposition tree of a network in reliability evaluation, IEEE Transactions Reliability, vol R-23, April 1983, pp 71-77.

[106] _____, Using decomposition tree for directed network reliability computation, IEEE Transactions Reliability, vol R-33, Dec 1984, pp 390-395.

[107] K.K.Aggarwal, Y.C.Chopra, and J.S.Bajwa, Capacity considerations in reliability analysis of communication systems, IEEE Transactions Reliability, vol R -31, June 1982, pp 177 -181.

[108] ______, Modification of cutsets for reliability evaluation of communication systems, Microelectronics and Reliability, vol 22, 1982, pp 337 -340.

[109] S.H.Lee, Reliability evaluation of a flow network, IEEE Transactions Reliability, vol. R -29, April 1980, pp 24 -26.

[110] K.B.Mishra and P.Prasad, Comments on : reliability evaluation of flow network, IEEE Transactions Reliability, vol R -31, June 1982, pp 174 -176.

[111] B.L.Deuermeyer, A new approach for network reliability analysis, IEEE Transactions Reliability, vol R -31, Oct 1982, pp 350 -354.

[112] D.Trtensky and P.Bowron, An alternative index for the reliability of telecommunication networks, IEEE Transactions Reliability, vol R -33, Oct 1984, pp 343 -345.

[113] K.K.Aggarwal, J.S.Gupta and K.B.Mishra, A simple method for reliability evaluation of a communication system, IEEE Transactions Communications, vol COM- 23, May 1975, pp 563 -566.

[114] M.D.Gaudreau, Recursive formula for the calculation of point to point congestion, IEEE Transactions Communication, vol COM -28, March 1980, pp 313 -316.

[115] K.K.Aggarwal and S.Rai, Symbolic reliability evaluation using logical signal relations,IEEE Transactions Reliability, vol. R –27, Aug 1978, pp 202 –205.

[116] C.L.Hwang, F.A.Tillman and M.H.Lee, System reliability evaluation techniques for complex/large systems – a review, IEEE Transactions Reliability, vol R –30, Dec 1981, pp 416 –423.

[117] W.S.Chan, Recursive algorithms for computing end to end blocking in a network with arbitrary routing plan, IEEE Transactions Communications, vol COM –28, Feb 1980, pp 153 –164.

[118] M.N.S.Swamy and K.Tulsiraman, Graphs,networks and algorithms, John Wiley and Sons, New York, 1981.

[119] A.K.Chowdhary, Comparative study of reliability evaluation methods in computer network, ME dissertation, University of Roorkee, Roorkee, 1986.

[120] J.S.Provan and M.O.Ball, Computing network reliability in time polynomial in the number of cuts, Operations Research, vol 32, May –June 1984, pp 516 –526.

VITA

Name	ARUN KUMAR
Birth date and place	July 3, 1952; Roorkee
Education	B.E.(Electronics and Communication) 1974, University of Roorkee, Roorkee. M.E.(Communication System) 1977, University of Roorkee; Roorkee.
Position held	Lecturer in Department of Electronics and Communication Engineering; University of Roorkee, (from April 1982). Assistant Engineer in Doordarshan; All India Radio,(during 1976 to 1982).

Research Publications

[1] Recursive Technique For Computing System Reliability, *IEEE Trans. Reliability* (accepted for publication).

[2] Computing Terminal Reliability of Computer Network, *Reliability Engineering*, 15, 1986.

[3] Computing Performance Index of a Computer Network, *Reliability Engineering* (accepted for publication).

[4] On Path Enumeration, *Int. J.Electronics*, vol 60,1986.

[5] An Algebraic Technique for Reliability Evaluation of Computer Network, *Presented in National Conference on Reliability Engineering at IIT Kharagpur*, Dec 1985.

[6] An Algebraic Method to Compute NNGOS under various call Control Rules (under preparation).

APPENDIX-1

```
PROGRAM PATHLIST(INPUT,OUTPUT);
TYPE
        MATRIX=ARRAY[1..10,1..10]OF REAL;
        DI=ARRAY[1..10]OF INTEGER;
VAR
        B,A:MATRIX;
        J,D:DI;
        I,L,M,N,P,K,S1,S2,X,Y,K1,Z:INTEGER;
(*  P R O C E D U R E  T O  F I N D  O U T  *)
(*  A N D  P R I N T  V A R I O U S  P A T H S  *)
(*  B E T W E E N  S O U R C E  A N D  S I N K   *)
PROCEDURE GEN(VAR B:MATRIX;J,D:DI;I,K,Z,K1:INTEGER);
        LABEL 10,20;
TYPE
        QI=ARRAY[1..10] OF INTEGER;
VAR
        G,H:QI;
        PASS:BOOLEAN;
        X2,X5,Y1,Y2,Y3:REAL;
        X1,X4,A,C,M,N,R,L,B1,B2,X3,I1,X7,X6,A1,A2:INTEGER;
BEGIN
        IF I=K THEN GOTO 20;
        FOR L:=1 TO D[I] DO
BEGIN
        J[I]:=L;
        R:=1;
        FOR M:=1 TO I-1 DO
BEGIN
        N:=J[M];
        Y1:=B[I,L];
        B1:=TRUNC(Y1);
        Y2:=B[M,N];
        B2:=TRUNC(Y2);
        PASS:=((B1=B2)OR(B1=0)OR(B2=0));
        IF PASS THEN
BEGIN
        R:=0;
        GOTO 10;
END
END;
        IF R<> 0 THEN
BEGIN
        IF I=K-1 THEN
```

```
        A2:=0;
        FOR A:=1 TO K-1 DO
BEGIN
        Z:=Z+1;
        IF Z=10 THEN
BEGIN
        K1:=K1+1;
        Z:=0;
END;
        C:=J[A];
        Y3:=B[A,C];
        X1:=TRUNC(Y3);
        X2:=(B[A,C] - X1);
        X3:=1;
        FOR I1:=1 TO K1 DO
        X3:=X3*10;
        X2:=X2*X3+0.5;
        X4:=TRUNC(X2);
        IF X1<>X4 THEN
BEGIN
        A2:=A2+1;
        G[A2]:=X1;
        H[A2]:=X4;
END;
END;
BEGIN
X6:=S2;
        WHILE X6<>S1 DO
BEGIN
        FOR A1:=1 TO K-1 DO
BEGIN
        IF G[A1]=X6 THEN
        BEGIN
          WRITE(G[A1]:2,H[A1]:2,'*');
          X6:=H[A1]
        END
        END
        END;
END;
END
        ELSE GEN(B,J,D,I+1,K,Z,K1);
        END;
10:     WRITELN;
```

```
20:     WRITELN;
        END;
BEGIN
        READ(K);
        K1:=1;
        Z:=0;
        X:=0;
        FOR L:=1 TO K DO
        FOR M:=1 TO K DO
        READ(A[M,L]);
        READ(S1);
        READ(S2);
        FOR M:=1 TO K DO
        IF (M<>S2) THEN
BEGIN
        X:=1+X;
        Y:=0;
        FOR L:=1 TO K DO
BEGIN
        IF (L<>S1) AND (A[M,L]<>0)  THEN
BEGIN
        Y:=1+Y;
        B[X,Y]:=A[M,L];
END;
END;
END;
        FOR N:=1 TO K-1 DO
        READ (D[N]);
        FOR P:=1 TO K-1 DO
        READ(J[P]);
        GEN (B,J,D,1,K,Z,K1);
END.
```

```
4
1.1 1.2 1.3 0.0
0.0 2.2 2.3 2.4
0.0 3.2 3.3 3.4
0.0 0.0 0.0 4.4
4
1
3 3 2
1 1 1
```

```
1 2* 2 3* 3 4*

1 2* 2 4*

1 3* 3 4*

1 3* 3 2* 2 4*
```