1992

High-performance supra-high-speed packet-switched networking using neural arbiters.

Bhumip. Khasnabish

University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation


https://scholar.uwindsor.ca/etd/2305

This online database contains the full-text of PhD dissertations and Masters’ theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000 ext. 3208.
NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.
HIGH-PERFORMANCE SUPRA-HIGH-SPEED PACKET-SWITCHED NETWORKING USING NEURAL ARBITERS

by

Bhumip KHASNABISH

A Dissertation Submitted to the Faculty of Graduate Studies and Research Through the Department of Electrical Engineering in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy at the University of Windsor

Windsor, Ontario, Canada,

1992
The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-78917-4
ABSTRACT OF THE DISSERTATION

'HIGH-PERFORMANCE SUPRA-HIGH-SPEED PACKET-SWITCHED
NETWORKING USING NEURAL ARBITERS'

by

Bhumip Khasnabish

Doctor of Philosophy in Electrical Engineering, 1992

University of Windsor, Windsor, Ontario, Canada, N9B 3P4.

Advisors: Dr. Majid Ahmadi and Dr. Malayappan Shridhar.

The past few years have visualized tremendous research interests along with associated breakthroughs in Optical Fiber Communications (OFCs) and Artificial Neural Networks (ANNs). The first one has resulted in the elimination of the transmission-speed bottleneck of the pre-mid-1980 era, and hence cost-effective deployment of optical fibers in both short- and long-haul telecommunication networks. And the second has led to the development of massively parallel interconnection structure with simple, i.e., of the add-compare-select type, processing elements for solving large-scale, multi-criterion decision making problems in both on-line and off-line fashions at very high speed.

To utilize effectively the vast fiber bandwidth, researchers have proposed the Wavelength Division Multiplexing (WDM) technique which reduces the impact of speed mis-match between nodal processing and link transmission. Further enhancements in performance can be achieved when neurally arbitrated multi-connected regular topology -- constructed using WDM -- is used as the operational connection structure of any arbitrary physical topology.

This dissertation deals with the development of new and efficient techniques for evaluating architecture and packet routing strategy, and enhancing performance using ANNs, of the Manhattan Street Network (MSN) which is a special kind of two-connected regular mesh topology. MSN uses uni-directional links with adjacent channels carrying
packets in opposite directions. The end nodes of every row and column are directly connected using wrap-around links.

A new analytical technique for evaluating the architecture of arbitrarily large MSNs in a traffic-distribution-specific manner has been developed. The validity of this technique has been verified through computer simulation. A novel simple analytical technique for evaluating deflection routing in the MSNs has also been developed. The advantage of this technique lies in the fact that although it computes the Mean Packet Transfer Time (MPTT) by simply adding the Mean Inter-Node Distance (MIND), deflection and waiting penalties, it offers a reasonable degree of accuracy as ratified by simulation. Finally, it is shown that when Grossberg’s ANN along with a fuzzy logic based pre-processor is used for packet route arbitration, congestion-free operation of a network can be easily achieved.
To my mother, Meena Khasnabish, who struggled so that
I could pursue my ambition for higher studies,

and

to the memories of my sister, Mipon Khasnabish, and father, Jyoti Bikash
Khasnabish, whose unnatural deaths are rooted in the partition of India, 1947.
ACKNOWLEDGEMENTS

The author is grateful to his advisors Dr. M. Ahmadi, and Dr. M. Shridhar for valuable guidance throughout the progress of this dissertation. Thanks are also due to Dr. J. J. Soltis, and Dr. H.K. Kwan for allowing me to have frequent discussions with them. Dr. S. Erfani of the AT&T Bell Labs., Dr. M. Hlynka of the Department of Mathematics and Statistics, Dr. R. S. Lashkari of the Industrial Engineering Department, and Dr. K. Thulasiraman of Concordia University helped me with their interesting comments and opinions on all the relevant topics of this dissertation. Numerous suggestions from the External Examiner of this dissertation, Dr. J. F. Hayes of Concordia University, Montreal, Canada, have substantially improved both presentation and quality of this research work, I am indebted to him.

Some of my friends, who have been kind enough to allow me to shout at them on any uninteresting topics during the ups and downs of my research work, deserve special reference in these acknowledgements. They include: Ashmita, Billy, Devendra, Jalaja, Jim, Kim, Monica, Ranu, Rumana, and Susan.

Finally, I most sincerely acknowledge the contributions of my mother, brothers: Chayan, Tuhin and Bhula, and sisters: Kanan and Komando, in the remote tiny part of the world named Bangladesh. They not only kept me aware of what has been going on there during my absence, but also used to write to me regularly words of encouragement about my studies and the degree here.
# TABLE OF CONTENTS

ABSTRACT iv

DEDICATION vi

ACKNOWLEDGEMENTS vii

LIST OF TABLES xii

LIST OF ILLUSTRATIONS xiv

LIST OF SYMBOLS AND ABBREVIATIONS xxiii

## CHAPTER

### I. INTRODUCTION 01

1.1 Future Networking: Trends and Demands 01

1.2 Novel Network Architectures And Their Operating Principles 04

1.2.1 Lightwave Networks: An Overview 06

1.2.2 Architectural Issues 08

1.2.3 Operational Features 10

1.3 From Circuit Switching to Fast Packet Switching 11

1.3.1 Nodal Switching Techniques 13

1.4 Motivations and Impact of this Research 16

1.5 Organization of this Dissertation 17

1.6 Conclusions 19

### II. TAXONOMY OF THE CURRENTLY PROPOSED LARGE SUPRA-HIGH-SPEED PACKET SWITCHING NETWORKS 20

2.0 Introduction 20

2.1 Direct or Uni-Level Design 21

2.2 Indirect or Multi-Level or Hierarchical Design 27

2.3 Conclusions and Future Trends 35

viii
III. EVALUATION OF NETWORK ARCHITECTURES

3.0 Introduction
3.1 The Problem Statement and the Proposed Technique
3.2 The UTD Pattern in the MSNs
   3.2.1 Determination of the MIND and \(T(z)\)
3.3 The ZTD Pattern in the MSNs
   3.3.1 Determination of the MIND and \(T(z)\)
3.4 Conclusions

IV. EVALUATION PACKET ROUTING

4.0 Introduction
4.1 The Routing Strategy to be Evaluated
4.2 A Step-Wise Description of the Proposed Technique
4.3 A Model for Determining the \(N_{df/\text{max}}\) in the MSNs
4.4 Performance Evaluation
   4.4.1 Variation of \(U\) with \(a\)
   4.4.2 Variation of \(D_{\text{avg}}\)
      4.4.2.1 Variation of \(N_{df}\) and \(D_{m}\) with \(a\) and \(p\)
      4.4.2.2 Variation of \(p\) with \(a\), \(b\), and \(\lambda\)
      4.4.2.3 Variation of \(D_{\text{avg}}\) with \(a\), \(b\), and \(\lambda\)
4.5 Simulation Scenarios and Results
4.6 Conclusions

V. CONGESTION AVOIDANCE USING NEURAL ARBITERS

5.0 Introduction
5.1 Different States of Network Operation
5.2 Types of Network Congestion
   5.2.1 Desirable Characteristics of the Congestion Control Mechanisms
   5.2.2 Mechanisms for Avoiding/Controlling Congestion
5.3 How the Artificial Neural Networks (ANNs) Based Approaches Can be Employed in Packet-Switched Systems
5.4 Motivations for Utilizing Artificial Neural Networks (ANNs) for Congestion-Free Operation of Packet-Switched Systems
5.5 Utilization of Neural Arbiters for Congestion Avoidance
   5.5.1 The Proposed Procedure
   5.5.2 Performance Evaluation
   5.5.2a Description of the Simulation Model
   5.5.2b Results
5.6 Conclusions

VI. SUMMARY, CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

6.1 Contributions of this Dissertation
6.2 Suggestions for Future Research

APPENDIX-A: SC-TC TRADEOFF IN THE MSNS AND EMBEDDING MSNs ONTO RING NETWORKs

A.1 Problem Statement And The Proposed Procedure
A.2 Analysis of Zero And Non-Zero Buffer Cases
   A.2.1 The Zero (Waiting) Buffer Case
      A.2.1.1 Justification of the Approximations used in this Section
      A.2.1.2 Determination of the Expression for $E(S_0)$
   A.2.2 A Non-Zero (Waiting) Buffer Case
      A.2.2.1 Determination of the Expression for $E(S_1)$
A.3 Results
A.4 Remarks
A.5 Embedding MSNs Onto Ring Networks

APPENDIX-B: DETERMINATION OF THE EXACT WAITING TIME FOR AN M/D/1 QUEUEING SYSTEM WITH FINITE (≤ 4) WAITING ROOM

B.1 The Zero Buffer Case
B.2 The Two Buffer Case
B.3 The Four Buffer Case

APPENDIX-C: SWITCHING PATTERNS OF A 3X3 SWITCH
APPENDIX-D: LOCAL BUFFER STEALER

APPENDIX-E: MAXIMUM SELECTION, MINIMUM SELECTION AND SORTING USING ARTIFICIAL NEURAL NETWORKS (ANNs)

E.1 Maximum Selection Network 205
E.2 Minimum Selection Network 207
E.3a Sorting Network (Type-1) 209
E.3b Sorting Network (Type-2) 210

APPENDIX-F: LEARNING IN THE ADAPTIVE RESONANCE THEORY (ART) BASED ARTIFICIAL NEURAL NETWORKS (ANNS)

F.1 Description of the ART-1 Machines 213
F.2 Major Properties of the ART-1 Machines 214
F.3 Operational Features of the ART-1 Machines 215
F.4 Learning in the ART-1 Machines 218
F.5 A Step-Wise Procedure for Developing ART-1 Model for a Given Binary Pattern Recognition Problem 223
F.6 A Numerical Example 229

BIBLIOGRAPHY

VITA AUCTORIS

xi
List of Tables

Table 4.1: Showing the Variation of $(N_{df})_{\text{max}}$ and $D_m$ with the Tuning Parameter, $a$, and the Probability of Deflections, $p$, for Two Types of Manhattan Street Networks (MSNs) for the Uniform Traffic Distribution (UTD) Pattern. 

Table 4.2: Showing the Variation of $(N_{df})_{\text{max}}$ and $D_m$ with the Tuning Parameter, $a$, and the Probability of Deflections, $p$, for the Zonal Traffic Distribution (ZTD) Pattern in a Type-1 Manhattan Street Networks (MSN). 

Table 4.3: Variation of the Probability (Choice Exists), Between Two Outgoing Links, at any Node en route to Destination for Case-A and Case-B in a $4 \times 4$ i.e., 16-Node Manhattan Street Network (MSN). 

Table 4.4: Variation of the Probability that an Arrival Finds that the System is Full with variation of Arrival Rate, $\lambda$, and the Size of Waiting Buffer, $b$. 

Table 4.5: Variation of the Probability of Deflection, $p$, with the Tuning Parameter, $a$, Size of the Waiting Buffer, $b$, and the Packet Arrival Rate, $\lambda$, for Case-A in a $4 \times 4$, i.e., 16-Node Manhattan Street Network (MSN). 

Table 4.6: Variation of the Average Waiting Time in Number of Slots for an M/D/1/(b+1) Queueing System with the Size of the Waiting Buffer, $b$, and the Packet Arrival Rate, $\lambda$. 

Table 4.7: Variation of the Penalties in Number of Slots Due to Waiting and Deflections in a $4 \times 4$ i.e., 16-Node Manhattan Street Network (MSN) for Case-A with Packet Arrival Rate Per Slot Per Station, $\lambda$, and the Size of the Waiting Buffer, $b$ Per Outgoing Link for $a=1.0$. [Note: $\varphi_{\text{avg}}=2$ Hops for the MSNs and the Minimum Mean Inter-Node Distance for a 16-Node MSN is $2.93=3.0$ Hops].
Table 4.8: Variation of the Penalties in Number of Slots Due to Waiting and Deflections in a 4x4 i.e., 16-Node Manhattan Street Network (MSN) for Case-A with Packet Arrival Rate Per Slot Per Station, \( \lambda \), and the Size of the Waiting Buffer, \( b \) Per Outgoing Link for \( a=2.0 \). [Note: \( \phi_{avg} =2 \) Hops for the MSNs, and the Minimum Mean Inter-Node Distance for a 16-Node MSN is 2.93=3.0 Hops]................................................101

Table 4.9: Variation of the Penalties in Number of Slots Due to Waiting and Deflections in a 4x4 i.e., 16-Node Manhattan Street Network (MSN) for Case-A with Packet Arrival Rate Per Slot Per Station, \( \lambda \), and the Size of the Waiting Buffer, \( b \) Per Outgoing Link for \( a=3.0 \). [Note: \( \phi_{avg} =2 \) Hops for the MSNs, and the Minimum Mean Inter-Node Distance for a 16-Node MSN is 2.93=3.0 Hops]................................................102

Table 5.1: Number of Nodes, \( N_i \), Separated by \( i \) Hops in a 4x4 Manhattan Street Network (MSN). Note that the Diameter of a 4x4 MSN is 5 Hops [78]..149

Table 5.2: Variation of the Normalized -- with Respect to the Mean Delay at Zero Load -- Mean Packet Transfer Time (MPTT) with the Number of Active User Pairs. MPTT is Measured in Terms of Hop-Count, as Defined Earlier in Section 5.5.................................................................150

Table 5.3: Variation of Network Throughput with the Normalized Number of Active User Pairs.................................................................152
List of Illustrations

Figure 1.1: Different Types of Switching, (a) Circuit Switching, and (b) Packet Switching.................................................................12

Figure 1.2: A Single Stage Switch.................................................................14

Figure 1.3a: A Multi-Stage Switch (Functional Level) without Feedback........14

Figure 1.3b: A Multi-Stage Switch with Feedback........................................15

Figure 2.1: Standard Seven Layer Architecture for Open System Interconnection (OSI), as Recommended by the International Standardization Organization (ISO).................................................................22

Figure 2.2: A 4x4 Manhattan Street Network (MSN) (adapted from [103]).....26

Figure 2.3: Various Bus Topologies for Local and Wide Area Networks; (a) Simple Bus Topology, as used in Ethernet (IEEE 802.3 Standard) and Token Bus (IEEE 802.4 Standard) Networks, (b) Dual Bus Topology, as used in the QPSX Man (IEEE 802.6 Standard), (c) Folded Bus Topology, as Proposed for the Expressnet [91], (d) Headed Bus Topology, as can be used for Physical Topology of a Multi-Hop Network (MHN) [35].................................................................28

Figure 2.4: Various Ring Topologies for Local and Wide Area Networks; (a) Unidirectional Ring Network, as can be used for IBM’s Token Ring (IEEE 802.5 Standard) Networks, (b) Dual-channel Ring Topology for FDDI Networks, (c) Chordal Ring Network, as can be used for Physical Topology to Embed Manhattan Street Network (MSN) [81].................................................................29
Figure 2.5: Various Other Topologies for Local and Wide Area Networks; (a) A Regular Mesh Topology, (b) An Irregular Mesh Topology, (c) An Eight-leaf Tree Network, (d) An Eight-node Star Network

Figure 2.6: Examples of Hierarchical Topologies for Wide Area Networks; (a) An Interconnected Star-Of-Star (SOS) Network, (b) An Interconnected Tree Network

Figure 2.7: A Taxonomy of the Currently Proposed Large Supra-High-Speed Packet Switching Networks

Figure 3.1: A 4x4 Manhattan Street Network (MSN) (adapted from [103])

Figure 3.2a: A 2x2 Manhattan Street Network (MSN)

Figure 3.2b: A 2x4 Manhattan Street Network (MSN), Constructed by Cascading TWO 2x2 MSNs

Figure 4.1: A 4x4 Manhattan Street Network (MSN) (adapted from [103])

Figure 4.2a: An Illustration of the Case-A for d=4 Hops

Figure 4.2b: An Illustration of the Case-B for d=4 Hops

Figure 4.3: Moves with Deflections to Reach the Destination in a 4x4 i.e., 16-node Manhattan Street Network (MSN) for (a) 1-Hop, (b) 2-Hop, (c) 3-Hop, (d) 4-Hop, (e) 5-Hop Source-Destination Patterns, [Legend: P=Preference Exists, NP=No Preference Exists, DF=Deflection Suffered, -->=Destination Reached]

Figure 4.4: Variation of Network Utilization, U, with the Tuning Parameter, a, for the Uniform Traffic Distribution (UTD) Pattern

Figure 4.5a: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, p, in a Type-1 MSN (a 36-Node Network) for Case-A for the Uniform Traffic Distribution (UTD) Pattern
Figure 4.5b: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, p, in a Type-1 MSN (a 36-Node Network) for Case-B for the Uniform Traffic Distribution (UTD) Pattern....67

Figure 4.6a: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, p, in a Type-2 MSN (a 16-Node Network) for Case-A for the Uniform Traffic Distribution (UTD) Pattern....69

Figure 4.6b: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, p, in a Type-2 MSN (a 16-Node Network) for Case-B for the Uniform Traffic Distribution (UTD) Pattern....70

Figure 4.7: Variation of the Delay, $D_m$, in unit of hop-count with the Probability of Deflections, p, for the Uniform Traffic Distribution (UTD) Pattern in a 36-Node MSN.................................................................71

Figure 4.8: Variation of the Delay, $D_m$, in unit of hop-count with the Probability of Deflections, p, for the Uniform Traffic Distribution (UTD) Pattern in a 16-Node MSN.................................................................72

Figure 4.9a: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, p, in a Type-1 MSN (a 36-Node Network) for Case-A for the Zonal Traffic Distribution (ZTD) Pattern with $\zeta = 0.75$.................................................................75

Figure 4.9b: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, p, in a Type-1 MSN (a 36-Node Network) for Case-B for the Zonal Traffic Distribution (ZTD) Pattern with $\zeta = 0.75$.................................................................76

Figure 4.10a: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, p, in a Type-2 MSN (a 16-Node Network) for Case-A for the Zonal Traffic Distribution (ZTD) Pattern with $\zeta = 0.75$.................................................................77
Figure 4.10b: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, p, in a Type-2 MSN (a 16-Node Network) for Case-B for the Zonal Traffic Distribution (ZTD) Pattern with $\zeta=0.75$. ................................................................. 78

Figure 4.11: Variation of Delay, $D_m$, in unit of hop-count, with the Probability of Deflection, p, for Different Values of $\zeta$ of Zonal Traffic Distribution (ZTD) Pattern in a 36-Node MSN ........................................... 79

Figure 4.12: Effect of Buffer Size, b, on the Probability of Packet Loss (i.e., Deflection) as a Function of the Packet Arrival Rate, $\lambda$ .......... 82

Figure 4.13a: Effect of Buffer Size, b, on the Probability of Deflection, p, as a Function of the Packet Arrival Rate, $\lambda$, for $a=1.0$, in a 4x4 Manhattan Street Network (MSN) ....................................................... 83

Figure 4.13b: Effect of Buffer Size, b, on the Probability of Deflection, p, as a Function of the Packet Arrival Rate, $\lambda$, for $a=2.0$, in a 4x4 Manhattan Street Network (MSN) ....................................................... 84

Figure 4.13c: Effect of Buffer Size, b, on the Probability of Deflection, p, as a Function of the Packet Arrival Rate, $\lambda$, for $a=3.0$, in a 4x4 Manhattan Street Network (MSN) ....................................................... 85

Figure 4.14: Variation of the Waiting Time for an M/D/1/(b+1) Queue with Buffer Size, b ................................................................. 91

Figure 4.15a-i Variation of the Total Waiting and Deflection Penalties with Buffer Size, b, and the Packet Arrival Rate, $\lambda$, with the Network Tuning Parameter; $a=1.0$, $b=2$, in a 4x4 Manhattan Street Network (MSN) ....................................................... 92
| Figure 4.15a-ii | Variation of the Total Waiting and Deflection Penalties with Buffer Size, \( b \), and the Packet Arrival Rate, \( \lambda \), with the Network Tuning Parameter; \( a=1.0, b=4 \), in a 4x4 Manhattan Street Network (MSN). ................................................................. 93 |
| Figure 4.15b-i | Variation of the Total Waiting and Deflection Penalties with Buffer Size, \( b \), and the Packet Arrival Rate, \( \lambda \), with the Network Tuning Parameter; \( a=2.0, b=2 \), in a 4x4 Manhattan Street Network (MSN). ................................................................. 94 |
| Figure 4.15b-ii | Variation of the Total Waiting and Deflection Penalties with Buffer Size, \( b \), and the Packet Arrival Rate, \( \lambda \), with the Network Tuning Parameter; \( a=2.0, b=4 \), in a 4x4 Manhattan Street Network (MSN). ................................................................. 95 |
| Figure 4.15c-i | Variation of the Total Waiting and Deflection Penalties with Buffer Size, \( b \), and the Packet Arrival Rate, \( \lambda \), with the Network Tuning Parameter; \( a=3.0, b=2 \), in a 4x4 Manhattan Street Network (MSN). ................................................................. 96 |
| Figure 4.15c-ii | Variation of the Total Waiting and Deflection Penalties with Buffer Size, \( b \), and the Packet Arrival Rate, \( \lambda \), with the Network Tuning Parameter; \( a=3.0, b=4 \), in a 4x4 Manhattan Street Network (MSN). ................................................................. 97 |
| Figure 4.16: | Variation of Different Components of Delay with Buffer Size, \( b \), and the Packet Arrival Rate, \( \lambda \), for Case-A in a 4x4 Manhattan Street Network (MSN) for the Unity Value of the Network Tuning Parameter, \( i.e., a=1.0 \). ................................................................. 98 |
| Figure 5.1a: | Delay Versus Offered Load Characteristics of Large High-Speed Packet-Switched Networks ................................................................. 113 |
Figure 5.1b: Throughput Versus Offered Load Characteristics of Large High-Speed Packet-Switched Networks.................................114

Figure 5.2: A 4x4 Manhattan Street Network (MSN) (adapted from [103]).........134

Figure 5.3: Architecture of a Node of the Manhattan Street Network (MSN) [BUF=Buffer].................................................................135

Figure 5.4: Nodal Structure of a Manhattan Street Network with Arbiters...136

Figure 5.5: The Format of a Packet..............................................................141

Figure 5.6: A Threshold Function to Determine the Priority Category of a Packet Based on its Sequence Number...............................142

Figure 5.7a: A Threshold Function to Determine the Priority Category of a Packet Based on the Residual Hop-Count.................................143

Figure 5.7b: Another Threshold Function to Determine the Priority Category of a Packet Based on the Residual Hop-Count.................................144

Figure 5.8: A Threshold Function to Determine the Priority Category of a Packet Based on the Number of Deflections Suffered by it...............145

Figure 5.9: A Generalized Uni-Modal Quasi-Trapezoidal Threshold Function..146

Figure 5.10: Fuzzy Computation of the Routing Preference Vector of the Incoming Packets.................................................................147

Figure 5.11: Equivalence Between Multi-Level Based Decision and Fuzzy Logic Based Decision...............................................................148

Figure 5.12: Inter-Node Distance Distribution for Uniform Traffic Distribution (UTD) in a 4x4 Manhattan Street Network.........................150

Figure 5.13: Variation of the Mean Packet Transfer Time (MPTT) with the offered Load in a 4x4 Manhattan Street Network (MSN)..................151

Figure 5.14: Variation of Network Throughput with the offered Load in a 4x4 Manhattan Street Network (MSN)........................................153
Figure 5.15a: Variation of the Network Access Delay with the Offered Load in a 4x4 Manhattan Street Network (MSN) when Neural Arbiters are Used.........................................................154

Figure 5.15b: Variation of the Probabilities of Allowing and Denying Network Access with the Offered Load in a 4x4 Manhattan Street Network (MSN) when Neural Arbiters are Used.................................155

Figure 5.16: Effect of Burstiness of the Active Users on the Maximum Allowable Number of Communicating Users in a 4x4 Manhattan Street Network (MSN) when Neural Arbiters are Used.................................156

Figure A-1: A 4x4 Switching Fabric (or Network). The Encircled Segment is Used for Studying the Proposed SC-TC Trade-off (Adapted from [103]).................................................................................................168

Figure A-2: The Exact Queueing Model for the Encircled Segment of Fig.A-1. All the Queues are of M/D/1/K type, where K is the System Size in Number of Packets.................................................................................169

Figure A-3: A Highly Approximate Queueing Model for the Encircled Segment of Fig.A-1. All the Queues are of M/M/1/K type. A Proof of this Approximation is Presented in section A.2.1.1. The Same Results Can Also be Obtained by Applying Elementary Superposition Theorem. The Buffers in Front of the Servers A and B are only to Hold the Packets in Service, i.e., they are serving buffers..........169

Figure A-4: Demonstration of the Fact that Delays Suffered in an M/D/1/K Queue is Always Upper-Bounded by that in an M/M/1/K Queue..............170

Figure A-5: State Transition Diagram for the Queueing System of Fig.A-3.....172

Figure A-6: Effect of Packet Arrival Rate, λ, on the Normalized Mean Nodal Forwarding Time, E(S₀), for the Queueing System Presented in Fig.A-3.................................................................................................173
Figure A-7: Effect of the Parameter, a, on the Normalized Mean Nodal Forwarding Time, $E(S_0)$, for the Queueing System Presented in Fig.A-3

Figure A-8: A Non-Zero Buffer Case. We Consider One Additional Buffer -- Compared to that in Fig.A-3 -- in the Forward Path. Here, the System Size, $K=2$, i.e., Only One Waiting Buffer, and the Other is a Serving Buffer, for the Forward Path. And the system size, $K=1$, i.e., No Waiting Buffer, for the Feedback Path

Figure A-9: State Transition Diagram for the Queueing System of Fig.A-8

Figure A-10: Effect of Packet Arrival Rate, $\lambda$, on the Normalized Mean Nodal Forwarding Time, $E(S_1)$, for the Queueing System Presented in Fig.A-8

Figure A-11: Effect of the Parameter, a, on the Normalized Mean Nodal Forwarding Time, $E(S_1)$, for the Queueing System Presented in Fig.A-8

Figure A-12: Demonstration of the Proposed SC-TC Trade-off. We Consider the Design of a Fabric (or Network) Segment with Zero Buffer whose Delay Related Performance is the Same as that of the Non-Zero Buffer System Considered in Fig.A-8. Note that the Ordinate Represents the Incremental TC Needed in the Feedback Path to Eliminate the Additional Buffer in the Forward Path of Fig.A-8

Figure A-13: Embedding a 2x2 Manhattan Street Network (MSN) on-to a 4-Node Ring Network

Figure A-14: Embedding a 4x4 Manhattan Street Network (MSN) Onto a 16-Node, 6-Channel Ring Network

Figure C-1: An nxm Switch, where for a 3x3 switch, n=3, and m=3

Figure C-2: All Possible 1x1 Switching Patterns (SPs) of a 3x3 Switch
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-3</td>
<td>All Possible 2x2 Switching Patterns (SPs) of a 3x3 Switch</td>
<td>201</td>
</tr>
<tr>
<td>C-4</td>
<td>All Possible 3x3 Switching Patterns (SPs) of a 3x3 Switch</td>
<td>202</td>
</tr>
<tr>
<td>D-1</td>
<td>The Buffer Stealer</td>
<td>204</td>
</tr>
<tr>
<td>E-1</td>
<td>The Maximum Selection Network</td>
<td>206</td>
</tr>
<tr>
<td>E-2</td>
<td>The Minimum Selection Network</td>
<td>208</td>
</tr>
<tr>
<td>E-3a</td>
<td>Type-1 Sorter Using Artificial Neural Networks</td>
<td>209</td>
</tr>
<tr>
<td>E-3bi</td>
<td>Threshold Functions for Sorting (Type-2) Using Artificial Neural Networks</td>
<td>210</td>
</tr>
<tr>
<td>E-3bii</td>
<td>Sorting (Type-2) Using Artificial Neural Networks</td>
<td>211</td>
</tr>
<tr>
<td>F-1</td>
<td>Architecture of the Type-1 Adaptive Resonance Theory (ART) Based Artificial Neural Networks (ANNs)</td>
<td>219</td>
</tr>
</tbody>
</table>
List of Symbols and Abbreviations

This section presents a definition of various symbols and acronyms used in this dissertation.

Symbols

\( \alpha \): Total number of allowable deflections per node \textit{en route} to the destination, also called tuning parameter. For an ideal network, \( \alpha = 0 \).

\( \alpha : \) Possibility with which an adopted decision is correct.

\( b \): Number of waiting buffers at the input of any outgoing link of a node of the network.

\( \beta \): Possibility with which an adopted decision is incorrect.

\( B \): Burstiness ratio of an active source station or Terminal Equipment (TE).

\( \delta_a \): \( 1 \) iff \( a = 0 \), zero otherwise.

\( \binom{n}{c} \): Binomial coefficient, i.e., \([n/((n-c)!c)!]\).

\( d \): The inter-node distance in number of hops, \( d = 1, 2, \ldots, d_{\max} \).

\( d_{\max} \): Diameter of the network under consideration.

\( d_{\mu} \): Minimum mean inter-node distance for the Uniform Traffic Distribution (UTD) pattern.

\( d_{\mu a} \): Actual value of the inter-node distance for the UTD.
$d_{mz}$: Minimum mean inter-node distance for the Zonal Traffic Distribution (ZTD) pattern.

$\delta_{mza}$: Actual value of the inter-node distance for the ZTD.

$D_{\text{avg}}$: Mean packet transmission time in terms of the number of slots for a healthy network operating in the stable region of operation.

$D_{m}$: Maximum value of the mean packet transfer time in terms of the number of slots for a healthy network operating in the stable region of operation.

$\lambda$: Packet arrival rate in number of packets/slot/station of the network.

$M/M/1$: A single-server queueing process where the inter-arrival and service times are exponentially distributed as indicated by the first and second M respectively, service discipline is first-come-first-serve, and the system capacity is infinity.

$M/M/1/K$: An M/M/1 queueing process with system capacity of K buffers.

$M/D/1$: A single-server queueing process where the inter-arrival time is exponentially distributed as indicated by the M, service time is deterministic as indicated by the D, service discipline is first-come-first-serve, and the system capacity is infinity.

$M/D/1/K$: An M/D/1 queueing process where the system capacity is K buffers.

$n$: Total number of deflections suffered by a packet en route to destination, \( n = (a.d) \).

$N$: Total number of nodes in the network.
\(N_{df}\): Mean number of deflections or misdirections suffered by a packet en route to destination, i.e., \(E(n)\).

\(N_d\): Number of nodes \(d\)-hops apart in an idealized version of the network under consideration.

\(N_H\): Total number of nodes contained in a sphere of radius \(H\).

\(p\): Probability of deflection at any node on the shortest path from source to destination.

\(P(X=d)\): Probability that the variable \(X\) assumes a value of \(d\).

\(\varphi\): Penalty in number of hops per misdirection of a packet in the network.

\(\Phi_r(\cdot)\): Threshold function associated with the \(r\)-th rule.

\[\sum_{d=1}^{d_{\text{max}}} \cdot \sqrt{\cdot} : \text{Summation from } d=1 \text{ to } d=d_{\text{max}}.\]

\(\psi\): Possibility with which an adopted decision is neither correct nor incorrect.

\(r\): A variable to indicate the identification number of a rule \((r = 1, 2, 3, \ldots, R)\).

\(\Theta\): Rank of an \(N \times N\) matrix.

\(S_{rk}\): Fuzzy set representing a category "High", "Medium" or "Low" as dictated by the \(r\)-th rule at the \(k\)-th neuron.

\(T_{\text{avg}}\): Average waiting time per node for an M/D/1/(b+1) queue.

\(T(z)\): Topology z-transform or the z-transform of the inter-node distance distribution for a specific traffic distribution (UTD or ZTD) pattern.
\( T_w: \) [\( w.d_{mu} \)] for the UTD pattern, and [\( w.d_{mz} \)] for the ZTD pattern.

\( U: \) Utilization of the network, \( U_{LB} \) stands for the lower bound of utilization and \( U_{avg} \) for the average value of the utilization.

\( w: \) Width i.e., the number of nodes in a row or column of a regular mesh network.

\( W: \) Average waiting time per node in terms of number of hops (or slots) for the M/D/1 queue.

\( W_{ki}: \) Weight associated with the synapse from the \( k\)-th neuron to the \( i\)-th neuron.

\( x: \) Any non-zero positive integer.

\( \mathbf{X}: \) An input vector.

\( Y_r: \) The output vector for the \( r\)-th rule.

\( \zeta: \) The probability with which a source generates packets destined to a specific zone, \( Z. \)

\( \lceil \cdot \rceil: \) The ceiling function. This function extracts the integer portion of the 'argument plus one'.
Abbreviations

Ack.: Acknowledgement.

ANN: Artificial Neural Network.

ART: Adaptive Resonance Theory.

BPS or bps: Bits Per Second.

CC: Connection Configuration.

GBPS/Gbps: Giga i.e., $10^9$ bits per second.

GOS/QOS: Grade/Quality Of Service.

IND: Inter-Node Distance.

ITC: Incremental Transmission Capacity (TC).

LTM: Long Term Memory.

MBPS: Mega i.e., $10^6$ bits per second.

MAN: Metropolitan Area Network.

MEND: MEan Number of Deflections.

Max-MEND: Maximum value of MEND, $(N_{dp})_{max}$.

MHN: Multi-Hop Network.

MIND: Mean IND.
MIMIND: MInimum MIND.

MPTT: Mean Packet Transfer Time.

MSN: Manhattan Street Network.

OAM: Operation, Administration, and Maintenance.

RI: Routing information.

RMN: Regular Mesh Networks.

SC: Storage Capacity.


SIN: Symmetric Interconnection Networks.

SP: Switching Pattern.

STM: Short Term Memory.

TBPS/Tbps: Tera i.e., $10^{12}$ bits per second.

TE: Terminal Equipment.

UTD: Uniform Traffic Distribution.

WAN: Wide Area Network.

WDM: Wavelength Division Multiplexing.

ZTD: Zonal Traffic Distribution.
Chapter 1

INTRODUCTION

In this prefatory chapter, we define the scope of this research and its underlying technological assumptions. First, a brief discussion of future networking trends and the role of performance evaluation in design and efficient operation of large supra-high-speed packet switching networks is presented. Then, we describe novel high-performance network architectures and the required operating characteristics. The term large means an area covered by $\geq 50$ Km diameter circle, and the term supra-high-speed indicates a speed range of $\geq 100$ Mbps. It is also shown how customer demands dictated the developments of fast packet switching techniques in the late 1980s. Finally, an overview of the entire dissertation is presented in a chapter-by-chapter format.

1.1 Future Networking: Trends and Demands

One of the most primitive desires of human beings is communication. Be it for business, or solely for the purpose of entertainment, it is expected [115] that the communication networks will evolve [37, 134] in the future at least at the same pace as they have in the past. Evolution of networks usually accompanies: • Higher transmission and processing speeds, • Coverage of wider geographic areas, and hence supporting larger number of users, • Larger per Terminal Equipment (TE) throughput, • Provision of multiple types, i.e., voice, data, video, facsimile etc., of services, • Higher quality, i.e., lower delay and higher availability, reliability and security, of services at lower price, etc.
Offering of better services at the same or lower price calls for an improvement in the network operations and managements. The major foci of this dissertation are development of new and efficient techniques for evaluating architecture, routing strategy, and enhancing performance using Artificial Neural Networks (ANNs) of a special kind of regular mesh topology known as the Manhattan Street Network (MSN).

Since the user TEs will have the capability to produce multiple types of traffic, the aggregate offered load per TE will be very high compared to the pure data or voice type networks. Therefore, the future large supra-high-speed networks will definitely demand significantly higher levels of throughput which must be achieved without sacrificing the delay criterion. Attention must also be given to maintain the cost per unit of service as low as possible.

It is interesting to note that, with current availability of economically feasible optical fibers -- which offer *bandwidth to burn* -- as transmission media [40, 60, 112, 140], the technological advances in electronic processing are rapidly becoming obsolete [39, 57, 59, 107, 108, 119]. Therefore, it is necessary to redesign the nodal operating systems using very simple and efficient elements so as to match the transmission speed offered by the lightwave links [87]. One may wish to solve this problem replacing the electronic processing by optical processing technology as well. The problem, however, is that the optical computing technology is not yet mature [66, 67, 107, 108, 119] enough to form the processing and control units of the network nodes.

We also notice that the efforts to design large supra-high-speed packet switching networks were initiated during the mid-1980s with the proposals of Metropolitan Area Networks (MANs) like Distributed Queue Dual Bus (DQDB) [113, 126], Fiber Distributed Data Interface (FDDI) [126, 139, 148] etc. These networks use bus and ring as physical or
hard topology, and offer both voice and data services. They span a geographic area greater than or equal to that covered by a 50 Km diameter circle, and use transmission links of speed $\geq 100$ Mbps (10 times higher than that in the original Ethernet).

If the growth of telecommunication networks continues at the present rate, the geopolitical boundaries may no longer be meaningful in various information markets. The values of products and services may become solely dependent on how fast they can be offered to the potential customers. That is, telecommunication is quickly becoming an efficient vehicle for redistribution of capital [115]. Consequently, if along with technological advances, the government regulatory and protectionist interferences are also removed, the world-wide supra-high-speed telecommunication networks (not necessarily owned by one company or country) can become the most potent socio-political force to promote world peace.

To end this section, few points about the role of performance evaluations in network design and operations are stated. Note that the available techniques for evaluating telecommunication networks performance are •Simulation methods, and •Analytical methods.

•Simulation Methods: This technique calls for writing of computer simulations (usually in C, Pascal or other higher level simulation language, or using package$^5$ ) for mimicking the activities of network nodes and links. Either (i) Activity scanning approach, or (ii) Event list based approach, whichever is more suitable for the environment in question, can be used. Simulations -- written in C or Pascal -- are very flexible, since it is easy to incorporate all possible changes and variations in them. Usually, the simulation packages

are not very flexible. In addition, they might need specialized equipments and operating systems for installation. Therefore, we have decided not to use packages for the investigations conducted in this dissertation. Simulation method is sometimes used for validating analytical results as well.

**Analytical Methods:** In this method, it may be necessary to incorporate hypothetical terms or factors or conditions to model all possible actions. Therefore, the developer of analytical model must have a real insight into the scheme being modeled. Sometimes, this approach might result in very complicated models with too little practical significance.

As the technology advances, the underlying assumptions of the previously developed analytical and simulation models become obsolete, and hence, there will always be a demand for performance modeling in the computers and telecommunication network industries. Other reasons for maintaining performance evaluation research groups may include the following: •To study and select the best among the competing techniques or approaches, •To convince the management or governing bodies, and •To get the proposed methods or techniques accepted by the National and/or International standardization organizations, and thus to be a highly competitive player in the marketplace of the corresponding equipments.

1.2 Novel Network Architectures And Their Operating Principles

Current availability of economically feasible optical fibers as transmission media [40, 112] endows the network designers with abundant bandwidth to trade it off with nodal buffer requirement [80], nodal complexity [104], etc. Furthermore, with today’s technology, the achievable transmission speed in lightwave links is so high that the
software-dominated layers of the standard seven-layer Open System Interconnection (OSI) architecture of the International Standardization Organization (ISO) present real bottleneck to the communicating TEs.

Note that most of the software controlled processing (i.e., protocols) overheads include the tasks of:

(i) **Scheduling**, i.e., channel management, path selection etc., for both network access control and transit packet forwarding.

When the soft-topology, i.e., virtual or logical interconnection structure among the nodes, of the network is regular, the information about network interconnection can be easily computed, i.e., there is no need to store them at the nodes. There are several other advantages of using a regular architecture for soft topology of a wide area network. For example: regularity makes the topology algebraically definable, hence simplified routing and flow control [22, 38] techniques can be used; the architecture can be easily tuned to embed the anticipated traffic pattern, and thus improved load balancing and fault-tolerance can be easily achieved; and multicasting and broadcasting can be easily performed. Examples of such regular architectures are -- Manhattan Street Networks (MSNs) [78, 103], Naturally Intelligent Lightwave-communication Networks (NILiNs) [84], etc.

(ii) **Encryption and decryption** of messages for preventing active or passive tapping.

When optical fibers are used as transmission media, it is very difficult to tap the signal without being detected [139].

(iii) **Buffering** of packets at the intermediate nodes.

When deflection routing [79, 83, 104] is used, the communication channels can be used as effective buffers.

(iv) **Retransmission** of erroneous or dropped packets.

When the network nodes are equipped with buffer stealer, the probability of retransmission may be reduced [83].
It is expected that the design of large supra-high-speed packet switching networks will make extensive use of the emerging fine-grained Wave-length Division Multiplexing (WDM) technology [23, 27, 43, 57, 59, 60, 88, 99, 119, 140, 141]. This allows construction of easily extendable, zone-wise regular network structures as shown in [84, 103], and most of the processing to be done in hardware.

WDM also allows the designer to partition the huge aggregate bandwidth of optical links into multiple channels or wave-lengths. Each of these channels has manageable capacity, and hence allows the effect of speed mis-match between transmission and nodal processing to be mitigated to some extent. Also, each TE can have the capability to tune dynamically to one or more wave-lengths in order to fulfill its traffic transportation requirements. This technology coupled with the Fast Packet Switching (FPS) [143, 144] technique (discussed in section 1.3) enables the network designers to consider new architectures which are regular in nature, have very little or no buffers at the nodes, and most importantly can be engineered to deliver message packets to the destination using as little as one hop [83, 84, 116] via distributed cut-through. Few examples of this type of designs are MSN [78, 79, 103, 104], Multi-Hop Networks (MHNs) [1, 35], and NILNs [84]. Essentially, all of them provide higher throughput to a large number of distributed users using lightwave links sharing the features of wavelength agility and tunability of the soft-topology.

1.2.1 Lightwave Networks: An Overview

The lightwave spectrum, i.e., the optical radiation with frequency between 150 THz and 1.5 PHz* represents an enormous amount of bandwidth that could be employed to transmit information at tremendously high speed. Therefore, optical communication is the

*GHz = Giga (10^9) Hz, THz = Tera (10^{12}) Hz, and PHz = Petra (10^{15}) Hz. [Exa = 10^{18}].
most likely technique to be employed in the future large supra-high-speed packet switching networks. The basic components of optical signal processing [60, 72, 73] are:

- Light source, i.e., Light Emitting Diodes (LEDs), laser diodes etc.
- Low-loss optical fibers, e.g., single-mode fibers,
- Light detectors and Connectors, couplers, tunable filters, lenses etc.

Although the lightwave transmission technology has reached its mature stage, i.e., cost-effective deployment of optical fibers is currently a reality, the optical processing technology is still in its infancy. Consequently, although few all-optical networks have been proposed (see e.g., [84] and the references in [81]), it appears that they will not be implementable in the foreseeable future. Note that the factors which determine the feasibility of any technology are: Practically achievable performance; Cost of implementation; and Reliability.

Recent research results [40, 112] have shown that the optical fibers are capable of replacing the long-distance telecommunication links, which proves that the cost and reliability factors of optical transmission technology are improving at a highly desirable rate. However, the optical processing technology is not mature enough for deployment in the real-life large telecommunication networks, either because of the large physical size of the processor or lack of implementable control mechanisms. Once these hurdles are overcome, the parameters which will determine the performance capabilities of a properly engineered lightwave communication networks becomes the bandwidth-distance product of the network.

With today's technology opto-electronic and electro-optic conversions at the nodal processors are mandatory. The achievable speed of electronic processing is in the range of Gbps, which is order of magnitude smaller [80, 83] than that of the achievable optical
transmission speed. Therefore, every single node which may or may not contain a TE of the current lightwave networks represents a real bottleneck entity. These bottleneck entities determine the maximum achievable throughput per user terminal (or TE).

1.2.2 Architectural Issues

From a performance point of view the following are highly desirable architectural features of large supra-high-speed packet switched telecommunication networks:

- It must support massive pipelining, which makes network operation efficient, and massive parallelism, which makes the network tunable and fault-tolerant.
- The network must be partitionable, and upward and downward scalable.
- The topology must be highly reconfigurable, i.e., it must allow dynamic reconfiguration of the soft-topology such that it can be matched to the demanded traffic matrix and pattern as quickly as the available technology can support.
- The network must be easily expandable. When the network topology is regular, extendability can be achieved without much difficulty. Furthermore, regularity in the network organization offers the advantage that all nodes have implicit knowledge about the topology of the network. The nodes can exploit this global information in undertaking routing and scheduling decisions. Regular topologies also provide dense networks with smaller diameter and a large number of alternate paths between any given pair of nodes. Therefore, shortest path routing can be easily achieved [16, 74, 89, 102-104, 116].
- If possible, the architecture should benevolently allow the network to learn its operating rules from the environment. Both off-line training and on-line (or background) learning should be implementable—just as the nervous system of human being keeps watch over blood circulation, respiration, etc.
With reference to the above points it is worthwhile to point out that the high-speed interconnection networks originally designed for massively parallel computing systems [36, 149] may also be suitable for large high-performance packet switching networks. Another important analogy, as mentioned in [116], is that the problem of communication management in large networks is very similar to the classical virtual memory management of uni-processor systems. The most efficient way to achieve this is via dynamic reconfiguration of the logical (or soft or virtual) topology of the network. Once this can be performed, we may no longer need to look at the problem of topological mapping [3, 70, 94]. The desirable characteristics [24, 81] of highly reconfigurable architectures are:

- **Partitionability**: This refers to the ability to create partitions or clusters of resources and interconnections among them. The larger the size of partition in a given network, the lower will be the inter-partition communication overheads and vice versa. Consequently, there may exist an optimal partition size for a given physical topology subject to the demands and distribution patterns of traffic. Kamoun [74] considers such problems in the context of hierarchical routing in large computer communication networks in his doctoral dissertation.

- **Scalability**: An architecture can be upward scalable or downward scalable or both. Upward scalability indicates the cost as a function of the number of physical crossconnects and routing delay. Downward scalability is a measure of attaining the efficiency of smaller networks as per demand. Cost and delay factors for multistage networks are \(O(N \cdot \log N)\) and \(O(\log N)\) respectively [Note: \(N\) refers to the number of nodes in the network], whereas the corresponding figures for regular mesh networks are \(O(N^2)\) and \(O(N)\) respectively. Multi-stage networks are usually not downward scalable, because the communication overhead of the entire network must be incurred from one node to any other node.

- **Modularity**: Modular architectures are simple in design and operation, easily extendable, fault-tolerant, and may be very cost-effective. Multi-stage networks (e.g.,
shuffle-exchange type nets) are amodular, and hence it may not be possible to use the redundency properly in case of faults [42]. Regular networks (e.g., MSNs, NILiNs etc.) are modular and hence easily restructurable.

- **Tunability:** When the network nodes have the capability to tune to channels of different wavelengths for both transmission and reception, the network is called tunable [95]. This feature makes a node capable of transmitting and receiving at its desired capacity during different times of its operation.

1.2.3 Operational Features

As mentioned earlier in this chapter, one of the highly desirable techniques to achieve efficient operation of large supra-high-speed packet switching networks is to reconfigure dynamically the soft-topology to match the traffic demand and pattern as closely as possible. Dynamic reconfiguration also renders network operation gracefully degradable, which is one of the most attractive operational features of any network [24, 116, 135]. Both static and dynamic reconfiguration methods have been discussed in the literature in the context of relevant problems.

- **Static Reconfiguration:** The soft-topology of the network is determined at the onset of execution of a task, and it remains intact for the duration (i.e., a pre-defined period of time) of the task. This technique works well only when the demands and distribution patterns of traffic are accurately predictable in advance. The reconfiguration overhead is nil, since it is done off-line, and this scheme is inherently very efficient. Examples of this type of reconfiguration can be found in [13, 118, 123, 133, 150]. A brief overview of the operational features of these networks is presented in Chapter 5.

- **Dynamic Reconfiguration:** In this case, the soft-topology of the network evolves as the collective behavior of the communication pattern, just as the collective computation mechanism of the Artificial Neural Networks (ANNs) [52, 117, 128]. Thus,
the interconnection pattern allocates network resources to those TEs who need them most. Both centralized and distributed reconfiguration mechanisms can be used.

Centralized controls are usually fault-prone while the decentralized techniques are significantly more complicated and necessitate more overhead. However, the decentralized control techniques are very robust and render network operation more flexible and fault-tolerant. In Chapter 5, we present an ANN based approach for dynamic reconfiguration of virtual topology which also offers congestion-free operation of networks.

1.3 From Circuit Switching to Fast Packet Switching

Early telecommunication networks employed the traditional circuit switching techniques for providing a uni-rate, i.e., 64 Kbps, voice transmission services to the subscribers [58, 130, 139]. In circuit switching, a path (or a route) is established before the outset of traffic transmission, and that path must remain intact throughout the duration of conversation irrespective of whether the end users (i.e., the conversing parties) are in talk or in silence or in idle mode. Figure 1.1(a) shows the circuit switching technique where the arrow from the source TE to the destination TE indicates that the message transfer occurs via one preset hop (path). In a typical conversation, the established channel remains idle for almost 40% of the connection time [17-19]. This technique is therefore not efficient for transmitting bursty (i.e. the periods of activity are very sparsely distributed), asymmetric (i.e., the amount of traffic from sender to receiver differs significantly from that in the other direction), and error sensitive (i.e., a single bit error can spoil the entire message, as in the financial transaction messages) multi-rate data traffic [58].
Figure 1.1: Different Types of Switching, (a) Circuit Switching, and (b) Packet Switching.

These requirements motivated the telecommunication network researchers in the early 1960s to develop the concept of packet switching for highly reliable data transportation via dynamic sharing of network resources, viz., transmission capacity of links, nodal storage and processing capabilities. In packet switching, a fixed size block -- e.g., 1 Kbits or 128 bytes of data -- is transferred at a time. The size of the packets depends on the transmission speed of the channel and the diameter of the network [58, 130, 139]. Each packet contains its own source and destination addresses, other relevant overheads for dynamic packet routing and end-to-end or link-level flow control, and error correcting codes. All these maintenance activities/overheads took advantage of the then maturing electronic storage and processing technologies to implement effective sharing of the costly electronic transmission bandwidth. The bottleneck elements in the network were the communication links. Figure 1.1(b) shows the packet switching technique where the arrows from the source TE to the destination TE indicate that the transfer of packets occurs in a hop-by-hop manner. Packet switching is very similar to the communication using
postal services, where each packet is treated like an individual letter (i.e., an envelope). Packet switching is very efficient for communicating bursty, asymmetric and error-prone traffic [58, 120]. Although packet switching was originally designed for non-time-critical data traffic transportation, it can also be used for communicating real-time traffic as well [29, 145], by exploiting their inherent characteristics. Details of packet switched voice communications can be found in [29, 145]. Burst switching [4, 5] is a form of packet switching where packet lengths are determined instantaneously according to the length of the signal bursts, e.g., voice pitch. Since the entire burst is accommodated in one packet, burst switching may outperform the ordinary packet switching for real-time traffic communications. Relevant investigations along this line can be found in [5] and [100].

Beginning from the early 1980s, potential customers of Metropolitan and Wide area level telecommunication networks started demanding multi-rate, time-critical and non-time-critical communication services from the same network. These applications cover a wide spectrum of communication services, since they range from bps rate telemetry signals to Mega-bps rate full-motion video signals for entertainment, remote medical diagnoses, super-computer communications, etc. These demands stimulated vigorous enthusiasm among switching architecture research communities. As a result, hundreds of novel broad-band and wide-band switching architecture proposals for fast circuit and packet switching started appearing in literature [25, 147].

1.3.1 Nodal Switching Techniques

Nodal switching techniques can be classified broadly into two groups: Single-Stage Switching and Multi-Stage Switching. In single-stage switching, the input traffic appears at the output after one stage delay, e.g., a delay of one slot or packet time, whereas in multi-stage switching, the delay is more than one slot. A single-stage switch is as shown in Figure 1.2. A functional level block diagram of a multi-stage switching is shown in Figure
1.3a. Figure 1.3b shows a multi-stage switch with feedback. Multi-stage switches usually offer fast circuit and packet switching.

![Single Stage Switch Diagram](image)

**Figure 1.2:** A Single Stage Switch.

![Multi-Stage Switch Diagram](image)

**Figure 1.3a:** A Multi-Stage Switch (Functional Level) without Feedback.
Figure 1.3b: A Multi-Stage Switch with Feedback.

Both single and multi-stage switches can have buffers at the inputs only, as used traditionally [58, 65, 130, 139]. Turner [143, 144] first considered internally distributed buffers in the multi-stage switches to achieve higher throughput performance and reduced blocking probability. Finally, output buffering has been considered by Karol et al.[75] and Maxemchuk [104]. Buffering at a combination or all of the aforementioned locations, i.e., input/internal/output etc., has also been considered [25, 65, 147] to improve further the performance of the switch with the expense of additional complexity in the switch maintenance activities.

The concept of introducing a sorting stage in a multi-stage switching network was introduced by Batcher [11] in 1968, followed by a flurry of activities along that line in the subsequent years.

The idea of feeding back or recirculating the contending packets to the corresponding input terminals was first suggested by Huang and Knauer [63]. They incorporated only partial feedback. This idea was later extended to complete or full
feedback, i.e., each output line is connected to one input line, by Hui and Arthurs [64]. Finally, the recirculation of the blocked packet with non-uniform speed links in the reverse paths has been investigated by Khasnabish, Ahmadi, Hlynka, and Shridhar [80] as discussed in Appendix-A. Note that the feedback and sorting mechanisms reduce the probabilities of blocking and deflections of the message packets.

The major motivating factor for incorporating buffers, sorters, and feedback in the multi-stage switches is to improve the switching throughput, as demanded by the fast packet and circuit switching schemes. The details of the currently proposed fast packet switching and fast circuit switching techniques can be found in [2, 30, 65, 141].

1.4 Motivations and Impacts of this Research

As mentioned in section 1.1, the major foci of this dissertation are: development of new and efficient techniques for evaluating architecture, routing strategy, and enhancing performance using Artificial Neural Networks (ANNs) of a special kind of regular mesh topology known as the Manhattan Street Network (MSN). Since the developed techniques are analytical in nature, these can be used to quickly evaluate architectures and routing techniques of arbitrarily large networks. This is necessary for efficient operation of networks via embedding anticipated or demanded traffic matrix onto the existing physical topology. The process of embedding can be performed using either expert systems or ANNs as discussed in Chapter 5.
1.5 Organization of this Dissertation

In Chapter 2, a taxonomy of the approaches adopted so far for large supra-high-speed packet switching networking is presented. A survey of the unclassified literature enables us to classify them into two major groups: (i) Direct or uni-level design method, and, (ii) Indirect or multi-level or hierarchical design approach. Numerous examples of each category is presented. In particular, it is noted that with the availability of economically feasible optical fibers as transmission media, the network bottleneck entities have moved from transmission links to the processing speed of the nodal operating systems. This is exactly opposite to the technological situations of the 1970s and early 1980s.

Chapter 3 presents a new analytical technique for evaluating lightwave network architectures for two different traffic distribution patterns, viz., Uniform Traffic Distributions (UTD) and Zonal Traffic Distributions (ZTD). Evaluating an architecture in the context of our interest calls for determining the Inter-Node Distance (IND) distribution, Mean IND (MIND), topology z-transform, and the penalty per deflection en route to destination.

It is now well-known that with the availability of economically feasible supra-high-speed communication links e.g., optical fibers [40, 60, 112, 140], the network bottleneck entities lie in the nodal processors [39, 57, 59, 107, 108, 119]. Consequently, the researchers in the MAN and WAN arenas started realizing that the packet routing mechanisms in such networks should be kept sufficiently simple in order to fully utilize the available communication bandwidth. Our concern in Chapter 4 is to present a new analytical technique for rapidly evaluating packet routing schemes in MANs or WANs.
consider both UTD and ZTD patterns of traffic distributions in the MSNs to demonstrate the feasibility of the proposed technique.

In Chapter 5, we present a survey of the techniques used for controlling congestion in large high-speed packet switching networks. Since the network bottleneck elements have moved from transmission links to the processing speed of the nodal operating systems, most of the previously (i.e., in the 1970s and early 1980s) proposed network access control and resource management techniques may not be applicable to the emerging large lightwave networks. In light of these facts, we propose an ANN based arbitration for fast and intelligent processing of transit packets, and assigning network access to the local packets. We employ Grossberg's [52] Adaptive Resonance Theory (ART) based neural network -- a two-layer ANN -- to maintain distributed cut-through links between the communicating users, and a buffer stealer (another ANN based controller) to throttle local sources' access to the network during heavy load conditions.

With the incorporation of these additional complexities at the nodal structure, it is possible to utilize most efficiently the available network resources under all possible (both certain and uncertain) operating conditions of a large supra-high-speed packet switching network. As before, we use the MSN as the example soft-topology for our investigations.

A review of the research results of this dissertation, and suggestions for future research works are given in Chapter 6. Simulation and analytical treatment of dynamic reconfiguration of networks with various types of traffic demands and patterns in an integrated environment are left as material for future investigations. By integrated environment, we mean a network supporting data, facsimile, video, and voice etc., types of traffic.
Appendices A to F provide clarifications and proofs of the results used in the main text of this dissertation. These materials are not absolutely essential to the continuity of the main subject, and are included for the sake of completeness only.

1.6 Conclusions

In this introductory chapter, we have presented a brief overview of the architectural, operational and technological assumptions for this investigation along with the future networking trends and demands. A very short summary of the chapters to follow -- in the same sequence as outlined here -- has also been included.
Chapter 2

TAXONOMY OF THE CURRENTLY PROPOSED LARGE SUPRA-HIGH-SPEED PACKET SWITCHING NETWORKS

In this chapter, we present a taxonomy of the approaches adopted so far for designing large supra-high-speed packet-switched networks. A survey of the relevant public domain literature enables us to classify the approaches into two major groups: (i) Direct or uni-level design method, and, (ii) Indirect or multi-level or hierarchical design approach. Numerous examples of each category are presented.

2.0 Introduction

A survey of the approaches proposed up to January 1992 for large supra-high-speed packet switching networks enables us to classify them into two major groups:

• Direct or uni-level design, and
• Indirect or multi-level or hierarchical design.

Attempts to adapt the mature local area and use store-and-forward packet switching networking techniques for providing multiple services over a large geographic area are the motivating factors for the direct design approach. Indirect design method is voted for by the proponents of voice based telecommunication network designers, and those investigators who propose to interconnect multiple Local Area Networks (LANs) using high-speed links
to provide integrated services over a metropolitan or wide area. The standard seven-layer OSI architecture for computer networking as recommended by the ISO, as shown in Figure 2.1, is the platform of this approach. The function of each layer of the OSI protocol stack is described elaborately in [130, 139]. In particular, we note that packet routing is performed in the Network Layer and the congestion control activities can be practised in any one or more of the seven layers.

Note that the main focus of this chapter is to propose a taxonomy of the approaches proposed for large supra-high-speed packet-switched networking. The motivation for this categorisation is to encapsulate the current activities of the researchers in supra-high-speed networks. We also mention very few characteristics of each of the example network mentioned here. References are cited, where appropriate, which provide elaborate discussion.

2.1 Direct or Uni-Level Design

In this approach, the designers attempt to directly utilize the technological advances and the mature LAN and Store-and-Forward networking techniques [28, 74, 86, 120], [130, 139], to provide multiple services over wide geographic areas [109, 111, 138]. Traditional electronic transmission media are being replaced by optical fiber channels [27, 30, 40, 57, 59, 60, 112, 140]. And, since these networks have to provide integrated services over a large geographic area, it is expected that demand-assignment based Media Access Control (MAC) mechanisms [58, 91, 130, 139] with synchronous or asynchronous packet transfer method will dominate this kind of design.
<table>
<thead>
<tr>
<th>Layer</th>
<th>Protocol</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Layer: File Server, Dir. Service etc.</td>
<td>Application Protocol</td>
<td>Data Unit (APDU)</td>
</tr>
<tr>
<td>Presentation Layer</td>
<td>PPDU</td>
<td></td>
</tr>
<tr>
<td>Session Layer</td>
<td>SPDU</td>
<td></td>
</tr>
<tr>
<td>Transport Layer</td>
<td>TPDU</td>
<td></td>
</tr>
<tr>
<td>Network Layer</td>
<td>Packet Level Transmission</td>
<td></td>
</tr>
<tr>
<td>Logical Link Control Sub-Layer (LLC, IEEE 802.2 Standard)</td>
<td>Frame Level Transmission</td>
<td></td>
</tr>
<tr>
<td>Media Access Control (MAC) Sub-Layer</td>
<td>Frame Level Transmission</td>
<td></td>
</tr>
<tr>
<td>Ethernet Standard</td>
<td>IEEE 802.3 Standard (Ethernet)</td>
<td></td>
</tr>
<tr>
<td>Token Bus Standard</td>
<td>IEEE 802.4 Standard (Token Bus)</td>
<td></td>
</tr>
<tr>
<td>Token Ring Standard</td>
<td>IEEE 802.5 Standard (Token Ring)</td>
<td></td>
</tr>
<tr>
<td>QPSX Net Standard</td>
<td>IEEE 802.6 Standard (QPSX Net)</td>
<td></td>
</tr>
<tr>
<td>Physical Layer</td>
<td>Bit Level Transmission</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.1: Standard Seven Layer Architecture for Open System Interconnection (OSI), as Recommended by the International Standardization Organization (ISO), [LLC=Logical Link Control].
Rubin and Baker [127] and Skov [132] discuss the MACs suitable for large high-speed packet switching networks. It has been observed [132] that most of the high performance MACs are so complicated that they may not be easily implementable with the current technology at the physical level. Note that both simulation and analytical techniques are used [27, 97] to evaluate the performance of these MACs. Both multi-channel ring and dual-bus topologies have been proposed as architectures for Direct Adaptation of LANs as being considered by the ANSI and the IEEE 802.6 committee on MANs.

ANSI is currently considering the Fiber Distributed Data Interface (FDDI) network to standardize it as the integrated services MAN. FDDI is a direct adaptation of the IBM's token ring (IEEE 802.5 standard) networks [139, 148], except that in FDDI:

- Counter-rotating multi-channel ring network is used,
- Total bandwidth is partitioned for synchronous and asynchronous services,
- Multiple (up to 9) tokens are provided, and
- The destination station removes the packet/frame from the network whereas in token rings, the source station itself deletes the packet from the channel after it has traversed the entire ring, and thus provides an implicit acknowledgement.

Since the network access is token based, it is basically an asynchronous network. The access delay to this network can be guaranteed to be less than a negotiable upper bound [148]. Performance evaluation of the FDDI MACs using mainly simulation techniques is given in Reference [27].

IEEE 802.6 committee on MANs is actively involved in standardizing the Queued Packet Synchronous eXchange (QPSX) network [113] for multi-service provision in metropolitan areas. Message transmission in QPSX networks occur as slots of data and the network access is on the basis of slot reservation by the ready stations. Although this
network successfully implements a distributed First-Come-First-Serve (FCFS) or First-In-First-Out (FIFO) queue of the ready stations, the issue of fairness in access assignment is still to be resolved. As in FDDI networks, QPSX networks are also capable of offering both real-time and non-real-time services to the users.

A comparison of the performance of FDDI and QPSX networks can be found in [126]. It appears that QPSX offers better delay-throughput characteristics compared to the FDDI over the entire useful range of offered load. Reference [126], however, argues that since these two networks were born out of the research activities of the peoples of two different schools of thought, i.e., data communications and voice communications, they satisfy the needs of their corresponding domains.

Consequently, while FDDI guarantees bounded access delay, the QPSX offers efficient utilization of the network resources over the entire spectrum of offered load.

Let us now consider Direct Adaptation of the Store-and-Forward Networks for the same purpose. Store-and-Forward packet switching Networks (SF-Nets) evolved in the early 1960s along with the advent of the packet switching technique [86, 120, 130, 139]. The basic idea is to transfer the packets of message from the source to destination, in a hop-by-hop manner, by first storing them in the intermediate nodes and then forwarding to the neighboring node using the best outgoing link. In most cases, the chosen outgoing link leads to the shortest path or the least loaded path. The nodes need to maintain and update the routing table for the source-destination pair tagged at the packets' header. Lai [92] presents a survey and classification of the packet forwarding techniques. Both regular and irregular mesh-type topologies have been considered [28, 56, 74, 86, 139] as architectures for SF-Nets. A major problem of these networks is the dynamic
maintenance of the routing table at the nodes for the purpose of congestion and flow controls.

With the current availability of economically feasible optical fiber transmission media [40, 72, 73, 112], it has been realized that large supra-high-speed packet switched networks with regular topologies and simple routing and flow control mechanisms can be used for efficient multi-service provision over wide geographic areas. There are several other advantages of using a regular architecture for virtual (or soft) topology of a wide area network. For example: regularity makes the topology algebraically definable; simplified routing and flow control techniques can be used; the architecture can be tuned to embed the anticipated traffic pattern, and thus improved load balancing and fault-tolerance can be easily achieved; and multicasting and broadcasting can be easily performed. The two most widely discussed examples of such activities are: Multi-Hop Networks (MHNs) [1, 35, 43], and Manhattan Street Networks (MSNs) [78-79, 102-104].

MHN is a logical shuffle-exchange network wrapped-around on the surface of a cylinder [1]. This logical structure is embedded in a physical structure which can be a ring, bus, tree, or mesh. MHN takes complete advantage of the abundant transmission capacity offered by the optical fiber channels. It uses simple routing and flow control techniques, and the nodes contain very simple non-blocking [12, 69] switching fabrics. Since these networks are multi-connected, they are fault-tolerant and can achieve a certain degree of parallelism. They, however, may not be easily extendable or restructurable.

MSN (Figure 2.2) is a class of regular mesh networks where the number of nodes in every row and column is even, the end nodes are connected using wrap-around links, and the adjacent rows and columns transport traffic in opposite directions [102-104].
Figure 2.2: A 4x4 Manhattan Street Network (MSN) (adapted from [103]).

It operates as a slotted system with very small number of nodal buffers. Logically, this structure can be assumed to be embedded in the surface of a sphere [81]. MSN can either be the physical topology, or it can be embedded onto any other physical structures, e.g., a multi-channel ring network, as shown in Appendix-A.

Note that both MHN and MSN are multi-connected and hence highly fault-tolerant, and can achieve higher throughput performance than any other configurations in this class. Although the delay performance of the MHN may be better than that of the MSN, the former may not be as easily extendable as the latter one.
Researchers are still actively investigating various aspects of these two networks (and their enhanced versions), and only time can inform the readers about the true winner.

Other regular mesh networks which have been considered for SF-Nets are:

- Networks with bi-directional links and odd number of nodes in every row and column, as proposed by Kamoun [74] where hierarchical clustering of nodes is considered for the purpose of packet routing.
- Networks with bi-directional links and even number of nodes in every row and column, as proposed by Podar [116], and Borgonovo and Cadorin [16]. Podar shows that pure Zig-Zag forwarding of packets leads to optimal routing in this type of networks.
- Networks with uni-directional links and odd number of nodes in every row and column, as proposed by Robertazzi [124].

2.2 Indirect or Multi-Level or Hierarchical Design

In this approach, the network is physically organized as multiple layers where the lowest layer provides access to the user terminals, and the topmost layer, i.e., the back bone network provides pure message transmission services using broadcast, multicast, store-and-forward packet switching, or circuit-switching techniques. It is also possible to use the TCP/IP (Transport Control Protocol/Internet Protocol) technique to provide connection-less service in the back bone. Different basic networks have evolved out of the investigations by the data-communication and voice-communication networking groups; we present two sub-classes of the indirect design approach in this section.
A first category includes the *Interconnection of High-Speed LANs* (HS-LANs) using suitable backbones. The commonly used basic architectures for local area networks are Bus (Figure 2.3), Ring (Figure 2.4), Star, Tree, and regular or irregular mesh networks (Figure 2.5) [58, 130, 139]. The resulting interconnected networks may be Bus-of-Bus, Bus-of-Ring, Bus-of-Star, Bus-of-Tree, Bus-of-Mesh etc., or, any other homogeneous or heterogeneous combination of the aforementioned basic structures [81].

![Diagram showing various bus topologies for local and wide area networks](image)

**Figure 2.3:** Various Bus Topologies for Local and Wide Area Networks;
(a) Simple Bus Topology, as used in Ethernet (IEEE 802.3 Standard) and Token Bus (IEEE 802.4 Standard) Networks, (b) Dual Bus Topology, as used in the QPSX MAN (IEEE 802.6 Standard), (c) Folded Bus Topology, as Proposed for the Expressnet [91], (d) Headed Bus Topology, as can be used for Physical Topology of a Multi-Hop Network (MHN) [35].
Figure 2.4: Various Ring Topologies for Local and Wide Area Networks; 
(a) Uni-directional Ring Network, as can be used for IBM's 
Token Ring (IEEE 802.5 Standard) Networks, (b) Dual-
channel Ring Topology for FDDI Networks, (c) Chordal 
Ring Network [6], as can be used for Physical Topology to 
Embed Manhattan Street Network (MSN) (Appendix-A and 
[81]).
Figure 2.5: Various Other Topologies for Local and Wide Area Networks; (a) A Regular Mesh Topology, (b) An Irregular Mesh Topology, (c) An Eight-leaf Tree Network, (d) An Eight-node Star Network.
Figure 2.6: Examples of Hierarchical Topologies for Wide Area Networks; (a) An Interconnected Star-Of-Star (SOS) Network, (b) An Interconnected Tree Network.
For example, interconnected tree and Star-Of-Star networks are as shown in Figure 2.6.

Since the most widely used LANs are Xerox's Ethernet (IEEE 802.3 Standard) and IBM's Token Bus/Ring (IEEE 802.4/802.5 Standard) networks, a majority of the current research efforts is directed towards interconnecting these networks. A number of proposals have appeared in the literature [20, 44] which range from complete sharing to complete partitioning of the bandwidth for intra- and inter-network packets/frames. An important aspect of interconnecting heterogeneous local computer networks is protocol conversion and matching at the gateways to-and-from the Back Bone Networks. These add extra overheads to the communication complexity of the system. Gerla et al.[41], Green et al.[44] and Lam [93] discuss these and related issues in detail.

Now, let us consider the approaches towards Upgrading the Existing Telecommunication Networks for providing integrated services over large geographic areas. The plain old public telecommunication networks have historically evolved as hierarchical, i.e., local, zonal, regional, country, continental, inter-continental networks [65]. Consequently, the proposed approach is based on partitioning the existing networks into various segments and upgrading each segment by enhancing both the channel capacity and the switching capability of the node. This is exactly what is being done for Integrated Services Digital Networks (ISDNs) using the CCITT (Comité Consultatif International de Télégraphique et Téléphonique) recommendations which include the SONET (Synchronous Optical NETworks) and ATM (Asynchronous Transfer Mode) standards.

SONET is one of the recently (as of June 1988) agreed upon CCITT recommendations for physical layer interface in optical networks [9, 14, 26, 55, 98, 111]. It defines synchronous frame structure for byte interleave multiplexing of digital traffic.
The basic Synchronous Transport Signal (STS-1) frame consists of 9 rows of 90 bytes (or octets) of data to be transmitted in 125 μsec (note that 1/8 KHz = 125 μsec; 8 KHz is the sampling frequency for PCM coding of voice signal in telephone). Therefore, in Synchronous Digital Hierarchy (SDH), the basic Optical Carrier (OC) transmission rate, namely, OC-1 becomes \((9 \times 90 \times 8 / 125 \times 10^{-6}) = 51.84 \text{ Mbps}\).

For broadband ISDNs, the User Network Interface (UNI) has been defined as OC-3, i.e., 155.52 Mbps rate or OC-12, i.e., 622.08 Mbps rate. And for Network Node Interface (NNI), the rate of \((m \times \text{OC-3})\) Mbps has been adopted (where \(m\) is an integer \(\geq 1\)). The architecture for SONET deployment could be multi-channel ring or any one of the mesh structures as mentioned earlier in this chapter.

ATM is another CCITT recommendation adopted in June 1989 for fast statistical packet switching [26, 62, 98]. It demands very high-speed processing capacity to cope with the asynchronousity of traffic, and to offer highly flexible and efficient packet switching.

ATM can support low bit-rate to high bit-rate, constant bit-rate to bursty, and connection-less to connection-oriented services. The basic cell size has been defined as 53 bytes (or octets) [5 octets header plus 48 octets payload or data]. The header contains all the necessary information for path and channel identifiers, flow control, error control, etc. For example, in a network, the low bit-rate and bursty sources can be ATM multiplexed in an OC-3 channel, and the constant bit-rate sources can be TDMed (Time Division Multiplexed) in an OC-12 channel.
The switching fabrics that can be adapted to satisfy the ATM demands are usually multi-stage, self-routing and with/without feedbacks, as discussed in [2, 25, 26, 61, 62, 65, 107, 108, 119, 141, 143, 144, 147].

Finally, Figure 2.7 provides a pictorial classification of the currently proposed large supra-high-speed packet switching networks as reviewed in this chapter.

![Diagram](image)

**Figure 2.7:** A Taxonomy of the Currently Proposed Large Supra-High-Speed Packet Switching Networks.
2.3 Conclusions and Future Trends

In this chapter, we have presented a taxonomy (Figure 2.7) of the currently proposed large supra-high-speed packet switching networks. We notice that the standardization organizations, such as ANSI, CCITT and ISO always adopt evolutionary approaches, because they want the future networks to be compatible with the existing ones. On the other hand, the majority of the investigators in the academia and a few in the industries like AT&T, IBM etc., are always in favor of revolutionary approaches to achieve the same goals.

Finally, we predict that the future supra-high-speed (i.e., lightwave speed) packet switching networks could most probably use any tree, star, ring, regular or irregular mesh, or any other topologies as those of the SF-Nets of late 1960s or the 1970s as their physical topologies (or hard topologies). Different regular soft-topologies or virtual topologies will be embedded onto the hard topology to satisfy various traffic distribution patterns. Routing would be performed using dynamically (with minimal software control) reconfigurable optical interconnects [66] -- with very few or no nodal buffer -- to cope with the speed of optical fiber transmission media. As a result, the standard seven-layer OSI structure may be shrunk to as low as four-layer [84] architecture. The most important of all, the network congestion would probably be avoided rather than controlled by tuning the soft topology dynamically, as suggested in [81], or via learning of the network access pattern and traffic demand and distribution patterns by the nodal operating system as considered in [62, 84].
Chapter 3

EVALUATION OF NETWORK ARCHITECTURES

This chapter presents a new analytical technique for evaluating lightwave network architectures for two different traffic distribution patterns, viz., Uniform Traffic Distributions (UTD) and Zonal Traffic Distributions (ZTD). Evaluating an architecture in the context of our interest calls for determining the Inter-Node Distance (IND) distribution, Mean IND (MIND), topology z-transform, and the penalty per deflection en route to the destination.

3.0 Introduction

A new analytical technique for evaluating lightwave network architectures for pre-specified traffic distribution in the network structure is presented. Evaluating an architecture in the context of our interest calls for determining the Inter-Node Distance (IND) distribution, Mean IND (MIND), and the topology z-transform, T(z), for a pre-specified version of the network structure and traffic distribution.

We employ a purely analytical method and hence this method can handle networks of any size and it produces the IND distribution as a by-product of the determination of the topology z-transform. Other approaches for determining similar characteristics include a
purely simulation method [116] and a geometric approach [74], which compute the IND
distribution first and then use the results to calculate the topology z-transform.

Note that, the optical fiber transmission links are inherently uni-directional, and
hence, any directed graph (di-graph) -- regular or irregular -- can be used to represent a
lightwave communication network. The optical computing technology is, however, still in
its infancy [108], and therefore it is recommended [1, 16, 35, 102-105, 109, 118, 124]
that simple, regular, and algebraically definable digraphs be used for lightwave
communications. Consequently, in this chapter we restrict our attention to regular digraph
models, for example, the Manhattan Street Network (MSN) [78, 103], of lightwave
communication networks.

3.1 The Problem Statement and the Proposed Technique

As mentioned in section 3.0, evaluating an architecture in the context of our interest
calls for determining the Inter-Node Distance (IND) distribution, Mean IND (MIND), and
the topology z-transform, T(z), for a pre-specified version of the network structure and
traffic distribution. Our main concern is to utilize these results for evaluating packet routing
policies [79], and for devising packet deflection strategies for avoiding network congestion
[83]. Now, the proposed method necessitates the execution of the following steps:

**Step 1:** Partitioning [76] the network into building block(s) of suitable size(s),

**Step 2:** Determining the topological properties of each block independently, and

**Step 3:** Cascading the basic blocks to construct the network of given size, and then
determining the architectural properties of the resultant network.
3.2 The UTD Pattern in the MSNs.

We consider a two-connected 4×4 Regular Mesh Network (RMN), designed for local and metropolitan area communication systems, as an example network (Figure 3.1). This network is popularly known as the MSN [103]. Two different types of MSNs were first identified in [78]. This configuration achieves higher throughputs and supports more source-destination pairs than the traditional loop and bus networks [102-105].

MSN usually contains an even number of nodes in every row and column, the adjacent row links and column links carry traffic in opposite directions, and the end nodes of every row and column are connected together using wrap-around links. For the uniform traffic distribution in such a network, the probability with which a node sends a packet to another node is the same for all nodes of the network except that the event of self-sinking is prohibited. Therefore, an arbitrary source in an N-node MSN selects the destination by tossing an (N-1)-sided fair dice. Thus, for a wxw (w is an even integer) MSN, there are N (=w²) nodes in the net and (N-1) potential destinations for any source node.

Therefore, we obtain, \( P_d = P(\text{distance}=d) = \frac{N_d}{(N-1)} \), where \( N_d \) is the number of source-destination pairs d-hop apart. This gives:

\[
N_d = [(N-1) \cdot P(\text{distance}=d)]
\]  

(3.1)

Or, the mean value of \( d \) for the UTD, i.e., \( d_{\text{mu}} \) (in the subscript of \( d \), the \( m \) stands for mean, and the \( u \) refers to the UTD):

\[
d_{\text{mu}} = \frac{1}{(N-1)} \sum_{d=1}^{d_{\text{max}}}\left[ d \cdot N_d \right]
\]  

(3.2)
Figure 3.1: A 4x4 Manhattan Street Network (MSN) (adapted from [103]).

where d is an integer type variable, d=1, 2, 3, ..., \(d_{\text{max}}\), and \(d_{\text{max}}\) is the maximum inter-node distance (i.e., the diameter) in the network. Consequently, we call this \(d_{\text{mu}}\) the Mean Inter-Node Distance (MIND). This \(d_{\text{mu}}\) can also be referred to as the Minimum MIND (MIMIND), since the MSN can be used as the seed soft-topology for embedding load-matched topology onto the hard (or physical) topology [83]. Now, the topology z-transform, \(T(z)\), is defined [74, 78] as:

\[
T(z) = \frac{1}{(N-1)} \sum_{d=1}^{d_{\text{max}}} \left[ N_d z^d \right] 
\]  

(3.3)
\[ T(z) = P_1 z^1 + P_2 z^2 + \ldots + P_{d_{\text{max}}} z^{d_{\text{max}}} \]

\[ NT(z) = N_1 z^1 + N_2 z^2 + \ldots + N_{d_{\text{max}}} z^{d_{\text{max}}} \]

Therefore, the IND distribution (i.e., the values \(N_1, N_2, \ldots, N_{d_{\text{max}}}\) etc.) can be obtained from \(T(z)\), given the network architecture and the traffic distribution pattern (Note: (i) the = sign is used since \((N-1)\) is being replaced by \(N\), and (ii) in the above development we have assumed the uniform traffic distribution. ZTD is considered in the next section).

For a 2x2 MSN (Figure 3.2a), \(w=2\), \(N=2^2=4\), and \(d_{\text{max}}=2\). In this network, two source-destination pairs (i.e., a to b, and a to d) are one-hop apart. And, two source-destination pairs (a to c via b, and a to d via d) are two-hop apart. (Note: the source itself is being allowed to be a non-zero-hop destination here, and we will take care of this discrepancy in the subsequent sections). Thus we obtain:

\[ NT(z) = 2 z^1 + 2 z^2 \] (3.4a)

Therefore, the diameter of the network which is the highest power of \(z\) in the expression of \(NT(z)\) is 2 hops, as mentioned earlier.

Similarly, for a 2x4 MSN (Figure 3.2b), \(N=8\), and \(d_{\text{max}}=3\), as will be shown soon. We view the 2x4 network as a cascaded version of two 2x2 networks, and incorporate the effect of cascading in the equation for \(NT(z)\) by a multiplication with the factor \((1 + z^{2/2})\), i.e., \((1 + z)\), for the following reasons.
Figure 3.2a: A 2x2 Manhattan Street Network (MSN).

Figure 3.2b: A 2x4 Manhattan Street Network (MSN), Constructed by Cascading TWO 2x2 MSNs.
The multiplication of the exponent of $z$ by the factor 2 is due to the fact that the mean distance from the newly added, i.e., the cascaded, network to the origin of the basic network is two hops, as shown below. From a1 to a, the distance is 2 hops; from b1 to a the distance is 1 hop; from c1 to a, the distance is 2 hops, and from d1 to a the distance is 3 hops. Therefore, the average distance from the cascaded network to the origin of the basic network, i.e., the node a of Figure 3.2b, is $[(8/4) = 2$ hops], as pointed out before. The reason for dividing the exponent of $z$ by 2 is as follows. The number of external output and input links to every node of the MSN is 2, and also the effect of the existence of the wrap-around links in MSN significantly reduces the average inter-node distance. Thus, we find:

$$NT(z) = (2z^1 + 2z^2)(1 + z^{2/2}) = 2z^1 + 4z^2 + 2z^3.$$  \hspace{1cm} (3.4b)

Since the highest power of $z$ in equation (3.4b) is 3, the diameter of a 2x4 MSN is 3 hops. Similarly, it can be shown that for a 4x4 MSN, the expression for NT(z) is:

$$NT(z) = (2z^1 + 2z^2)(1 + z^{2/2})(1 + z^{4/2}) = [2z^1 + 4z^2 + 4z^3 + 4z^4 + 2z^5]$$  \hspace{1cm} (3.4c)

where we have assumed that a 4x4 MSN can be constructed by vertically cascading, i.e., stacking, two 2x4 MSNs. The effect of stacking has very clearly appeared in equation (3.4c). From equation (3.4c) we find that the diameter of a 4x4 MSN is 5 hops as has already been found by simulation [78, 83].

The process of cascading and stacking can be used to derive the expressions of NT(z)s for larger MSNs. For example, the NT(z)s for 8x8, 16x16, and 32x32 MSNs are as given by equations (3.4d), (3.4e), and (3.4f), respectively.

$$NT(z) = [(2z^1 + 4z^2 + 4z^3 + 4z^4 + 2z^5)][(1 + z^2)][(1 + z^2)]$$  \hspace{1cm} (3.4d)

$$NT(z) = [(2z^1 + 4z^2 + 4z^3 + 4z^4 + 2z^5)][(1 + z^2)^2][(1 + z^4)^2]$$  \hspace{1cm} (3.4e)

$$NT(z) = [(2z^1 + 4z^2 + 4z^3 + 4z^4 + 2z^5)][(1 + z^2)^2][(1 + z^4)^2][(1 + z^8)^2]$$  \hspace{1cm} (3.4f)
Finally, since the above method is analytical, it can be very effectively used in any regular network of arbitrary size (number of nodes). We determine the closed form expressions for $d_{mu}$ and $T(z)$ in terms of the network parameter $w$, in the following section.

### 3.2.1 Determination of the MIND and $T(z)$

We identify two types of even integers. Type-1 even integers constitute the series \{2, 6, 10, 14, 18, \ldots\}, and the second type even integers form the sequence \{4, 8, 12, 16, 20, \ldots\}. Thus we can construct two types of MSNs [78].

After extensive experimentation, both analytical - using equations (3.4a), (3.4b), and (3.4c) - and simulation, it has been found [78, 83] that:

$$d_{\text{max}} = \begin{cases} 
  w & \text{when } w \text{ is even and } w = 2(2x+1) \\
  \lceil w+1 \rceil & \text{when } w \text{ is even and } w = 4x 
\end{cases}$$

(3.5)

where $x$ is any non-negative integer, and $w$ is the width of the network as defined earlier in this section.

**Type-1 Networks:** When $w$ is even but $w = 2(2x+1)$, we have by way of induction:

\[
\begin{align*}
N_1 &= 2, \\
N_k &= 4(k-1), \quad 2 \leq k \leq \lfloor (w/2) + 1 \rfloor \\
N_p &= 4(w-p+1), \quad \lfloor (w/2) + 1 \rfloor < p \leq (w-1) \\
N_w &= 2. 
\end{align*}
\]

(3.6)

Note that the actual value of $N_4$ is one less than the value obtained by using the above equalities. This is because the source station is reached after crossing four hops and the source itself 'cannot be' or 'is not intended to be' a potential destination.
We find that:

\[
\text{Mean IND} = d_{mu} = \frac{1}{(N-1)} \sum_{d=1}^{d_{\text{max}}} d.N_d = \frac{((w^3/2) + w^2 - 2w - 2)}{(N - 1)}
\]  
(3.7)

In the limiting case, i.e., as \( w \) approaches infinity, the above expression for mean IND assumes a value of \([(w/2) + 1]\), as can be shown by applying L'Hopital's rule to equation (3.7). Therefore, for very large MSNs, the mean IND is equal to \([(w/2) + 1]\), not \((w/2)\) as found by Maxemchuk [103]. We obtain the following expression for the \( T(z) \).

\[
T(z) = \frac{2[z + z^3 - z^w + 2z^{w+1} + z^{w+2} - 4z^{(w/2)+2} - z^4(1 - z)^2]}{(N - 1)(1 - z)^2}
\]  
(3.8)

**Type-2 Networks:** When \( w \) is even and \( w = 4x \), we have by induction:

\[
\begin{align*}
N_1 &= 2, \\
N_k &= 4(k-1), \quad 2 \leq k \leq [(w/2)] \\
N_{[(w/2)+1]} &= 4([(w/2) - 1], \\
N_{[(w/2)+2]} &= 4([(w/2) - 1], \\
N_p &= 4(w-p+1), \quad [(w/2) + 2] < p \leq w \\
N_{(w+1)} &= 2.
\end{align*}
\]  
(3.9)

As in the previous case, the actual value of \( N_4 \) is one less than that obtained by using the above set of relationships.

The following expression can be easily derived.

\[
\text{Mean IND} = d_{mu} = \frac{1}{(N-1)} \sum_{d=1}^{d_{\text{max}}} d.N_d = \frac{((w^3/2) + w^2 - 4)}{(N - 1)}
\]  
(3.10)
Here again, the MIND asymptotically approaches the value of \([(w/2) + 1]\), not \((w/2)\) as found in References [103, 104].

The following is the expression for \(T(z)\) in this case.

\[
T(z) = \frac{2z[1 + z^2](1 - z(1/w))^2 - z^4(1 - z)^2}{(N - 1)(1 - z)^2}
\]

(3.11)

3.3 The ZTD Pattern in the MSNs

In case of ZTD, a node (or a station) sends a packet to other nodes of a specific zone \(Z\), with some probability \(\zeta\) and to the nodes outside that zone with probability \((1 - \zeta)\). The motivation behind considering such traffic distribution pattern is the fact that each node of a Regular Mesh Network (RMN) belongs to one horizontal ring and one vertical ring [82]. Assuming that \(H\) is the maximum number of hops a packet can cross and still remain in the zone \(Z\), we can without loss of generality consider that \(H\) is the radius of a sphere at whose center the source is located. The number of nodes contained in the sphere, \(N_H\) is thus:

\[
N_H = \sum_{d=1}^{H} N_d
\]

(3.12)

where \(N_d\) is the number of nodes reachable in exactly \(d\) hops. Now, since the RMN under consideration is a Symmetric Interconnection Network (SIN), each node is contained in the localities of \(N_H\) other nodes and is outside the localities of \([N - 1 - N_H]\) nodes. Consequently, a uniform traffic distribution is created on the communication links even though the routing distribution is non-uniform.
Therefore, we have:

\[
P(\text{distance}=d) = \begin{cases} 
\frac{\zeta N_d}{N_H} \quad &\text{when } 1 \leq d \leq H \\
\frac{(1 - \zeta) N_d}{(N - 1 - N_H)} \quad &\text{when } H < d \leq d_{\text{max}}
\end{cases}
\]

(3.13)

3.3.1 Determination of the MIND and T(z).

In this section, we determine the MIND, and the topology z-transform, T(z), for zonal traffic distribution in two types of MSNs. The same definitions as used in section 3.2 are utilized here. The MIND for ZTD, i.e., \(d_{\text{mz}}\) becomes:

\[
d_{\text{mz}} = \sum_{d=1}^{d_{\text{max}}} [d \cdot P(\text{distance}=d)] = \frac{\zeta}{N_H} \sum_{d=1}^{H} [d \cdot N_d] + \frac{(1 - \zeta)}{(N - 1 - N_H)} \sum_{d=(H+1)}^{d_{\text{max}}} [d \cdot N_d]
\]

After some algebraic manipulations, we obtain the following:

\[
d_{\text{mz}} = \frac{\zeta(N - 1) - N_H}{N_H(N - 1 - N_H)} \sum_{d=1}^{H} [d \cdot N_d] + \frac{(1 - \zeta) \cdot d_{\text{m}}}{(N - 1 - N_H)}
\]

(3.14)

where \(d_{\text{m}} = \sum_{d=1}^{d_{\text{max}}} [d \cdot N_d]\)

For example, for the ZTD pattern in a 36-node (i.e., type-1) MSN, \(N=36, w=6,\) and \(H=(w - 1)=5,\) hence, \(N_H = (N_1 + \ldots + N_5) = (2+4+8+11+8) = 33,\) and \(N_w = N_6 = 2.\)

Using equations (3.6) and (3.13), \(P_w = P(\text{distance} = w) = (2/2)(1 - \zeta) = (1 - \zeta),\) \(d_{\text{m}} = 130,\) \(d_{\text{mu}} = [130/(N - 1)] = 3.72,\) and \(d_{\text{mz}} = \{6.0 - 2.424 \zeta\}.\)
Similarly, for the ZTD pattern in a 16-node (i.e., type-2) MSN, \( N=16, w=4 \), and \( H=(w-1)=3 \), hence, \( N_H=(N_1+N_2+N_3)=(2+4+4)=10, \ N_w=N_4=3, \ N_{w+1}=N_5=2 \). Using equations (3.9) and (3.13), \( p_w=p(\text{distance} = w)=[(2/5)(1 - \zeta)] = 0.4(1 - \zeta) \), \( d_m=44, \ d_{mu}=[44/(N - 1)]=2.93 \), and \( d_{mz}=[4.4 - 2.2 \zeta] \).

The topology z-transform, for the ZTD pattern, can now be easily computed [82] using the same technique as utilized in section 3.2.1.

### 3.4 Conclusions

In this chapter, we have presented a new analytical technique for evaluating lightwave network architectures for two different traffic distribution patterns. The effectiveness of the proposed approach has been demonstrated using the Manhattan Street Network [78, 103] architecture as an example topology.

Our method (i) can be easily applied to a network of any size, (ii) directly produces the topology z-transform, \( T(z) \), an important performance evaluation tool [74, 79] for a given traffic distribution pattern in the network, and (iii) gives the inter-node distance distribution as a by-product of analytical computation of \( T(z) \).

We have also shown that the architectural properties are very much dependent on the traffic distribution patterns in the network.
Chapter 4

EVALUATION OF PACKET ROUTING

In this chapter, we present a simple analytical technique with good degree of accuracy for evaluating simple packet routing schemes in metropolitan or wide area networks. We conjecture that the Basket, Chandy, Muntz, and Palacios (BCMP)-type models [10, 110, 142] are not quite capable of capturing the dynamics of such routing strategy, because the underlying assumptions (e.g., identical and exponential service time distributions) of the BCMP models do not apply to the situations considered here. Furthermore, we notice that the technique proposed in [121] may be applicable to our case only when the feature of limited buffer space is incorporated.

Note that our main objective is to develop a simple analytical method for quick evaluation of simple packet routing techniques.

4.0 Introduction

We deal with the evaluation of a packet routing policy proposed by Maxemchuk [104] for a class of Metropolitan Area Networks (MANs) called the Manhattan Street Networks (MSNs). MSN can also be used as a digraph equivalent of any other supra-high-speed MANs or Wide Area Networks (WANs).
In this context, **Routing** is concerned with determining whether or not the current node is the destination node for a given packet and, finding and assigning an outgoing link to the next neighboring node *en route* to the intended destination. The former function is usually performed by checking the destination address in the packet's header, and the latter one can be accomplished by using a routing table at the nodes or by utilizing the Routing Information (RI) at the packet's header.

Table based routing techniques necessitate storing and updating of routing information at every network node, thereby requiring storage and exchange of (nodes and links) status information throughout the entire network.

Efficient RI based routing policies can be derived if the MANs have some regularity in the structures, e.g., when regular rectangular lattices, regular mesh networks etc. are used as MANs, the desired path can be computed at the intermediate nodes using very simple closed form mathematical expressions. Usually, multiple shortest path routes exist in these networks, and by devising suitable *node addressing schemes*, it is possible to eliminate the routing tables at the nodes. Each node directs traffic towards greater or lesser numbered rows and columns based on the RI at packet's header and simple mathematical computations.

The most attractive features of Regular Mesh Networks (RMNs) are: (i) extendibility, (ii) fault-tolerance capability, (iii) existence of multiple routes for any given source-destination pair, and (iv) coverage over a wide geographic area.

Maxemchuk has in an interesting article [102] provided an extensive discussion of the features of RMNs from the point of view of its utilization as the architecture for MANs.
As the originator of the Manhattan Street Networks (MSNs), Maxemchuk presented his concepts at the 1985 GlobeCom conference [103]. In these networks, each link is unidirectional and the number of rows and columns are even. Adjacent links (rows or columns) transmit packets in opposite directions and the end nodes are connected through wrap-around links. A graph-theoretic interpretation of such a network is presented in Figure 4.1. Although RMNs with odd numbers of rows and columns and with half/full-duplex links have been proposed by various researchers [8, 16, 74, 124], the main focus of this chapter is the evaluation of the first routing policy proposed by Maxemchuk in his article [104].

Figure 4.1: A 4x4 Manhattan Street Network (MSN) (adapted from [103]).
4.1 The Routing Strategy to be Evaluated

The routing strategy to be evaluated has the following characteristics:

(i) Network nodes operate as discrete-time slotted systems.

(ii) Each node receives at most two packets at the beginning of a slot.

(iii) Based on some pre-specified criteria one packet is routed through the preferred outgoing link. The other packet is routed through the available outgoing link, which could be a deflection or mis-direction depending on the packets preference for output link.

(iv) A new packet is introduced into the network only if an output link is available, or when an empty slot is available in any of the input links.

Usually, a routing policy is considered to be optimum when (i) mean packet transfer time is minimized thereby maximizing the effective utilization of network resources, (ii) local and global network congestion is avoided by dropping packets from the network after they have travelled a prescribed number of hops (some researchers believe that this task should be performed by the flow control devices), and (iii) network reliability is improved by exploiting the inherent redundancy in the interconnection structure of the network.

The proposed technique for evaluating a routing rule is based on determining the Inter-Node Distance (IND) distribution and the topology z-transform for an ideal version of the network with a specific traffic distribution pattern. Deviations from the ideal conditions are taken into account by incorporating a tuning parameter and a penalty indicator in the performance equations. The value of the tuning parameter is zero for the ideal network.
The tuning parameter provides flexibility to the users in order to send packets to the intended node even when multiple links (or nodes) of the network are in non-operating mode. The penalty indicator depends on the network's topology and the type, i.e., uni-directional or bi-directional, of communication links used in the network.

4.2 A Step-Wise Description of the Proposed Technique

In this section, we present a step-wise description of the proposed procedure for determining the maximum value of the mean packet transfer time ($D_m$), and the mean (or average) packet transfer time ($D_{avg}$) in terms of the number of hops.

Step 1: Given the network topology, devise an addressing scheme such that the network becomes a Symmetric Interconnection Network (SIN), i.e., all nodes possess the same view of the network. For MSN this is performed using the technique presented in [104].

Step 2: Determine the Inter-Node Distance (IND) distribution for an idealized version of the network for any specific traffic distribution (i.e., UTD or ZTD) pattern. Either simulation method or analytical technique, e.g., the signal flow graph method as used in [78] and [146], or the technique presented in Chapter 3, can be utilized for this purpose. Also, find the average packet transfer time in terms of the number of hops for the traffic distribution pattern under consideration. We call this the Minimum Mean IND (MIMIND) and represent it by $D_m(a=0)$.

Step 3: Determine the topology z-transform, $T(z)$, i.e., the z-transform of the IND distribution obtained in step 2.
Step 4: Determine the value of the penalty per deflection, \( \phi \), for an idealized version of the network under consideration. \( \phi \) is defined as the number of additional hops a packet needs to travel for any misdirection (or deflection) suffered by it.

Step 5: Obtain an analytical expression to represent the number of deflections, \( N_{df} \), for a pre-defined network access control scheme in terms of (i) the \( T(z) \) as obtained in step 3, (ii) the tuning parameter, \( a \), (iii) the network parameter, \( w \), and (iv) the probability of deflection, \( p \). Also find the Max-MEND, \( (N_{df})_{\text{max}} \).

Step 6: Find the exact relationship among the following parameters: \( a \), \( b \), \( \lambda \) (packet arrival rate per slot per station) and \( p \).

Step 7: Compute \( D_m \) and \( D_{\text{avg}} \) using the following relationships, and plot their variations with \( a \), \( b \), \( \lambda \), and \( p \).

\[
D_m(a) = D_m(a=0) + \{\phi (N_{df})_{\text{max}} (a) + (\delta_a T_w)\} \tag{4.1}
\]

\[
D_{\text{avg}}(a, b, \lambda) = D_m(a=0) + \{(D_m(a=0) - 1.0) T_{\text{avg}}\}
+ \{(\phi_{\text{avg}} (N_{df})_{\text{avg}} (a, b, \lambda)) (1.0 + T_{\text{avg}})\} \tag{4.2}
\]

The first term in equation (4.1) is the MIMIND which depends on the network size \( w \), and the second term represents the penalty due to deflection or waiting but not both. Similarly, the first term in equation (4.2) is the MIMIND, the second term represents the waiting time in an ideal network for Case-A (as described below), and the third term is the penalty due to deflections plus the additional waiting time because of those deflections.

We consider two different network access control strategies for packets from the dormant station. In the first case (Case-A), a new packet leaves the source station only
when the desired outgoing link is available to it. In the other case (Case-B), a new packet is put into the network whenever an empty slot is available in any one of the links incoming to the local node (or station).

We note that an equation similar to (4.1) has also been used by Greenberg and Goodman [45] for computing the $D_m$. The authors in [45], however, use a different technique for computing the $(N_{df})_{\text{max}}$. Our technique can estimate both the upper bound and the mean of the packet transfer time.

Finally, we have executed up to the 4-th step of the above-mentioned procedure in Chapter 3.

4.3 A Model for Determining the $(N_{df})_{\text{max}}$ in the MSNs

In this section, we present a model to derive the expressions for determining the Mean Number of Deflections (MEND) suffered by a packet. Therefore, our concern here is the 5-th step of the algorithm presented in the previous section.

Note that each deflection adds $\varphi$ hops to the IND. We assume that the total number of deflections is uniformly distributed among the routing decision points (i.e., the nodes) and $p$ is the probability of deflection at any node from the source to the destination.

Now, as defined in section 4.2, two different cases may arise. In the first case (Case-A), a new packet leaves the source station only when the desired outgoing link is available to it. In Case-B, a new packet is put into the network whenever an empty slot is available in any one of the links incoming to the nodes.
Note that, although the network access delay may be lower in case-B, this method may lead to higher Mean Packet Transfer Time (MPTT) and hence, lower effective network utilization.

Thus, for a d-hop \((d = 1, 2, \ldots, d_{\text{max}})\) route, the maximum number of deflections per node is \([n/(d-1)]\), for case-A, and \([n/d]\), for case-B. These two cases are illustrated in Figures 4.2a and 4.2b, respectively. The moves with one or more deflections to reach the destination are as shown in Figure 4.3, for a 16-node MSN.

The value of the \(N_{df}\) can be determined using the following formula:

\[
N_{df} = \sum_{n=1}^{\infty} [n \cdot P(\text{No. of Deflections} = n)]
\]

Now, the probability that a packet suffers \(n\) deflections is the same as the probability that a d-hop IND is covered in \((d+\varphi n)\) hops. For case-A, at every other node except for the source node, the total number of deflections is \([n/(d-1)]\). Thus, the first deflection at a node is followed by \([n/(d-1)]\) deflections at the same node out of \([n/(d-1)]\) attempts. The last attempt provides a success, i.e., a non-deflection move.
Figure 4.2a: An Illustration of the Case-A for d=4 Hops.

Figure 4.2b: An Illustration of the Case-B for d=4 Hops.
Total 2 Deflections:..................DF--NP--NP--P--P-->

No Deflection:.......P-->

Total 1 Deflection:...DF--NP--NP--P--P-->

Total 2 Deflections:..................DF--NP--NP--NP--P--P-->

Figure 4.3a

Total 3 Deflections:........................................................................DF--NP--NP--P--P-->

Total 2 Deflections:........................................DF--NP--NP--NP--P--P-->

Total 1 Deflection:......DF--NP--NP--P--P-->

No Deflection:............P--P-->

Total 1 Deflection:...DF--NP--NP--NP--P--P-->

Total 2 Deflections:.....................DF--NP--NP--P--P-->

Total 3 Deflections:........................................................................DF--NP--NP--NP--P--P-->

Figure 4.3b

Total 2 Deflections:.....................DF--NP--NP--NP--P--P-->

Total 1 Deflection:............DF--NP--NP--P--P-->

No Deflection:............NP--P--P-->

Total 1 Deflection:......DF--NP--NP--NP--P--P-->

Total 2 Deflections:.....................DF--NP--NP--P--P-->

Figure 4.3c
Figure 4.3d

Figure 4.3e

Figure 4.3: Moves with Deflection(s) to Reach the Destination in a 4x4 Manhattan Street Network (MSN) for (a) 1-Hop, (b) 2-Hop, (c) 3-Hop, (d) 4-Hop, (e) 5-Hop Source-Destination Patterns, [Legend: P=Preference Exists, NP=No Preference Exists, DF=Deflection Suffered, -->=Destination Reached].
Let us assume identical and independent occurrence of the above phenomenon \((d-1)\) times for successful delivery of the packet to the destination for the case when \(\text{IND}=d\). Mathematically speaking, we can express the \(P(\text{No. of deflections} = n)\) as follows.

\[
\sum_{d=1}^{d_{\text{max}}} \left[ P(\text{distance}=(d-1)) \left[ n/(d-1) \right]^{(d-1)} (1-p) p^{(n-d+1)/(d-1)} \right]^{(d-1)} \\
= \sum_{d=1}^{d_{\text{max}}} \left[ P(\text{distance}=(d-1)) \left[ n/(d-1) \right]^{(d-1)} p^n (1-p)^{(d-1)} \right] \\
= \sum_{d=1}^{d_{\text{max}}} \left[ P(\text{distance}=(d-1)) \left[ q/(d-1) \right]^{(d-1)} p^n \right], \quad \text{where } q=(1-p).
\]

Therefore, for case-A:

\[
(N_{df})_A = \sum_{n=1}^{\infty} \left[ n \sum_{d=1}^{d_{\text{max}}} \left[ P(\text{distance}=(d-1)) \left[ q/(d-1) \right]^{(d-1)} p^n \right] \right].
\]

Proceeding in a similar manner, we obtain, for case-B:

\[
(N_{df})_B = \sum_{n=1}^{\infty} \left[ n \sum_{d=1}^{d_{\text{max}}} \left[ P(\text{distance}=(d-1)) \left[ q/d \right]^{d} p^n \right] \right].
\]

Now, we assume that the header of each packet has a deflection counter and that a packet with \(\text{IND}=d\) is allowed to suffer at most \((a_d)\) deflections. The maximum allowable number of deflections becomes \((a_d d_{\text{max}})\), where \(a\) is a positive integer called the tuning parameter. The underlying philosophy for this assumption is the fact that a packet 'is never allowed' or 'should never be allowed' to circulate in the network forever. And that, the allowable number of deflections should be directly proportional to the packet's minimum
path length in terms of the number of hops. This allows a packet to reach the destination even when the network load is high or when the network is faulty. The end-to-end retransmission mechanisms provide further protection by retransmitting the dropped packets.

With the above explanations we get the following expressions for the mean number of deflections (MENDs). In general, when $A=[a \ p \ \{p\}]$, we obtain for the delayed access scheme (i.e., case-A):

$$\langle N_{df}\rangle_A = \sum_{d=1}^{d_{\max}} \left[ a(d-1) \sum_{d=1}^{d_{\max}} \left[ P(\text{distance}=(d-1)) A^{(d-1)} \right] \right].$$

For Type-1 MSNs, $w=2(2x+1)$ and $d_{\max}=w$. Thus, for case-A,

$$\langle N_{df}\rangle_{1A} = \frac{a \ w \ (w - 1) [T(A)_1 - P(d=w) A^{w}]}{2} \quad (4.3)$$

Similarly, for case-B,

$$\langle N_{df}\rangle_B = \sum_{d=1}^{d_{\max}} \left[ a \ d \sum_{d=1}^{d_{\max}} \left[ P(\text{distance}=d) A^d \right] \right].$$

Therefore,

$$\langle N_{df}\rangle_{1B} = \frac{a \ w \ (w +1) T(A)_1}{2} \quad (4.4)$$

where $T(A)_1$ is the topology z-transform for type-1 MSNs for the traffic distribution pattern under consideration. For example, in case of the UTD, we have from Chapter 3:
\[ T(A)_1 = \frac{2[A + A^3 - A^w + 2A^{w+1} + A^{w+2} - 4A^{(w/2)+2}] - A^4 (1 - A)^2}{(N - 1)(1 - A)^2} \]  

And, using the results of section 3.3, we find that for the case of ZTD pattern in a 36-node MSN:

\[ T(A)_1 = A^6 + \frac{\zeta [2A + 4A^2 + 8A^3 + 11A^4 + 8A^5 - 33A^6]}{33.0} \]  

Note that the parameter \( \zeta \) in equations (4.6) represents the probability with which a station generates a packet destined to a specific zone \( Z \), as defined in section 3.3.

For Type-2 MSNs, \( w = 2(2x) = 4x \) and \( d_{\text{max}} = w \). Hence, for case-A,

\[ (N_{\text{df}})_{2A} = \frac{a w(w + 1) T(A)_2 - P(d=w+1) A^{w+1}}{2} \]  

Similarly, for case-B,

\[ (N_{\text{df}})_{2B} = \frac{a(w + 1)(w + 2) T(A)_2}{2} \]  

where \( T(A)_2 \) is the topology z-transform for type-2 MSNs for the traffic distribution pattern under consideration. For example, in case of the UTD, we have from the previous chapter:

\[ T(A)_2 = \frac{2A [(1 + A^2)(1 - A^{(w/2)})^2] - A^4 (1 - A)^2}{(N - 1)(1 - A)^2} \]  

And, using the results of section 3.3, we find that for the ZTD pattern in a 16-node MSN:

\[ T(A)_2 = (0.6A^4 + 0.4A^5) + \zeta [0.2A + 0.4A^2 + 0.4A^3 - 0.6A^4 - 0.4A^5] \]
Note that the parameter $\zeta$ in equations (4.10) represents the probability with which a station generates a packet destined to a specific zone $Z$, as defined in section 3.3.

4.4 Performance Evaluation

The two most important measures of performance of a MAN or WAN are its utilization and the mean packet transfer time (MPTT). The choice of routing strategy significantly affects these two parameters. MPTT is the time difference between the instant when a packet is put into the network and the moment when it is received at the destination. Utilization of a network at a given load refers to the ratio between the MPTT at no load to the MPTT at that particular load.

In this section, we discuss the last two steps, i.e., step 6 and step 7 of the algorithm presented in section 4.2. First, we investigate the variation of network utilization $U$ with the tuning parameter $a$ and then the effects of variations of other parameters of interest on the MPTT are examined in detail. As expected, at low loads the effective network utilization is high and the MPTT is low, and therefore, the tuning parameter does not play any significant role. At high loads, however, the net offered load can be controlled by varying the tuning parameter, and thereby reducing the effective utilization of the network which results in lower MPTT.

4.4.1 Variation of $U$ with $a$

The tuning parameter $a$ as defined in sections 4.2 and 4.3 is the number of allowable deflection(s) per node en route to the destination. This allowance is required because the network is not ideal. In a non-ideal network, neither the links nor the buffer space is of infinite capacity, and there may exist few faulty nodes or links. As mentioned in
section 4.3, the total number of deflections a packet is allowed to suffer before it is dropped from the network is proportional to the distance between source and destination.

Therefore, for the UTD pattern, the actual expression for \( d_{mu} \), i.e., \( d_{mu}^{a} \) becomes:

\[
d_{mu}^{a} = [d_{mu} + (\varphi N_{dp})]
\]

Or, \( d_{mu}^{a} \leq [d_{mu} + (\varphi (N_{dp})_{max})] \)

In the above equations, we have neglected the waiting time in buffers, since the kind of networks being considered here, i.e., the MSNs, are expected to be built using supra-high-speed links with very few or no buffer at the nodes [45, 104]. Consequently, the effective utilization of the network which is the utilization of the links becomes:

\[
U = \frac{d_{mu}}{d_{mu}^{a}} = \frac{d_{mu}}{[d_{mu} + (a \varphi d_{mu})]} = \frac{1}{[1 + (a \varphi)]} = \frac{100}{[1 + (a \varphi)]} \%
\]

The Lower Bound (LB) of utilization is obtained when \( \varphi \) is replaced by \( \varphi_{max} \) which is \textit{four} for the MSNs of any size. Thus,

\[
U_{LB} = \frac{100}{[1 + (4a)]} \%
\]

Similarly, the average value of the utilization is obtained when \( \varphi = \varphi_{avg} = 2 \) hops, for the MSNs. Therefore,

\[
U_{avg} = \frac{100}{[1 + (2a)]} \%
\]

The variation of \( U_{LB} \) and \( U_{avg} \) are plotted against the parameter \( a \) in Figure 4.4.
Figure 4.4: Variation of Network Utilization, $U$, with the Tuning Parameter, $a$, for the Uniform Traffic Distribution (UTD) Pattern.
We note that the effective utilization of the network depends on the (i) type, i.e., unidirectional or bi-directional, of link used, and (ii) network load control or tuning parameter \( a \). Similar results also hold for the ZTD pattern of traffic distribution.

### 4.4.2 Variation of \( D_{\text{avg}} \)

The average value of the packet transfer time consists of the following three components: minimum mean inter-node distance, waiting time and deflection time. In the sequel, we investigate the variation of these three components against all the parameters of practical interest.

#### 4.4.2.1 Variation of \( N_{\text{df}} \) and \( D_{\text{m}} \) with \( a \) and \( p \)

In this section, we examine the variation of \( N_{\text{df}} \) and \( D_{\text{m}} \) with the parameters \( a \) and \( p \). First, the variation of \( N_{\text{df}} \) for a 36-node MSN (a type-1 network) for the uniform traffic distribution is studied using equations (4.3) and (4.4). These are plotted in Figures 4.5a and 4.5b, respectively.

We observe that \( N_{\text{df}} \) increases with an increase in the value of the probability of deflection up to a certain point (where the peak occurs), after that it decreases with the increase of \( p \). The negative slope regions of these curves represent the unstable zones of operation of the network; the network should be prevented from operating in such zones either by dropping the packets or by restricting the access of packets from dormant users. The scheme considered in this chapter achieves this primarily via access restriction.
Figure 4.5a: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, $p$, in a Type-1 MSN (a 36-Node Network) for Case-A for the Uniform Traffic Distribution (UTD) Pattern.
Figure 4.5b: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, $p$, in a Type-1 MSN (a 36-Node Network) for Case-B for the Uniform Traffic Distribution (UTD) Pattern.

An intuitive expectation is that the peaks of $N_{df}$ will increase with an increase in $a$, and that the peaks for higher values of $a$ will occur at higher values of $p$. These features are very well-depicted in these curves. We also find that in case-A (i.e., equation (4.3) and Figure 4.5a) the maximum value of the $N_{df}$ is always lower than that in case-B (i.e., equation (4.4) and Figure 4.5b). This is because the number of nodes at which a packet can suffer deflection in case-A is at least one less than that in case-B.
Furthermore, we observe that as the offered load increases, the probability of packet deflection at a node for the limited buffer case increases. This causes the total number of deflections suffered by a packet to rise, and consequently, the effective network utilization diminishes. The tuning parameter \( a \) which controls the total number of allowable deflections has two implications: (i) it explicitly defines the number of deflections suffered by a packet per node *en route* to the destination, and (ii) it is implicitly related to the size of the buffer at the input of the outgoing links at a node.

It is observed that an inverse relationship exists between the tuning parameter, \( a \) and the buffer size, \( b \). The reason is as follows. When a large number of buffers is available at the nodes, a packet never suffers from deflection. Consequently, the largest \( \text{IND} \) becomes \( w \), which is the diameter for the type-1 network. The corresponding \( D_m \) becomes \( [D_m(a=0) + T_w] \).

Similar performance curves for the UTD pattern in a type-2 network, i.e., a 16-node MSN, are depicted in Figures 4.6a and 4.6b using equations (4.7) and (4.8), respectively.

Observing Figures 4.5a through 4.6b, we find that an inverse relationship exists between \( (N_{df})_{\text{max}} \) and \( N \), the size of the network, for all values of \( a \) and \( p \). Therefore, for a very large MSN, i.e., for metropolitan or wide area coverage, the peak value of the \( N_{df} \) can be expected to be very small, even when the network has only a small number of buffer at every node. This is one of the most desirable features for the metropolitan and wide area networks. And, in the case of MSNs, this occurs due to the existence of multiple shortest path routes for every possible source-destination pair. *Furthermore, the number of equal-length paths per source-destination pattern increases with an increase in the network size.*
Figures 4.7 and 4.8 show the variation of $D_m$ with $a$ and $p$. $D_m$ has been computed, as shown in Table 4.1, using the expression $[D_m(a=0) + (4 (N_{dl})_{max})]$, i.e., by neglecting the packet waiting time at the intermediate nodes. This is because we are now considering very low or no buffer case which is quite justified since it is expected that the practical versions of the $MSNs$ will have a very small number of buffers at every node [104].

Figure 4.6a: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, $p$, in a Type-2 MSN (a 16-Node Network) for Case-A for the Uniform Traffic Distribution (UTD) Pattern.
Figure 4.6b: Variation of the Mean Number of Deflections (MENDs) with the Probability of Deflections, $p$, in a Type-2 MSN (a 16-Node Network) for Case-B for the Uniform Traffic Distribution (UTD) Pattern.
Figure 4.7: Variation of the Delay, $D_m$, in unit of hop-count with the Probability of Deflections, $p$, for the Uniform Traffic Distribution (UTD) Pattern in a 36-Node MSN.
Figure 4.8: Variation of the Delay, $D_m$, in unit of hop-count with the Probability of Deflections, $p$, for the Uniform Traffic Distribution (UTD) Pattern in a 16-Node MSN.
Table 4.1: Showing the Variation of $(N_{df})_{max}$ and $D_m$ with the Tuning Parameter, $a$, and the Probability of Deflections, $p$, for Two Types of Manhattan Street Networks (MSNs) for the Uniform Traffic Distribution (UTD) Pattern.

<table>
<thead>
<tr>
<th>Tuning Parameter</th>
<th>Prob. of Deflection</th>
<th>Maxim. Value of the Parameters, for a 16-Node i.e., 4x4 Network, for Case-A and Case-B, respectively.</th>
<th>Maxim. Value of the Parameters, for a 36-Node i.e., 6x6 Network, for Case-A and Case-B, respectively.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>$p$</td>
<td>$(N_{df})_{a}$</td>
<td>$D_m-a$</td>
</tr>
<tr>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>03</td>
</tr>
<tr>
<td>1</td>
<td>0.50</td>
<td>0.22</td>
<td>04</td>
</tr>
<tr>
<td>2</td>
<td>0.67</td>
<td>0.64</td>
<td>06</td>
</tr>
<tr>
<td>3</td>
<td>0.75</td>
<td>1.13</td>
<td>08</td>
</tr>
<tr>
<td>4</td>
<td>0.80</td>
<td>1.63</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>0.83</td>
<td>2.12</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>0.86</td>
<td>2.67</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>0.87</td>
<td>3.05</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>0.89</td>
<td>3.68</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>0.90</td>
<td>4.26</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>0.91</td>
<td>4.74</td>
<td>22</td>
</tr>
</tbody>
</table>
In Figures 4.7 and 4.8, while plotting the $D_m$ versus $p$ curves, we have assumed that the $a=0$ and $a=10$ points on the horizontal axis correspond to the $p=0$ and $p=1.0$ points respectively. It appears that the $D_m$ remains almost constant up to moderately large values of $a$ and $p$, i.e., up to small values of buffer size, $b$. Also, $D_m$ is always smaller for the Case-A, for reasons mentioned earlier in this section.

Next, we consider the zonal traffic distribution pattern for $\zeta=0.75$, and conduct similar performance evaluations for both type-1 and type-2 MSNs. The corresponding results are presented in Figures 4.9a through 4.10b. Equations (4.3) and (4.4) with $T(A)_1$ as given by equation (4.6), and equations (4.7) and (4.8) with $T(A)_2$ as given by equation (4.10) have been utilized for this purpose. Note that the behavior of these curves is very similar to those of Figures 4.5a through 4.6b.

Finally, the variation of the delay, $D_m$ with the tuning parameter $a$ is plotted in Figure 4.11 for different values of $\zeta$ in a 36-node (i.e., type-1) MSN. For larger values of $\zeta$, the variation of $D_m$ follows almost the same behavior as depicted in Figures 4.7 and 4.8. However, for $\zeta=0$, the $D_m$ remains almost constant throughout the entire spectrum of $a$ and $p$, as shown in Table 4.2 and Figure 4.11. This is due to the fact that the network is lightly (i.e., $< 50\%$) loaded in this case; only a small fraction of the source-destination pairs are communicating, and the rest are idle for this type of traffic distribution in the network.

4.4.2.2 Variation of $p$ with $a$, $b$, and $\lambda$

A deflection occurs when a packet, after arriving at a non-destination node, finds that the buffer of its chosen outgoing link is full and hence cannot be transmitted through the desired outgoing link. The probability of deflection, $p$, at any node is therefore, the
probability that a choice exists between the two outgoing links and the buffer of that chosen link is full. We are assuming that these two events are independent. It is required to determine the probability that both the outgoing links are not chosen links, and the probability that the buffer of the preferred link is full. The former probability distribution can be determined for \( a=0 \) and \( a\neq 0 \), and the latter one can be determined from the analysis, as presented in Appendix-B, for an M/D/1 queue with finite buffer space.

![Graph](image)

**Figure 4.9a:** Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, \( p \), in a Type-1 MSN (a 36-Node Network) for Case-A for the Zonal Traffic Distribution (ZTD) Pattern with \( \zeta=0.75 \).
Figure 4.9b: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, $p$, in a Type-1 MSN (a 36-Node Network) for Case-B for the Zonal Traffic Distribution (ZTD) Pattern with $\zeta=0.75$. 
Figure 4.10a: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, $p$, in a Type-2 MSN (a 16-Node Network) for Case-A for the Zonal Traffic Distribution (ZTD) Pattern with $\zeta=0.75$. 
Figure 4.10b: Variation of the MEan Number of Deflections (MENDs) with the Probability of Deflections, $p$, in a Type-2 MSN (a 16-Node Network) for Case-B for the Zonal Traffic Distribution (ZTD) Pattern with $\zeta=0.75$. 
Figure 4.11: Variation of Delay, $D_m$, in unit of hop-count, with the Probability of Deflection, $p$, for Different Values of $\zeta$ of Zonal Traffic Distribution (ZTD) Pattern in a 36-Node MSN.
Table 4.2: Showing the Variation of $(N_{df})_{max}$ and $D_m$ with the Tuning Parameter, $a$, and the Probability of Deflections, $p$, for the Zonal Traffic Distribution (ZTD) Pattern in a Type-1 Manhattan Street Networks (MSN).

<table>
<thead>
<tr>
<th>Tuning Parameter</th>
<th>Prob. of Deflection</th>
<th>Maxm. Value of the Parameters, for a 36-Node MSN, for Case-A and Case-B, respectively, with $\zeta=0.0$.</th>
<th>Maxm. Value of the Parameters, for a 36-Node MSN, for Case-A and Case-B, respectively, with $\zeta=1.0$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>$p$</td>
<td>$(N_{df})_{a}$</td>
<td>$D_m^a$</td>
</tr>
<tr>
<td>0</td>
<td>0.00</td>
<td>0.000</td>
<td>06</td>
</tr>
<tr>
<td>1</td>
<td>0.50</td>
<td>0.004</td>
<td>06</td>
</tr>
<tr>
<td>2</td>
<td>0.67</td>
<td>0.020</td>
<td>06</td>
</tr>
<tr>
<td>3</td>
<td>0.75</td>
<td>0.045</td>
<td>06</td>
</tr>
<tr>
<td>4</td>
<td>0.80</td>
<td>0.074</td>
<td>07</td>
</tr>
<tr>
<td>5</td>
<td>0.83</td>
<td>0.102</td>
<td>07</td>
</tr>
<tr>
<td>6</td>
<td>0.86</td>
<td>0.138</td>
<td>07</td>
</tr>
<tr>
<td>7</td>
<td>0.87</td>
<td>0.153</td>
<td>07</td>
</tr>
<tr>
<td>8</td>
<td>0.89</td>
<td>0.200</td>
<td>07</td>
</tr>
<tr>
<td>9</td>
<td>0.90</td>
<td>0.243</td>
<td>07</td>
</tr>
<tr>
<td>10</td>
<td>0.91</td>
<td>0.270</td>
<td>07</td>
</tr>
</tbody>
</table>
(i) $\alpha = 0$. This implies that the network is ideal, i.e., the links are of infinite capacity or that the nodes are equipped with infinite buffer space. Therefore, a packet never suffers from deflection, since it is always routed through the desired outgoing link. In this case, only the waiting time at the buffers of the intermediate nodes are to be added to the MIMIND for computing the MPTT.

(ii) $\alpha \neq 0$. This implies that the network is not ideal, which means that the links are not of infinite capacity nor are the nodes equipped with infinite buffer space. This may also include the case when one or more nodes or links of the network is/are faulty. The higher the number of allowable deflections, the higher the value of $\alpha$, i.e., the higher the amount of time a packet is allowed to be in transit before it reaches the destination. Thus, the access rate of the packets from the dormant terminals may become lower. Consequently, the overall effective utilization of the network may decrease.

The cases for $\alpha = 1, 2, \text{ and } 3$ are studied using the geometric construction presented in Figure 4.3, for a 16-node MSN. The values of the probabilities that a choice exists between the two outgoing links for 1-hop, 2-hop, 3-hop, 4-hop, and 5-hop source-destination patterns are computed using the aforementioned geometric construction (Figure 4.3), and are reported in Table 4.3. This table also contains the mean values of the Prob.(Choice Exists).

The values of the Prob.(System Full) for an M/D/1 queue with finite buffer space, e.g., $b = 0, 2, 4, \text{ etc.}$, are given in Table 4.4 for different values of $\lambda$ and these variations are plotted in Figure 4.12. The relevant analytical developments are presented in Appendix-B. Finally, the variation of the probability of deflection with the variation of $\alpha, b, \text{ and } \lambda$ is presented in Table 4.5, and plotted in Figures 4.13a, 4.13b, and 4.13c, respectively.
Note that the Prob.(Packet Loss) in a network with finite nodal storage capacity is very similar to the Prob.(Packet deflection), in this case. This occurs due to the fact that the deflection routing votes in favor of deflecting the packets rather than dropping them immediately unless they have already travelled a pre-specified number of hops in the network.

![Graph showing the effect of buffer size on packet loss](image)

**Figure 4.12:** Effect of Buffer Size, $b$, on the Probability of Packet Loss (i.e., Deflection) as a Function of the Packet Arrival Rate, $\lambda$. 
Figure 4.13a: Effect of Buffer Size, $b$, on the Probability of Deflection, $p$, as a Function of the Packet Arrival Rate, $\lambda$, for $a=1.0$, in a 4x4 Manhattan Street Network (MSN).
Figure 4.13b: Effect of Buffer Size, $b$, on the Probability of Deflection, $p$, as a Function of the Packet Arrival Rate, $\lambda$, for $a=2.0$, in a 4x4 Manhattan Street Network (MSN).
Figure 4.13c: Effect of Buffer Size, $b$, on the Probability of Deflection, $p$, as a Function of the Packet Arrival Rate, $\lambda$, for $\alpha=3.0$, in a 4x4 Manhattan Street Network (MSN).
Table 4.3: Variation of the Probability (Choice Exists), Between Two Outgoing Links, at any Node *en route* to Destination for Case-A and Case-B in a 4x4 i.e., 16-Node Manhattan Street Network (MSN).

<table>
<thead>
<tr>
<th>Tuning Parameter</th>
<th>Source-Destination Separation</th>
<th>Total Number of Deflections Allowed, for two different cases</th>
<th>Prob. (Choice Exists) between two outgoing links at any node <em>en route</em> to destination</th>
<th>Mean* value of the Prob. (Choice Exists)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d</td>
<td>Case-A</td>
<td>Case-B</td>
<td>Case-A</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.#</td>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2</td>
<td>-</td>
<td>.22-.33</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>-</td>
<td>.20-.33</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>-</td>
<td>.12-.24</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0.375</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>-</td>
<td>0.140</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6</td>
<td>-</td>
<td>.13-.26</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>8</td>
<td>-</td>
<td>.07-.21</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>0.273</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6</td>
<td>-</td>
<td>0.100</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>9</td>
<td>-</td>
<td>.10-.24</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>12</td>
<td>-</td>
<td>.05-.20</td>
</tr>
</tbody>
</table>

* computed by taking the weighted average.  
# means does not hold here.
Table 4.4: Variation of the Probability the an Arrival Finds that the System is Full with the variation of the Arrival Rate, $\lambda$, and the Size of Waiting Buffer, $b$.

<table>
<thead>
<tr>
<th>Packet Arrival Rate per station per slot.</th>
<th>Probability(System Full) for an M/D/1 Queue with waiting buffer size, $b$, and hence system size ($b+1$) units of packets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>$b=0$</td>
</tr>
<tr>
<td>0.00</td>
<td>0.000</td>
</tr>
<tr>
<td>0.10</td>
<td>0.095</td>
</tr>
<tr>
<td>0.30</td>
<td>0.259</td>
</tr>
<tr>
<td>0.50</td>
<td>0.394</td>
</tr>
<tr>
<td>0.70</td>
<td>0.504</td>
</tr>
<tr>
<td>0.90</td>
<td>0.594</td>
</tr>
<tr>
<td>0.95</td>
<td>0.613</td>
</tr>
</tbody>
</table>
Table 4.5: Variation of the Probability of Deflection, $p$, with the Tuning Parameter, $a$, Size of the Waiting Buffer, $b$, and the Packet Arrival Rate, $\lambda$, for Case-A in a 4x4 i.e., 16-Node Manhattan Street Network (MSN).

<table>
<thead>
<tr>
<th>Packet arrival rate, $\lambda$</th>
<th>Probability of deflection, $p$, for Case-A for different values of the tuning parameter, $a$.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$a=1$</td>
</tr>
<tr>
<td></td>
<td>$b=0$</td>
</tr>
<tr>
<td>0.10</td>
<td>4.2e-2</td>
</tr>
<tr>
<td>0.30</td>
<td>1.1e-1</td>
</tr>
<tr>
<td>0.50</td>
<td>1.7e-1</td>
</tr>
<tr>
<td>0.70</td>
<td>2.2e-1</td>
</tr>
<tr>
<td>0.90</td>
<td>2.6e-1</td>
</tr>
<tr>
<td>0.95</td>
<td>2.7e-1</td>
</tr>
</tbody>
</table>

^ in unit of port slot per station.
4.4.2.3 Variation of $D_{avg}$ with $a$, $b$, and $\lambda$

As stated in section 4.2, the generalized expression for the MPTT consists of three components: the first one can be attributed to the topological structure of the network, and the other two depend on both the routing policy and the nodal storage capacity. The relationship is as shown below.

$$D_{avg}(a, b, \lambda) = D_m(a=0) + \{(\varphi_{avg} (N_{df}avg (a, b, \lambda)) (1.0 + T_{avg})\}$$

$$+ \{(D_m(a=0) - 1.0) T_{avg}\}$$

(4.13)

$$D_{avg}(a, b, \lambda) = D_m(a=0) + \{(\varphi_{avg} (N_{df}avg (a, b, \lambda)) (1.0 + T_{avg})\}$$

$$+ \{D_m(a=0) T_{avg}\}$$

(4.14)

Equations (4.13) and (4.14) are applicable to case-A and case-B respectively, as described in sections 4.2 and 4.3. In the above equations:

The **first term** represents the mean inter-node distance which is the average number of hops separating the source and destination, as found in Chapter 3 for different types of MSNs.

The **second term** represents hop-equivalent of the delay suffered due to deflections, i.e., this figure represents the deflection penalty. For simplicity it has been assumed that one hop transmission time is equivalent to one slot. The deflection penalty consists of the number of additional hops to be travelled due to deflections, plus the total waiting time to cross those additional hops. As mentioned previously, this occurs due to the unavailability of adequate buffer space at the nodes. Analytical developments regarding this component of the delay is presented in section 4.3.
The third term of equation (4.13) represents the pure waiting time in queues of an ideal network. It consists of total waiting time *en route* to destination from the source. For the case of limited buffer space in an M/D/1 queue, the relevant data can be obtained from Table 4.6 or Figure 4.14.

Now, given the network size, \( w \) (i.e., the width of the network), \( D_m(a=0) \) can be computed using the analytical expressions developed in Chapter 3. It is required to use equation (3.7) for type-1 MSNs, and equation (3.10) for type-2 MSNs.

The value of the \( \varphi_{\text{max}} \) is 4 for the MSNs and, therefore, we assume that the \( \varphi_{\text{avg}} \) is 2 hops. \( (N_{df})_{\text{avg}}(a, b, \lambda) \) is then computed using the following procedure. Given a value of \( p \) for a particular set of \( a, b \) and \( \lambda \), find \( (N_{df})_{\text{avg}} \) using equation (4.3) for case-A, and equation (4.4) for case-B in type-1 MSNs. For type-2 MSNs, it is required to use equation (4.7) for case-A, and equation (4.8) for case-B. This \( (N_{df})_{\text{avg}} \) is then used to compute the deflection penalty which is a component of the second term in equations (4.13) and (4.14).

The variations of the second and third terms of equation (4.13) with the nodal storage capacity, \( b \) for each of the outgoing link, and the packet arrival rate \( \lambda \), are reported in Figures 4.15a, 4.15b, and 4.15c. The relevant data are presented in Table 4.7, Table 4.8, and Table 4.9, for \( a=1, a=2 \) and \( a=3 \), respectively.

The final results, i.e., the variation of different components, viz., waiting and deflection, of penalties with \( b \) and \( \lambda \) are plotted in Figure 4.16 for \( a=1 \). Proceeding in a similar manner, we can obtain these families of curves for case-B, using equation (4.14). Figure 4.16 shows that for a given nodal storage capacity and a particular packet arrival rate per station, there exists an optimum (in the sense of minimum MPTT) network operating
point for every possible value of the tuning parameter. In general, if the users are allowed to control the tuning of the network, a malicious user might jam the network by assigning very high value for the tuning parameter [77]. The same situation also occurs during heavy load conditions and when faults exist in the network.

A very efficient technique to combat such jamming or congestion is to dynamically tune the soft-topology embedded onto the hard-topology of the network. The basis of such embedding can be short-term or long-term predictive load estimation, or on-line (i.e., real-time) load monitoring as shown in the next Chapter.

![Figure 4.14: Variation of the Waiting Time for an M/D/I/(b+I) Queue with Buffer Size, b.](image-url)
Figure 4.15a-i  Variation of the Total Waiting and Deflection Penalties with Buffer Size, \( b \), and the Packet Arrival Rate, \( \lambda \), with the Network Tuning Parameter; \( a=1.0, \ b=2 \), in a 4x4 Manhattan Street Network (MSN).
Figure 4.15a-ii  Variation of the Total Waiting and Deflection Penalties with Buffer Size, \( b \), and the Packet Arrival Rate, \( \lambda \), with the Network Tuning Parameter; \( a=1.0, b=4 \), in a 4x4 Manhattan Street Network (MSN).
Figure 4.15b-i  Variation of the Total Waiting and Deflection Penalties with Buffer Size, $b$, and the Packet Arrival Rate, $\lambda$, with the Network Tuning Parameter; $a=2.0, b=2$, in a 4x4 Manhattan Street Network (MSN).
Figure 4.15b-ii Variation of the Total Waiting and Deflection Penalties with Buffer Size, $b$, and the Packet Arrival Rate, $\lambda$, with the Network Tuning Parameter; $a=2.0$, $b=4$, in a 4x4 Manhattan Street Network (MSN).
Figure 4.15c-i  Variation of the Total Waiting and Deflection Penalties with Buffer Size, $b$, and the Packet Arrival Rate, $\lambda$, with the Network Tuning Parameter; $a=3.0, b=2$, in a 4x4 Manhattan Street Network (MSN).
Figure 4.15c-ii Variation of the Total Waiting and Deflection Penalties with Buffer Size, $b$, and the Packet Arrival Rate, $\lambda$, with the Network Tuning Parameter; $a=3.0, b=4$, in a 4x4 Manhattan Street Network (MSN).
Figure 4.16: Variation of Different Components of Delay with Buffer Size, $b$, and the Packet Arrival Rate, $\lambda$, for Case-A in a 4x4 Manhattan Street Network (MSN) for the Unity Value of the Network Tuning Parameter, i.e., $a=1.0$. 
Table 4.6: Variation of the Average Waiting Time in Number of Slots for an M/D/1/(b+1) Queueing System with the Size of the Waiting Buffer, \( b \), and the Packet Arrival Rate, \( \lambda \).

<table>
<thead>
<tr>
<th>Packet arrival rate per slot per station</th>
<th>Waiting time in number of slots (or packets)</th>
<th>Buffer size = 2 packets</th>
<th>Buffer size = 4 packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.10</td>
<td></td>
<td>0.105</td>
<td>0.106</td>
</tr>
<tr>
<td>0.30</td>
<td></td>
<td>0.340</td>
<td>0.364</td>
</tr>
<tr>
<td>0.50</td>
<td></td>
<td>0.580</td>
<td>0.734</td>
</tr>
<tr>
<td>0.70</td>
<td></td>
<td>0.764</td>
<td>1.260</td>
</tr>
<tr>
<td>0.90</td>
<td></td>
<td>0.845</td>
<td>1.780</td>
</tr>
<tr>
<td>0.95</td>
<td></td>
<td>0.850</td>
<td>2.150</td>
</tr>
</tbody>
</table>
Table 4.7: Variation of the Penalties in Number of Slots Due to Waiting and Deflections in a 4x4 i.e., 16-Node Manhattan Street Network (MSN) for Case-A with Packet Arrival Rate Per Slot Per Station, $\lambda$, and the Size of the Waiting Buffer, $b$ Per Outgoing Link for $a=1.0$. [Note: $\varphi_{avg} = 2$ Hops for the MSNs, and the Minimum Mean Inter-Node Distance for a 16-Node MSN is 2.93=3.0 Hops].

<table>
<thead>
<tr>
<th>Packet arrival rate, $\lambda$, in unit of packets per slot per station.</th>
<th>Buffer size = 2 packets</th>
<th>Buffer size = 4 packets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pure waiting time, slots.</td>
<td>Deflection and waiting time due to deflection, slots</td>
</tr>
<tr>
<td>0.10</td>
<td>0.210</td>
<td>2.6e-4</td>
</tr>
<tr>
<td>0.30</td>
<td>0.680</td>
<td>0.011</td>
</tr>
<tr>
<td>0.50</td>
<td>1.160</td>
<td>0.073</td>
</tr>
<tr>
<td>0.70</td>
<td>1.528</td>
<td>0.250</td>
</tr>
<tr>
<td>0.90</td>
<td>1.690</td>
<td>0.550</td>
</tr>
<tr>
<td>0.95</td>
<td>1.700</td>
<td>0.633</td>
</tr>
<tr>
<td></td>
<td>0.212</td>
<td>8.72e-6</td>
</tr>
<tr>
<td></td>
<td>0.730</td>
<td>1.75e-4</td>
</tr>
<tr>
<td></td>
<td>1.470</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td>2.520</td>
<td>0.080</td>
</tr>
<tr>
<td></td>
<td>3.560</td>
<td>0.280</td>
</tr>
<tr>
<td></td>
<td>4.300</td>
<td>0.270</td>
</tr>
</tbody>
</table>
Table 4.8: Variation of the Penalties in Number of Slots Due to Waiting and Deflections in a 4x4 i.e., 16-Node Manhattan Street Network (MSN) for Case-A with Packet Arrival Rate Per Slot Per Station, \( \lambda \), and the Size of the Waiting Buffer, \( b \) Per Outgoing Link for \( a=2.0 \). [Note: \( \varphi_{avg} = 2 \) Hops for the MSNs, and the Minimum Mean Inter-Node Distance for a 16-Node MSN is 2.93=3.0 Hops].

<table>
<thead>
<tr>
<th>Packet arrival rate, ( \lambda ), in unit of packets per slot per station.</th>
<th>Buffer size = 2 packets</th>
<th>Buffer size = 4 packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure waiting time, slots.</td>
<td>Deflection and waiting time due to deflection, slots</td>
<td>Pure waiting time, slots.</td>
</tr>
<tr>
<td>0.10</td>
<td>0.210</td>
<td>5.04e-8</td>
</tr>
<tr>
<td>0.30</td>
<td>0.680</td>
<td>7.55e-5</td>
</tr>
<tr>
<td>0.50</td>
<td>1.160</td>
<td>2.70e-3</td>
</tr>
<tr>
<td>0.70</td>
<td>1.528</td>
<td>2.54e-2</td>
</tr>
<tr>
<td>0.90</td>
<td>1.690</td>
<td>1.10e-1</td>
</tr>
<tr>
<td>0.95</td>
<td>1.700</td>
<td>1.42e-1</td>
</tr>
</tbody>
</table>
Table 4.9: Variation of the Penalties in Number of Slots Due to Waiting and Deflections in a 4×4 i.e., 16-Node Manhattan Street Network (MSN) for Case-A with Packet Arrival Rate Per Slot Per Station, $\lambda$, and the Size of the Waiting Buffer, $b$ Per Outgoing Link for $\alpha=3.0$. [Note: $\varphi_{avg} = 2$ Hops for the MSNs, and the Minimum Mean Inter-Node Distance for a 16-Node MSN is 2.93=3.0 Hops].

<table>
<thead>
<tr>
<th>Packet arrival rate, $\lambda$, in unit of packets per slot per station.</th>
<th>Buffer size = 2 packets</th>
<th>Buffer size = 4 packets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pure waiting time, slots.</td>
<td>Deflection and waiting time due to deflection, slots</td>
</tr>
<tr>
<td>0.10</td>
<td>0.210</td>
<td>4.73e-12</td>
</tr>
<tr>
<td>0.30</td>
<td>0.680</td>
<td>2.46e-07</td>
</tr>
<tr>
<td>0.50</td>
<td>1.160</td>
<td>5.00e-05</td>
</tr>
<tr>
<td>0.70</td>
<td>1.528</td>
<td>1.36e-03</td>
</tr>
<tr>
<td>0.90</td>
<td>1.690</td>
<td>1.17e-02</td>
</tr>
<tr>
<td>0.95</td>
<td>1.700</td>
<td>1.76e-02</td>
</tr>
</tbody>
</table>
4.5 Simulation Scenarios and Results

The structure, operation, and dynamics of two MSNs are simulated†; one with 16
nodes, i.e., a 4x4 MSN, and the other with 36 nodes, i.e., a 6x6 MSN. The representative
results for the 16-node MSN using the UTD are presented here. The dynamics of these
networks have been observed, and some (i.e., during the steady state of operation) are
recorded to gather statistical results on MPTT, Upper bound of the MPTT, Variation of the
number of deflections suffered by a packet en route to destination with tuning parameter,
probability of deflection, and the offered load, and Mean and upper bound of the overall
network throughput.

In the simulation, the following Network Parameters (i.e., design variables)
were considered:

- Each node is a 3x3 switch with output buffering capability; buffer sizes studied are 0, 2,
and 4. Appendix-C shows all possible switching patterns of a 3x3 switch.

- Transmission capacity of each channel (i.e., wavelength) is assumed to be 10 Gbps,
[23, 40, 112]. In Appendix-A, it has been shown that a 4x4 MSN can
be embedded onto a 6-channel ring network. This can be easily achieved
by Wavelength Division Multiplexing (WDM) technique.

- Maximum packet size of 2400 bytes is used. Note that, one screen of a data terminal
or PC contains 2000 bytes (25 lines, each with 80 characters, and 1 character =
1byte=8 bits). And, one voice burst (200-300 msec) contains 2400 bytes, when 64
kbps PCM coding of voice signal is used. Therefore, maximum packet
transmission time = [(2400x8)/(10x10^9)]=1.92 μsec.

† this simulation necessitates a knowledge of the C programming language.
• Nodal processing speed (using AlGaAs/GaAs heterojunction bipolar transistor technology) = 1 Gbps. And, the total time consumed for:

  • processing (next desirable node's address computation using the formulations given in [104] plus packets priority category determination plus conflict, if any, resolution) packets header,

  • slot boundary adjustment [103-104],

  • switching pattern implementation, and

  • transferring of the packets to the input buffers of the outgoing links is assumed to be $\geq 2.0 \mu$sec.,

• Hop length = 1000 meter, and assuming that the velocity of light in the optical fiber is $2.5 \times 10^8$ meter/sec, the packet propagation time = 4.0 $\mu$sec.

Thus, the components of delay are: Nodal processing time, Hop transmission time, and Hop propagation time, with the hop propagation time as the most dominating component. As mentioned before, we have assumed that the time is slotted and actions are synchronized in time slot whose duration is sufficient to transfer a packet of fixed size.

We find that one Hop-Count (defined as the time taken by a packet for one hop transmission, propagation, and the processing overhead at one end of the hop) is = 7.920 $\mu$sec. We measure the MPTT in units of this hop-count (equivalent to one slot of time). Note that the dominating component of the hop-count is the packet propagation time which is solely dependent on the length of the hop. Waiting delay at nodal queue is measured in unit of slots rather than that of packet transmission time. This is because we intend to lump the opto-electronic and electro-optic conversion delays within that component.
The simulation program also mimicked the behavior of each of the aforementioned networks in the following operational modes.

- Case-A and case-B of network access schemes (defined in sections 4.2 and 4.3),
- Uniform and zonal traffic distribution patterns,
- A wide range of network tuning (the parameter $\alpha$ is varied form 0 to 10),
- Both one packet in the network without any contention en route, and multi-packet with pre-specified number of contentions along the shortest path travel to the destination have been studied.
- In case of contentions, the priority category of a packet is determined on the basis of number of deflections suffered and the ties are resolved by tossing a two-sided (since there can be at most two packets contending for one output link at any time slot) fair coin.
- A wide range of offered load has also been considered in the simulation ($\lambda$ is varied from 0.0 to 0.95. Note that the unit of $\lambda$ is packets per slot per station).

Few remarks regarding the procedure adopted for taking measurements are in order. Ten measurements -- each for at least 100,000 slots -- were taken to compute the average values (and their 95% confidence intervals) of each of the parameters of interest ($D_m$, $D_{avg}$, $N_{df}$, $p$, etc.) for both 16-node and 36-node MSNs. In an attempt to reduce the effect of initial transients, no observation was recorded until •At least 4000 packets have successfully travelled through the network, and •Each node has received sufficiently large (e.g., 10) number of packets. Measurements were taken intermittently, e.g., once in every 10 packets receipts, during the steady state operation (i.e., equilibrium condition) of the networks. No explicit flow control mechanism except those policed by the access control
strategy (case-A or case-B) and threshold-based packet dropping are implemented in the simulation.

We close this section with few comments regarding the close agreement between the simulation and analytical results. A comparison of the values of $D_m$ obtained by simulation with those obtained analytically is presented in Figure 4.8. In each of the cases, we find that the values of $D_m$ obtained by analysis are within $\pm 12\%$ of the values obtained by simulation, and hence our analytical technique can accurately (with tolerable errors) compute the useful performance parameters.

4.6 Conclusions

We have presented a new analytical technique for determining the mean and the upper bound of the packet transfer time in a class of regular mesh networks called the Manhattan Street Networks (MSNs).

The analytical results match closely those of simulations as found in [79] and shown in Figure 4.8. One of the most attractive features of this technique is its simplicity. The applicability of the proposed technique has been demonstrated for a simple routing rule proposed in [104] for two different traffic distribution patterns, viz., zonal and uniformly distributed traffics, in two types of MSNs.

We have also considered two different network access control schemes for packets. As expected, the delay in the first case is always lower compared to that in the second case.
The proposed technique can be very effectively used for rapidly evaluating routing strategies in large supra-high-speed packet-switched metropolitan and wide area networks.
Chapter 5

CONGESTION AVOIDANCE USING NEURAL ARBITERS

In this chapter, we present a survey of the techniques used for controlling congestion in large high-speed packet-switched networks. We note that with the availability of economically feasible optical fibers as transmission media, the network bottleneck elements have moved from the transmission domain to the processing domain. Consequently, most of the previously, i.e., in the 1970s and early 1980s, proposed network access control and resource management techniques may not be effectively applied to the emerging large lightwave networks. In view of these facts, we propose Artificial Neural Network (ANN) based arbitration for fast and intelligent processing of transit packets, and assigning network access to the local packets. We employ Grossberg's [52] Adaptive Resonance Theory (ART) based neural network, a two-layer ANN, to maintain distributed cut-through links between the communicating users. And, a buffer stealer -- another ANN based controller, as discussed in Appendix-D -- is used to throttle local source's access to the network during heavy load conditions. With the incorporation of these additional complexities at the nodal structure, it is possible to utilize most efficiently the available network resources under all possible, i.e., both certain and uncertain, operating conditions. As before, we use the Manhattan Street Network (MSN) architecture as the example of soft-topology for our investigations.

108
5.0 Introduction

Congestion is the state of network operations where either because of mismanagement or excessive requests or faults in the network, demands for resources exceed the available capacity. In large high-speed packet-switched networks, the basic network resources [68] are:

- Transmission capacity of links, i.e., communication bandwidth,
- Nodal storage capacity,
- Nodal processing capacity, i.e., the speed and the capabilities of the protocols(s) of nodal operating system, and
- Number of nodal pre- and post-processors.

All of the above resources are available in limited quantity in the network, except for the communication bandwidth offered by the commercially available optical fiber transmission media. It is often said that in lightwave networks we have bandwidth to burn.

Now, to satisfy the communication demands of the Terminal Equipments (TEs), i.e., the rate at which the TEs wish to communicate, it is required to guarantee a certain level of un-interrupted flow of traffic (i.e., information) from one TE to the other(s). Although the network designers plan the traffic transportation capability, nodal processing capacity etc. to cope with the expected demands, many undesirable phenomena, e.g., a natural disaster which might cause link or node failure, zonal communication surge due to a sudden event of regional importance, temporary black-out of nodes or links due to component failure etc., might cause the difference between the traffic acceptance rate and the traffic delivery rate to increase beyond a pre-specified value. At this state of operation of the network, not only does the Grade-Of-Service (GOS) deteriorate, but also the availability of the network diminishes. This state is commonly known as the congested state of the network. Under such circumstances, the network contains more per-TE-pair transit packets
than that allowed by the design. As a result, the network *throughput* (i.e., the rate at which the network is delivering packets to the destination) *decreases* and the *Mean Packet Transfer Time (MPTT)* *increases significantly*, unless the influx of traffic is throttled via some *artificial mechanisms*.

Thus, we observe that, the main reasons [25, 68] of congestion in a network which does not contain any faulty link or node are:

- Improper design of network access control protocols,
- Inefficient design of the nodal operating system which includes the routing algorithm and the link control protocols,
- Existence of one or more bottleneck elements (nodes or links or a combination of them) which cause speed mis-match in the network, etc.

Congestion can be *local*, i.e., limited to one particular zone of the network or more seriously it can be *global*, i.e., throughout the entire network. An extremely congested network is usually referred to as being in the *deadlock* state, where the network *stops* receiving packets from, and delivering them to the TEs. That is, the network is *full of transit packets* which are *not flowing through!* Alternatively, it can be in the *livelock* state, where packets are *circulating forever* (!) *in the network*, without reaching the destination [28, 139]. Furthermore, the duration of both of the above types of congestion can be either *short-term* or *long-term* [68].

### 5.1 Different States of Network Operation

In the traditional store-and-forward type packet-switched networks, when a node is congested either because of inadequate nodal buffer space or processing capability, it
simply discards the packets. The same phenomenon happens when a link receives a request to transmit traffic at a rate greater than its capacity. The protocols in the upper layers of the ISO seven-layer structure must arrange for retransmitting the discarded packets. The scenario is a little bit different in the supra-high-speed packet-switched networks, e.g., the lightwave networks, where instead of discarding the packets they could be deflected [79, 104, 105]. Both retransmission and deflection cause the *en route* or *in transit* traffic to increase, the result of which is a reduction in effective utilization of the network resources.

A close observation of the standard [28, 58, 68, 130, 139] throughput versus offered load, and the mean delay versus offered load curves enables one to distinguish two possible types and three possible regions of network operation. The two types include linear (or quasi-linear) and parabolic type of variations. And, the three regions of operation are: Acceptable, Very Risky, and Disastrous or Congested regions, as shown in Figures 5.1a and 5.1b. These curves also show the ideal behavior of the throughput versus the offered load, and mean delay versus offered load curves. As can be easily seen, the throughput should ideally increase linearly with the offered load up to the unity (normalized) point where the corresponding load is 100%, and after that it should remain constant at unity irrespective of any change in the offered load. On the other hand, the ideal mean delay variation should be linear (with zero slope) up to the 100% load point after which it should be infinitely large.

Figure 5.1a presents one of the most undesirable effects of uncontrolled operation of a network. We observe that the mean delay, i.e., the mean packet transfer time remains almost constant in the acceptable region of the network operation. From the point of onset of congestion, the mean delay follows a parabolic variation with the increase of offered load. After that, if the offered load is further increased, the network enters the disastrous or congested region of operation where the mean delay becomes infinitely high. Therefore, if
the mean delay is to be maintained within certain upper bound, it is recommended that the offered load must be kept below the point of the \textit{onset of congestion}.

Figure 5.1b depicts typical relationships between throughput and offered load for uncontrolled and controlled networks. For an uncontrolled network, the throughput increases almost uniformly with an increase in the offered load up to the \textit{onset of congestion}, i.e., the peak point. After that, any additional load causes a decrease in the throughput. This trend continues until the network comes to a complete standstill, i.e., it becomes \textit{deadlocked}. This is the most undesirable state of network operation, because the network resources are being kept captured, the packets, however, are not getting delivered to their destination TEs, resulting in \textit{zero effective utilization}. An equivalent situation called \textit{livelock} (discussed in the previous section) may occur in lightwave networks when packets suffer a large number of deflections, and yet are neither being delivered to the destination nor being discarded from the network.

Therefore, it is desirable to allow the throughput to increase linearly or quasi-linearly with an increase in the offered load up to a \textit{certain maximum value}, as can be afforded by the network without severely degrading the performance. And, after that, throughput should be maintained at a constant value, irrespective of the offered load, as shown by the \textit{controlled} curve of Figure 5.1b. This can be achieved, in practice, by incorporating artificial traffic control mechanisms (e.g., packet dropping, admission control, etc.) as discussed in the next section.
Figure 5.1a: Delay Versus Offered Load Characteristics of Large High-Speed Packet-Switched Networks.
Figure 5.1b: Throughput Versus Offered Load Characteristics of Large High-Speed Packet-Switched Networks.
5.2 Types of Network Congestion

Depending on whether a congestion is local or global, short-term or long-term, as can be found by monitoring the network operation over a considerable period of time, different measures can be adopted for recovery. Another relevant point is that in which layer (data link layer, network layer, transport layer, etc.) of the standard OSI structure the congestion recovery mechanism should be incorporated. In general, short-term congestions can be overcome by periodically freezing some percentage of communication capacity or by capacity rationing, while frequent occurrence of long-term congestions call for addition of new resources in the network [28, 68]. In particular,

• **Local Short Term Congestion** can be recovered by dropping or deflecting packets, or by discouraging access from the local stations by assigning higher priority to the transit packets.

• **Local Long Term Congestion** can be recovered by increasing *nodal processing* and/or *link transmission* capacity, i.e., replacing the existing links with higher capacity links or adding more processors/buffers to the nodes.

• **Global Short Term Congestion** can be cured by implementing Local Short Term Congestion control schemes throughout the entire network. Another technique would be to reboot the operation of the network once an indication of the Global Short Term Congestion is noticed at the Operation Administration, and Management (OAM) center.

• **Global Long Term Congestion** is the worst situation of network operation. Global Long Term Congestion is equivalent to global communication blackout. Although the following techniques can be employed to rescue the network from such conditions of operation, it may be better to redesign the entire network by taking into account the peak of the most recent demand and traffic distribution patterns. Other procedures may include the following.
* The simplest procedure would be to delete all the transit packets from the network, provided that the TEs keep copies of all the packets. However, this is not a very efficient method from the point of view of network utilization, since these packets have already used the network resources.

* The second method recommends the maintenance of standby capacity (both link capacity and nodal storage) in the network which is used only when the state of network congestion is diagnosed.

* The third method is to use any ad-hoc means by the network operator in the OAM center to recover that very segment or zone of the network which is severely congested.

Finally, since prevention is better than cure, intelligent network designers must always incorporate congestion avoidance mechanisms in the network operating system. Other motivating factors in favor of congestion avoidance are:

- It helps to maintain the network operating point below or near (but neither at nor exceeding) the -
  * Peak of the throughput versus offered load curve, and
  * Knee of the mean delay versus offered load curve,

- It takes a much longer time to set the network free from the congested state than it does for the congestion to form, and

- It reduces the network outage rate, and hence the loss of revenues, by maintaining the promised GOS or QOS to the customers.

5.2.1 Desirable Characteristics of the Congestion Control Mechanisms

Although the desirable characteristics of congestion control mechanisms depend on both the type, i.e., local or global, short-term or long-term, etc., and the location, i.e., the
layer (of the OSI structure) in which it is occurring, the following are a few generic and
most desirable features [28, 34, 68, 101] of the congestion control techniques:

• It should be simple enough to permit implementation without incorporating
  excessive sophistication in the traffic processing mechanisms,
• It must be robust, so that it can operate even when the network is faulty,
• It must ensure fairness, so that during heavy load condition, the effective rate of
  traffic throttling from all the active users is uniform, and finally,
• It must be testable via simple procedures and implementations.

It happens quite often that a scheme performs very well, but its implementation is
very complicated. There always exists a trade-off between the complexity of a scheme and
the grade to which it can perform to achieve a certain goal.

5.2.2 Mechanisms for Avoiding/Controlling Congestion

In this section, we present a very brief overview of various mechanisms for
avoiding and/or controlling congestion in large high-speed packet-switched networks.
Many of them may not be applicable in the emerging large supra-high-speed, i.e.,
lightwave networks. Therefore, it is required to develop novel congestion control
techniques. The mechanisms are:

• Permit Based Control (PBC) or Isarithmic Control,
• Feedback Based Control (FBC), which include:
  * Source Throttling (ST), and
  * Window Size Control (WSC),
• Drop and Throttle based Control (DTC),
• Traffic Smoothing (TS), using e.g., Kalman filtering technique,
• Topology Tuning (TT), using e.g., neural arbiter or any other method, etc.
In the permit based control method, the flow of packets in the network is controlled in such a way that the total number of packets in transit is always below a certain upper limit [28]. Each node is initially allocated a fixed number of permits. Before launching its packets to the network, a source station must acquire permits to do so. At the intermediate nodes, a travelling packet is not subject to isarithmic control, and hence it does not need to capture any permit. Finally, at the destination node, the packet releases the permit to the pool of permits at that node, unless it already holds its own quota of permits. Otherwise, the permit is forwarded to a neighboring node. In a sense, the permit based control technique is similar to the channel allocation scheme of circuit switching, where a channel is established first, then and only then traffic transportation can resume. It has been shown in [28] that this method effectively controls the transit delay of packets at high load by keeping the throughput within reasonable bound. The major drawbacks of the scheme are [139]:

- Dynamic maintenance and redistribution of permits,
- Additional communication overheads due to permit based network access. This may not pose any problem in the lightwave networks.
- Recovery of lost permits; the network might stop working if no permit is available.

In the feedback based control method, the sending process (i.e., the source) utilizes the feedback information which might be an explicit Acknowledgement message (Ack.) from the receiving process (i.e., the destination) to control the flow of packet from its buffer to the network. The feedback based control technique can be implemented in any one or more of the following ISO layers: Data Link Layer, Network Layer, and Transport Layer. Furthermore, if properly engineered, e.g., if the window size is set on the basis of the source-destination separation, the feedback based control method can be used for both local and global congestion controls. Two widely discussed [28, 68, 120, 122, 130, 139] schemes in this category are: Source Throttling, and Window Size Control methods.
In the source throttling method, the load offered to the nodal buffers or communication links is measured continuously and once it exceeds a pre-specified threshold value (e.g., 70% of the utilization) the source is signalled back to reduce the offered load. Otherwise, the packets from the already warned sources are discarded from the network. Although this scheme is very complicated for practical implementation, it may be very attractive for congestion control in large networks [28, 120]. Some researchers call this method as choke packet based congestion control technique [139].

The window based control technique (WSC) is functionally similar to the choke packet based method for flow control. Here, both sender and receiver maintain windows of specific size, and the sender can delete packets from its buffer only after it has received explicit or implicit Ack. from the receiver. When explicit feedback messages (i.e., Acks) are used, the network must support the flow of these message packets with the highest priority. Thus, explicit feedback messages incorporates additional maintenance and operational overheads to the network.

One possible solution to this problem is to incorporate a feedback information field which contains the Congestion Indication (CI) bit in the network's standard packet format as suggested in [68, 120, 122]. These investigators also pointed out a number of relevant factors regarding the utilization of the CI bits for setting the window size, which include filtering out of noise (both level filtering and signal filtering) from the CI bits, assigning higher weights to the most recent CI bits, etc.

A number of techniques for dynamically adjusting the window size are discussed in [68] which include both additive and multiplicative increase and decrease of the window size. In particular, Ramakrishnan and Jain [122] show that an additive increase and multiplicative decrease provides fair allocation of network resources to a group of distributed communicating processes. One important point in this regard is the fact that the frequency of control and that of the feedback must match each other, otherwise an oscillatory or
unstable operation of the network will result [33, 68], which will accompany a degradation of network performance.

Finally, we note that although the congestion avoidance technique proposed by Ramakrishnan and Jain [122] does not add much of communication overhead, it does add to processing burdens, e.g., level filtering, signal filtering, etc. And, since with the availability of economically feasible optical fibers [40, 112], the designers are endowed with an enormous amount of communication capacity-on-hand to trade it off with system flexibility, reliability, expandability, maintenance, etc., the method proposed in [122] may not be very suitable for congestion avoidance in the lightwave networks.

The drop-and-throttle based control scheme is executed locally at the nodal operating system. However, it can perform both local and global congestion control. In this scheme, the nodal buffer space is partitioned into two zones: one for the packets from the local source, and the other for the transit packets. Different limits on the number of transit and local packets at the nodal buffers can be set and the transit packets are always given higher priority over the local packets. Furthermore, a packet closer to the destination may be assigned higher priority over the one which is far away. Also, it may be often beneficial to drop packets once they have travelled a pre-specified number -- proportional to the source-destination separation [79] -- of hops with an expectation that the protocols at the higher levels of the OSI structure will arrange for retransmission of the dropped packets. Relevant references regarding this technique are [28, 74, 79, 105, 130, 139].

The traffic smoothing technique calls for estimating the offered load on the basis of on-line monitoring and utilization of historical data on the offered load. The excess demands on network resources can then be throttled even before they have partially utilized expensive communication resources. One procedure which has very strong theoretical foundations in the area of estimation in noisy and/or uncertain environment is the Kalman
Filtering (KF) technique. This is a linear weighted sum of an updating of the previous best estimator, and a projection of the innovation due to the most recent observation. Since the weights can be computed recursively, optimal Kalman filtering offers an optimal recursive estimation in stochastic environment. In large high-speed networks, this can be used to estimate the expected load levels of links or nodes at different periods of the day, and then these results can be utilized to adaptively compute the best path for both packet routing and trans-routing. Thus, this technique can be used for global congestion control in large networks on a real-time basis. Details of this technique along with its extensions and applications can be found in [114].

The topology tuning scheme is a suitable technique for global congestion control in large supra-high-speed packet switching networks as shown in this chapter. The goal or the objective here is to dynamically embed the instantaneous traffic matrix onto the hard-topology which is the physical interconnection structure among the nodes. The instantaneous traffic matrix which is the demanded interconnection structure or pattern is called the soft-topology, and the term topology tuning refers to the aforementioned process of embedding the soft-topology in the hard-topology. Since the process of embedding has to be performed on-line, it may be preferable to have an estimate of the periodic traffic matrix from historical records on load demands. At the beginning of each time period, the nodal switches must start rearranging the assigned link capacities with minimal disruption of the existing assignments to embed the estimated traffic matrix for that very time period. The period may be quarter-, half-, or one-hour time slot. The duration of the period can also be estimated on the basis of the frequency with which the entries of the traffic matrix change; the slower the rate of change, the larger should be the period and vice versa. Since all the nodal switching activities will be taking place at the beginning of each period, it can be expected that minimal switching will occur during the period. Thus, this technique not
only renders network operation congestion-free, but also reduces the MPTT because of the distributed cut-through [130, 139] effect, and network status based access assignment.

The applicability of the topology tuning scheme is highly dependent on the characteristics of the traffic and the patterns of their generation, i.e., the distributions. For example, if the traffic is neither bursty nor unpredictable in volume, i.e., the bps rate is fixed, among a set of zones during certain time of the day, off-line generation of multi-period tuning schedule can be practised for avoiding network congestion as suggested in [123]. Relevant techniques for off-line determination of embedding of planar and non-planar graphs can be found in [3, 15, 21, 31, 70, 89, 90, 96]. In the context of our problem, essentially, all these methods call for a mathematical programming formulation of the soft-topology design problem, i.e., given the hard-topology and the link capacities (the constraints) find the soft-topology and the traffic flow (the design variables) such that the MPTT (the objective function) is minimized. Some researchers have also considered quasi-on-line reconfiguration of network structures in the context of multiple micro-processor interconnections [13, 118, 133, 149, 150]. We now discuss some of them.

One of the earliest reconfigurable highly parallel network structure is Snyder's [133] CHiP computer architecture. CHiP is a collection of homogeneous processing elements, switch lattice, and a controller. The processing elements are not directly connected to each other but are connected at regular intervals to each other, as dictated by the controller. Switch lattice is a regular structure formed by programmable switches connected via data paths. Each switch contains local memory to store several connection configuration messages, e.g., location 1 of the local memory may contain the necessary information to embed a mesh structure, location 2 may have the stored message for embedding a tree topology etc. It is the task of the controller to load the switch memory as
demanded by the problem, and to broadcast the configuration related information to all other non-initiator switches. A configuration setting is established only when a direct static connection is enabled between data paths (like circuit switching) by the switches. The most attractive feature of Snyder's architecture is its flexibility, i.e., multiple interconnection patterns can be embedded by maintaining locality.

Pradhan [118] proposes two highly regular dynamically reconfigurable topologies which can be defined algebraically. The lowest and the highest degrees of a node in these structures are 2 and 6 respectively. Since these structures are algebraically definable, algorithmic routing techniques can be easily employed. The second topology, where the i-th node is connected to the j-th node when either i=2j mod n or i=(2j+1) mod n is claimed to be a generalization of the deBruijn graph. We conjecture that this type of topology will be dominating the design of the soft-architecture of the emerging lightwave communication networks. In Pradhan's scheme, dynamic reconfiguration in both hierarchical and horizontal directions is achieved by using a reconfiguration message which is broadcast from the initiator to all other nodes (=Switch+Processing Elements+Buffers) via neighbors. Therefore, the method of reconfiguration is very similar to that of Snyder's [133]. Like other regular networks, Pradhan's architectures are also extendable and partitionable. Extendibility helps the network to grow dynamically in size, and partitionability makes it easily reconfigurable and gracefully degradable during faults.

Biswa and Srinivas [13] present an approach to embed reconfigurable tree architecture (N different binary trees with N different nodes or processing elements) on augmented multi-stage (number of stage, k=Log₂N; e.g., for N=8, three stages - each of 4 processing elements - are needed) shuffle-exchange networks. Here, unlike Snyder's [133] CHiP computer, the interconnection pattern is controlled by a central configuration controller along with an augmented stage of processing elements. The major features of this
scheme are: simplified routing, and hardware structure of the processing elements, fast switching from one configuration to another, because of simultaneous reception of configuration control information (= for straight and × for exchange) at every stage, and direct and simultaneous establishment of conflict-free paths, rather than via stage by stage. Consequently, the set up and transfer delays are reduced to $O(1)$ from $O(\log_2 N)$. Thus, we see that the configuration control overhead includes one complete stage of processing elements and a central controller.

Our concern in this chapter is fast and on-line topology tuning, i.e., very high-speed generation of the target soft-topology which is the demanded traffic matrix using neural arbiters [47-52, 117, 128] with an objective to avoid network congestion.

5.3 How the Artificial Neural Networks (ANNs) Based Approaches Can be Employed in Packet-Switched Systems

One of the most attractive feature of the ANNs is their capability to adapt to novel situations via learning. They can also approximate (since they make statistical decisions) highly complicated input-output relationships by selecting significant inputs, i.e., they can extract a set of feature parameters from the input data. These features, coupled with the capability of group or collective computation (because of the highly parallel and selective nature of interconnections among the neurons) endows the ANNs with the extremely high computational efficiency needed to solve any real-life large-scale optimization problem within reasonable amount of time. That is to say, for a given set of inputs, the output state of the ANN converges towards the global minima of the computational energy and hence can be used for solving optimization problems provided that the appropriate connections
and the relevant input features can be found [128]. Consequently, we find [83] that ANN based approaches can be very effectively applied for solving the following problems in the context of supra-high-speed packet switched systems:

- **Network design**, i.e., design of the physical (or hard) topology, which is a constrained optimization problem, and considered to be an NP (Non-deterministic Polynomial) -hard problem for large networks.

- **Traffic estimation** in operating networks based on e.g., historical data. The results can be used not only for congestion avoidance, but also for planning future capacity expansion and/or rearrangements within the existing network.

- **Access control** in networks in order to satisfy the pre-specified level of service availability and fairness constraints.

- **Efficient control of network operation**, i.e., achieving efficient utilization of available network resources and simultaneously satisfying the advertised level of GOS taking into account the nature of traffic distributions. This leads to congestion-free operation of networks which is the main focus of this chapter.

### 5.4 Motivations for Utilizing Artificial Neural Networks (ANNs) for Congestion-Free Operation of Packet-Switched Systems

In order to fulfil varieties of customer demands, e.g., voice, data, and video transmission services, from the same network, the telecommunication networks have evolved from early constant bit-rate service provider to the current-day integrated services networks which utilize fast packet or circuit switching technique for the purpose of traffic
transportation [65, 130, 139]. The packet switching technique gives the network an ability to offer both constant and variable bit rate services to its users, and the current availability of economically feasible optical fibers [40, 112] fulfil the bandwidth demands of such environment. The real challenge in this arena is to design a flexible, and hence intelligent, traffic controller to efficiently utilize the available network resources maintaining a pre-specified service quality even when the network is faulty, and the precise characteristics of the traffic offered by the Terminal Equipments (TEs) are not known. By traffic characteristics, we mean one or more of the following features: burstiness, traffic transportation rate demanded, etc.

As we have seen in the previous sections, the detrimental effects of network congestion are simultaneous degradation of service quality and allocation of network resources. And, the main reasons for network congestion are: lack of proper regulation in network access, inefficient routing of packets, and one or more faults in the network.

What is needed here is a controller whose operating logic domains are very ill-defined at the beginning of nodal operations. The controller dynamically learns to tune its operating principles to a variety of situations and conditions as it grows from a naive/virgin to an experienced/adult in the network.

A theoretical framework closest to this kind of scenario has been pioneered by Lotfi A. Zadeh [151] in the context of pattern recognitions, and relevant applications in as early as 1965. Subsequently, Zadeh also advocated utilization of his fuzzy logic theory for management of uncertainty in large systems [152, 153]. Realization of such a dynamically tunable controller based on fuzzy logic theory necessitates the following four steps:
• **Acquisition** of the uncertain knowledge regarding the operating status and relevant decisions, i.e., gathering of bound oriented uncertain decisions. For example, if the range of variation of a certain parameter is below, say, 5 units, categorize this behavior as the low activity region, and so on. Simulation based system evaluation can be a very useful tool for acquiring such knowledge.

• **Representation** of the fuzzy knowledge acquired in the previous step. This can be done either by using quasi-histogram type graphs, as practised by Zadeh [151-153] and his followers, or by using continuous curves which are usually bounded between -1 and +1 (or 0 and +1) as utilized by the neural network researchers [128]. These curves are popularly known as the threshold functions which are the decision makers in the neural networks.

• **Efficient implementation** of the fuzzy knowledge based inference engine described in the previous two steps. Again, it is the ANNs [117, 128] which provide a systematic architectural implementation procedure. ANNs are a special kind of massively parallel, and sometimes distributed, statistical computation machine consisting of very simple processing elements.

Computation within the processing elements is very simple in the sense that it is a threshold (linear or non-linear) based operation, like the add-compare-select type operation of the final stage of a Viterbi decoder [137, 146].

Communication overhead is either negligible or very low. Furthermore, computation and communication proceed in a quasi-parallel fashion in the ANNs [117]. Note that, the parallelism gives the necessary speed, and the probabilistic nature of the computation gives it the capability of making fuzzy or flexible decisions in an environment like supra-high-speed packet-switched telecommunication networks, shop floor of flexible manufacturing systems etc.
Continuous refinement or on-line tuning of the knowledge acquired in the first step. Here again, a conscious researcher must take the refuge of the ANNs. Because they have the capability to learn from the dynamics (mathematically represented by a set of coupled linear or non-linear differential equations to formalize the state update procedure) of the environment [52, 128] by adjusting the synaptic weights or connection strengths.

Thus, we see that ANN based controllers are perfectly capable of arbitrating network access and resource management decisions even in the most uncertain environments [62, 106, 117]. What we need to find is a simple ANN structure, possibly consisting of two layers of neurons, and capable of fast learning and synthesizing decisions as quickly as demanded by the application. Our research shows that, it is the Adaptive Resonance Theory (ART) based ANNs of Grossberg [52] which possess all these desirable characteristics. Consequently, we propose to use ART-Nets for congestion avoidance in large supra-high-speed packet-switched networks.

Traditional digital and/or expert system based controllers are either not going to work efficiently or will stop working in such unpredictable environments simply because they are not tunable to the previously unconceived (by the designer) conditions/situations of the network. They may also call for an unmanageable amount of complexity in their design and operation. Expert system based approach is not adequate for either service provisioning or resource management in uncertain operating environments because of the following reasons.

- Expert systems are usually storage intensive; they need a knowledge base to performs serial or semi-parallel processing of the conditions.
- Since expert systems perform rule based checking of the conditions extracted from the environment, the boundaries of the problem in this domain are rigid, i.e., the constraints are hard. Expert systems needs to learn (if it can be
trained at all!) numerous rules to perform the same task as can be done by one simple thresholding operation of an ANN.

Expert systems draw logical inferences in a deterministic manner, and are incapable of synthesizing knowledge. Consequently, it is unable to encounter novel situations. The ANNs, on the other hand, quasi-stochastically or deterministically, arrive at statistical decisions, and are capable of learning from the environment. Therefore, they can arbitrate decisions even in the most uncertain situations.

5.5 Utilization of Neural Arbiters for Congestion Avoidance

In section 5.3, we have presented a very brief overview of the usefulness of ANNs for efficient operation of large supra-high-speed packet-switched networks. The main reasons for operating a rule based system using fuzzy logic variable to make decision under uncertain environment [152] and subsequent modifications, i.e., adaptively assigning the synaptic weights for different conditions, are:

(1) The conventional rigid boundary operations, e.g., the basic functions of binary logic based system, i.e., the AND and OR, are taken to be MAXIMUM and MINIMUM of the certainty factors of the domain respectively. It may be possible that some information is completely ignored or lost during the decision making process. Appendix-E presents maximum and minimum selection networks using neural arbiters.

(2) It might happen that the uncertainties in the rules, facts, and logical operations are not treated in a unique manner.
(3) In an uncertain environment, an adopted decision is correct with a possibility of \( \alpha \), incorrect with a possibility of \( \beta \), and neither correct nor incorrect, i.e., uncertain, with a possibility of \( \psi \) (note: \( 0 \leq \alpha, \beta, \) and \( \psi \leq 1.0 \), and \( \psi = [1 - (\alpha + \beta)] \)) as dictated by the amount of evidence extracted from the environment. If \( \alpha \) exceeds \( \beta \) by a pre-specified threshold, then and only then can the possibly correct decision be assumed to be actually correct and can be executed, otherwise the status quo is maintained.

Furthermore, since our objective is to implement fuzzy rule based decision maker using associative memory system as done in the ANNs, we need:

- a layer of Short Term Memory (STM) systems which is a dynamic database to derive fuzzy inference with pre-specified certainty factor (threshold value) using the conditions extracted from the environment, and
- a tunable Long Term Memory (LTM) system which is some sort of static or quasi-static database. This is also called connection weights or synaptic weights. It precisely adapts the fuzzy decisions to the instantaneous requirements of the uncertain environment.

In particular, we should avoid the Back Propagation (BP) method of learning, not only because it is slow, but also because it tends to be unstable [52] in the environment we are considering. As mentioned before, we find that it is the ART network of Grossberg [52] which matches perfectly with our requirements.

A survey of the current literature shows that only a very few researchers have considered the deployment of ANNs for managing uncertainty in telecommunication networking environments. Two such applications are very briefly discussed below along with their drawbacks.

Hiramatsu [62] considers the utilization of multi-layer ANNs for regulating network access, where the ANN learns by Back Propagation of error [128]. And, the ANN needs
to go through training and control cycles before it starts effective operation for regulating the incoming calls in the ATM (Asynchronous Transfer Mode) environment. During the control cycle, the ANN uses a Leaky Pattern Table (LPT) method to probabilistically discard the too old network access patterns from the training set. The major shortcomings of this method are: (i) unless the training set of access patterns are chosen properly, the inaccurate synaptic weights might result in inefficient call regulation, (ii) since it uses multi-layer ANNs, the training process here is not as simple as it is in the two-layer ANNs [117], and finally, (iii) the drawbacks caused by the inability of the back propagation method to learn stably in dynamic environments [52].

Marrakchi and Troudet [106] consider one of the most primitive ANN, the Hopfield network [128] for the purpose of controlling the states, i.e., the connection configurations of a cross-bar switch. Here again, the problem is to find a suitable training set of connection configurations. Mathematically speaking this problem is equivalent to finding the rank, \( \oplus \) of an \( N \times N \) matrix (for an \( N \times N \) switch), and to train the controller with a set of \( \oplus \) connection patterns. The problem of convergence of Hopfield network is well-known [128], and hence the authors in [106] propose to utilize some additional control circuitry like a Hamming network [117, 128] to accelerate the decision making process.

Without further lengthening the discussion on motivation for selecting the ART-Nets, we concentrate mainly on an ANN based arbitration of transit traffic processing for avoiding network congestion. We need to encode the fuzzy rules for packet forwarding at the intermediate nodes based on a number of features at the packets header. Note that the importance as judged by the threshold functions for each of these features changes as the packet proceeds from a source to a destination. The factors causing these changes are: the dynamics of the network, traffic processing and control policies, service quality requirements, etc.
Our concern in this section is to show how Grossberg's neural network structure called the ART-1 machines -- which is a two-layer ANN -- can be applied for the purpose of switching state arbitration in large supra-high-speed packet switching systems. A very short overview of of the ART-1 machines is presented in Appendix-F.

5.5.1 The Proposed Procedure

The procedure used for arbitrating the connection configuration of the nodal switch is a very simple one. The switching decision is based on the weighted average of multiple criteria or features that are extracted from the packets header. These indicate the status of the links, nodes, packets, etc. If there is no conflict among the demands (for outgoing links) of the incoming packets, the demanded switching pattern is implemented, otherwise the ART-1 based conflict resolution algorithm is executed.

The pre-processor of the arbiter evaluates the demand of each packet on the basis of a variety of threshold functions designed to minimize MPTT and maximize the utilization of the available network resources, then either drops them or allows them to participate in the competition on the basis of the score (weighted sum of multiple criteria) of each packet.

The decision making at every step is on the basis of fuzzy operations, as explained in the subsequent sections. The results presented in [79] and [104] have also been utilized in some phases of this investigation.

5.5.2 Performance Evaluation

In this section, we first provide descriptions of the simulation model of nodal structure and the arbitration policy. Relevant performance results are then presented.
5.5.2a Description of the Simulation Model

We consider a 4x4 Manhattan Street Network (MSN) [78] as the logical topology (Figure 5.2). MSN can be embedded onto a variety of physical or hard topologies as shown in Appendix-A. The structural attributes of the simulated network node without and with neural arbiter are as shown in Figure 5.3 and Figure 5.4, respectively. The buffer stealer (see Appendix-D for details) in Figure 5.4 is a simple priority arbiter which contains a type-1 sorter (see appendix-E for details). The reason for selecting MSN as the example network is that a number of useful performance measures for this network without using neural arbiter has appeared in an earlier publication [79]. Consequently, the effect of incorporating the neural arbiter in the nodal structure can be easily visualized.

Our simulation program needs two kinds of inputs, viz., (i) Network parameters which describe the physical characteristics of the network, and (ii) Operational parameters regarding the logical operation of the network. These parameters are as follows:

- Specific burstiness (defined as the ratio of the average or mean bit-rate to the peak or maximum bit-rate) value, B. For constant bit-rate type traffic the value of burstiness is unity. Other necessary parameters in this regard are: average bit-rate values of the existing connections, the packet arrival rate per time unit (e.g., a slot) per station, and the packet size.

- Link capacity, i.e., the transmission rate of the individual links. We assume that each unit of the logical channel from a node to another is of the same capacity.

- A set of threshold functions for controlling the operation of the ART-1 based arbiter. Each of these threshold functions is defined with very specific performance goal. These thresholds adapt their shape (i.e., the control pattern) because of their capability of on-line tuning or learning.

- Number and patterns of the switching connections. These are presented in Appendix-C in details, and used for training the ART-1 based arbiter.
Figure 5.2: A 4x4 Manhattan Street Network (MSN) (adapted from [103]).
Figure 5.3: Architecture of a Node of the Manhattan Street Network (MSN) [BUF=Buffer].
Figure 5.4: Nodal Structure of a Manhattan Street Network with Arbiters.
We have adopted the activity scanning technique of simulation, because it is better than the event scheduling approach [110] for modeling systems where many segments must work together to come up with a result (decision) as in synergistic computation. The program is written in C language. It mimics the operation of the network nodes and links in accepting message packets from all possible inputs (incoming links and the local source) of a node and then passing them from node to node on the basis of the decision received from the ART-1 based arbiter.

As pointed out earlier, the arbiter makes the switching decision on the basis of the status of outgoing links and the information extracted from the packets header. The delay field of the data structure representing a packet is incremented as the packet travels from node to node until it reaches the destination.

The components of delay are: nodal processing time, hop transmission time, and hop propagation time. Note that in the emerging lightwave communication networks, the hop propagation time is the most dominating component of the hop delay. Finally, we have assumed that time is slotted and the actions are synchronized in a time slot whose duration is sufficient to transmit a packet of fixed size.

Each run of the simulation program has been considered to be of duration of at least 100,000 slots. Ten such runs are used to compute the final values of the performance parameter of interest, viz., mean and variance of network access delay, mean and variance of the MPTT, effective network utilization, average number of deflections suffered by a packet which has reached the destination, etc. The results presented are those for the UTD pattern. One can easily extend the simulation model to incorporate the zonal traffic distribution patterns.
For example, one instance of our simulation has the following parametric definitions:


- Nodal processing speed (using AlGaAs/GaAs heterojunction bipolar transistor technology) = 1 Gbps (see e.g., [23, 25, 60]) therefore, the total time consumed by pre-processing, arbitration (assuming that the arbiter can recognize the switching patterns within 10 top-down/bottom-up primings), switching pattern implementation, and transferring of the packets to the input buffers of the outgoing links is = 3.60 micro-sec. If the offered load is low (i.e., ≤ 0.3), finer recognition is performed, else coarse categorisation is done by the ART-based arbiter. By finer recognition we mean the the vigilence parameter is ≥ 0.80.

- Size of the buffers at the input of each of the non-local outgoing links of a node is 2 units, i.e., it can hold at most two packets. The capacity of the transmit buffer of the local source is 8 units, and that of the receive buffer is 4 units.

- Hop length = 1000 meter, and assuming that the velocity of light in the optical fiber is 2.5x10^8 meter/sec, we find that the packet propagation time = 4.0 μsec.

- Therefore, we find that one Hop-Count (defined as the time taken by a packet for one hop transmission, propagation, and the processing overhead at one end of the hop) is = 9.520 μsec. We measure the MPTT in units of this hop-count (equivalent to one slot of time). Note that the dominating component of the hop-count is the packet propagation time which is solely dependent on the length of the hop.
We now discuss the procedure for encoding (or representation) of fuzzy rules. Fuzzy rules have if-then type representational features [152]. And, since the ANNs are in general non-linear systems, they represent the most suitable building blocks for realizing fuzzy rule based system. An additional capability of the ANNs is their ability to learn from the environment with or without a supervisor [128]. The if attribute checks for the category of membership of each of the components of the feature vector. The then is concerned with a threshold crossing (by the input vector) based decision making process. In a sense, this is equivalent to synergistic computation.

For example, if the input vector, \( X = [x_1, x_2, x_3, \ldots, x_K] \), the fuzzy set (characterized by the membership function which represents each label's meaning) \( S_{rk} \) is like "High", "Medium" or "Low", \( W_{ki} \) are the synaptic weights, and \( Y_r \) is an output vector, then the \( r \)-th fuzzy rule can be encoded as follows:

\[
\text{If } ((x_1 \in S_{r1}) \& (x_2 \in S_{r2}) \& \ldots \& (x_k \in S_{rk}) \& \ldots \& (x_K \in S_{rK}))
\]

\[
\text{then } Y_r = \Phi_r (W_{ki}, X), \quad \text{where } k=1, 2, \ldots, K, \text{ and } r=1, 2, \ldots, R.
\]

Note that the \( \Phi_r (.) \) is the threshold function for the rule designed to achieve a specific goal.

Next, we discuss the input/output signals to/from the arbiter. These signals play a vital role in the design of the arbiter and hence, efficient operation of the network. Input signals contain various status and condition information extracted from the environment, i.e., the header of the incoming packet as shown in Figure 5.5, status of the incoming and outgoing links, and that of the local station, etc. These information include:

- Source and destination addresses of the packets,
• Next desirable node's logical address. Since the logical topology under consideration (i.e., the MSN) can be defined algebraically, the next desirable nodes logical address can be computed easily from the addresses of the current node and the destination [104].

• Sequence number of the packet. The priority category of the packet is extracted from the sequence number using the threshold function of Figure 5.6.

• Residual hop count, i.e., the additional number of hops still to be traversed by the packets to reach the destination. Two contradictory priority category extraction threshold as shown in Figure 5.7a and Figure 5.7b are used in this investigation.

• Number of deflections suffered by the packets under consideration. The threshold function utilized to determine the priority category membership of the packets using the value of the number of deflections suffered is as shown in Figure 5.8. This function is derived from the threshold function of Figure 5.9 which has been found to be very useful for optimum allocation of resources in a network of facilities. For example, the rate at which a user is allowed to transmit packets on the basis of node or link utilization can be easily controlled by using a threshold function derived from Figure 5.9, taking only the middle and right side of the trapezoid.

• The type of packet, i.e., whether it is a local or a transit packet, etc. The transit packet is always given higher priority over the local packet.

Based on this information, the preprocessor represented by an ellipse in the arbiter block of Figure 5.4 must fuzzily decide which of the outgoing links of the switch is the most desired one for each packet in question. This information is then fed to the ART-1 based arbiter. Since we are considering a switch with three input links, the size of the input vector to the ART-based arbiter is 3. For the first input link to the ART-based arbiter, it is selected
-- using the 2/3 rule (see Appendix-F for details) -- from the vector, \([ I_1-O_1 (f_{i1}, a_{i1}), I_2-O_1 (f_{i2}, a_{i2}), L_1-O_1 (f_{i1}, a_{i1})] \). Similarly, for the second input link it is extracted form \([ I_1-O_2 (f_{i1}, a_{i1}), I_2-O_2 (f_{i2}, a_{i2}), L_1-O_2 (f_{i2}, a_{i2})] \), and for the third input link, it is selected from \([ I_1-L_0 (f_{i0}, a_{i0}), I_2-L_0 (f_{i2}, a_{i0}), L_1-L_0 (f_{i0}, a_{i0})] \). The routing preference vector for the packets in each of the incoming links of the switch is computed using a number of features extracted from the packets header and the rules depicted in Figures 5.6, 5.7a or 5.7b, and 5.8. These diagrams also show the weights, i.e., the \(W_{ki}\), given to the features for specific ranges.

<table>
<thead>
<tr>
<th>SYNC</th>
<th>S</th>
<th>D</th>
<th>N</th>
<th>C</th>
<th>D</th>
<th>Res</th>
<th>INFO</th>
<th>SEQ</th>
<th>ECC</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

(0) SYNCHRONIZATION FIELD
(1) SOURCE IDENTIFICATION VECTOR
(2) DESTINATION IDENTIFICATION VECTOR
(3) NEXT NODE IDENTIFICATION VECTOR
(4) CURRENT NODE IDENTIFICATION VECTOR AND DELAY
(5) NUMBER OF DEFLECTIONS SUFFERED
(6) RESIDUAL HOP COUNT
(7) THE INFORMATION OR PAYLOAD OR MESSAGE
(8) PACKET SEQUENCE NUMBER
(9) ERROR CORRECTING CODE

Figure 5.5: The Format of a Packet.
Figure 5.6: A Threshold Function to Determine the Priority Category of a Packet Based on its Sequence Number.

For example, the routing preference vector for the packet in the first incoming link (i.e., $I_1$) is: $[I_1-O_1 (f_{11}, a_{11}), I_1-O_2 (f_{12}, a_{12}), I_1-L_0 (f_{10}, a_{10})]$. The notation $I_1-O_1 (f_{11}, a_{11})$ indicates that the connection from the first incoming link (i.e., $I_1$) to the first outgoing link (i.e., $O_1$) supported with a score of favor ($f_{11}$) and protested by the factor against ($a_{11}$).

Note that $0 \leq (f_{11} + a_{11}) \leq 1$, and the remaining part, if any, accounts for not sure or uncertainty. Figure 5.10 shows the computation of the routing preference vector for the packet in the first incoming link (i.e., $I_1$). Similar computations are also performed simultaneously for the packets in the other two incoming links. For example, if the packet in the first incoming link (i.e., $I_1$) has suffered $[0.5 (N_{df})_{max}]$ number of deflections, its priority category is 3 as obtained from Figure 5.8. If the residual hop count for this packet
is \leq (D/4), its priority category is 4, as found from Figure 5.7b. Using Figure 5.6 we find that this packet's priority category is 1, assuming that the packet's sequence number is \leq (Maxm/3). Since we are assuming that this is a transit packet, it is required to compute the addresses of the nodes, i.e., \((r_{\text{nxt}}, c)\) and \((r, c_{\text{nxt}})\), connected to the current node \((r, c)\) using the formulations given in [104].

\[
r_{\text{nxt}} = [r - (1 - c \mod 2)(1 - m \delta(r + (m/2) - 1)) + (c \mod 2)(1 - m \delta(r - (m/2)))]
\]

\[
c_{\text{nxt}} = [c - (1 - r \mod 2)(1 - n \delta(c + (n/2) - 1)) + (r \mod 2)(1 - n \delta(c - (n/2)))]
\]

where \(m\) and \(n\) represent the numbers of rows and columns in the network, respectively, and \(\delta(.)=1\) only when the argument is zero.

\[
\text{PRIORITY CATEGORY}
\]

\[
\text{CURRENTNODE-DESTINATION SEPARATION}
\]

\[
\text{Figure 5.7a: A Threshold Function to Determine the Priority Category of a Packet Based on the Residual Hop-Count.}
\]
Now, if the packet is currently at node (1,0) and its destination is (0,0), then assuming that the first outgoing link (i.e., \( O_1 \)) leads to the destination, we have: \( f_{11} = (3+4+1)/11 = (9/11) \) and \( a_{11} = (3-3)+(4-4)+(3-1)/11 = (2/11) \). The uncertainty is zero in this case. Thus, the first output of Figure 5.10, i.e., \( l_1-O_1 (f_{11}, a_{11}) \) is \( l_1-O_1 (9/11, 2/11) \).

![Current Node-Destination Separation](image)

**Figure 5.7b:** Another Threshold Function to Determine the Priority Category of a Packet Based on the Residual Hop-Count.

The equivalence between fuzzy logic based categorization and threshold based group membership determination is as shown in Figure 5.11. The task of the ART-based arbiter is to select a switching pattern (out of the 26 possible configurations as determined in the Appendix-C, if a 3-in-3-out switch is considered or out of the 57 patterns if physical 3×3
but virtual 3x4 switch is used) on the basis of the input vector, and to feed the information regarding the SP from the R-Layer (see Figure F-1 and Figure 5.4) to the switch.

The selected switching pattern is then established with minimum disruption, unless it is explicitly requested, of the existing connections. It is the latter requirement, i.e., the maintenance of the established connection, which calls for a modification (i.e., adding a pre-processor) of the original ART-1 based pattern recognition machine. Since the ART-1 machines are capable of fast learning and directly accessing the learned pattern, it is expected that the decision regarding the switching pattern (or connection configuration) to be established is fed back to the switch as fast as demanded by the transmission links.

![Threshold Function Diagram]

Figure 5.8: A Threshold Function to Determine the Priority Category of a Packet Based on the Number of Deflections Suffered by it.
At this point it is worth mentioning that the parameters which affect the learning, i.e., determination and adaptation of the synaptic weights, are to be selected very carefully (e.g., L=2, as discussed in Appendix-F). We find the range of these parameters by conducting extensive simulation experiments. These ranges and the corresponding priority categories are as shown in Figures 5.6 through 5.8.

\[
\Phi_{TR}(X) = \begin{cases} 
\frac{1}{1 + e^{-(x-\gamma)/T}} & \text{if } (\gamma - a) \leq x \leq (\gamma + a) \\
1 & \text{if } (\gamma + a) \leq x \leq (\theta - b) \\
\frac{1}{1 + e^{+(x-\theta)/T}} & \text{if } (\theta - b) \leq x \leq (\theta + b)
\end{cases}
\]

Figure 5.9: A Generalized Uni-Modal Quasi-Trapezoidal Threshold Function.
Figure 5.10: Fuzzy Computation of the Routing Preference Vector of the Packet in the Incoming Link # 1, i.e., $I_1$. 
Figure 5.11: Equivalence Between Multi-Level Based Decision and Fuzzy Logic Based Decision.
5.5.2b Results

It is well-known [78] that for a wxw MSN, (w is the width, i.e., the number of nodes per row or column of the network) the total number of nodes is, \( N = w^2 \), and the total number of hops is \( 2 \times N \). The girth of this network is 4 hops, and its diameter, \( d_{max} \) is either \( w \) hops or \( (w+1) \) hops depending on whether \( w \) is equal to \((4x+2)\) or \(4x\), where \( x \) is a non-negative integer. Therefore, for a 4x4 MSN, \( N=16 \), Diameter=5 hops, and Mean Inter-Node Distance (MIND)=2.93=3 hops. For a 6x6 MSN, the corresponding parameters are, \( N=36 \), \( d_{max} = 6 \) hops, and MIND=3.7143=4 hops. In particular, the IND distribution for a 4x4 MSN is as presented in Table 5.1 and plotted in Figure 5.12.

Next, we present the variation of the normalized value of the MPTT with the offered load. For example, the MIND at zero load is 3 hops for a 4x4 MSN, and this network can support at most 62.5\%, i.e., 10-out-of-16, of the terminal equipments with traffic burstiness of unity. Table 5.2 shows the variation of the normalized values of the MPTT with the offered load. The corresponding results are as plotted in Figure 5.13.

### Table 5.1: Number of Nodes, \( N_i \), Separated by i Hops in a 4x4 MSN. Note that the Diameter of a 4x4 MSN is 5 Hops [78].

<table>
<thead>
<tr>
<th>Number of Hops, i.</th>
<th>Number of nodes, ( N_i ), separated by i hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure 5.12: Inter-Node Distance Distribution for Uniform Traffic Distribution (UTD) in a 4x4 Manhattan Street Network.

Table 5.2: Variation of the Normalized - with Respect to the Mean Delay at Zero Load - Mean Packet Transfer Time (MPTT) with the Number of Active User Pairs. MPTT is Measured in Terms of Hop-Count, as Defined Earlier in Section 5.5.

<table>
<thead>
<tr>
<th>Percentage of 'maximum allowable number of users with burstiness=1' communicating Actively (a ratio ≤ 1.00)</th>
<th>Normalized MPTT without using the Neural Arbiter and Buffer Stealer.</th>
<th>Normalized MPTT with using the Neural Arbiter and the Buffer stealer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.10</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.20</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.30</td>
<td>1.15</td>
<td>1.05</td>
</tr>
<tr>
<td>0.40</td>
<td>1.30</td>
<td>1.20</td>
</tr>
<tr>
<td>0.50</td>
<td>1.80</td>
<td>1.22</td>
</tr>
<tr>
<td>0.60</td>
<td>2.40</td>
<td>1.40</td>
</tr>
<tr>
<td>0.70</td>
<td>4.00</td>
<td>1.60</td>
</tr>
<tr>
<td>0.80</td>
<td>8.00</td>
<td>1.80</td>
</tr>
<tr>
<td>0.90</td>
<td>14.0</td>
<td>3.20</td>
</tr>
<tr>
<td>0.96</td>
<td>-</td>
<td>6.00</td>
</tr>
</tbody>
</table>
Figure 5.13: Variation of the Mean Packet Transfer Time (MPTT) with the offered Load in a 4x4 Manhattan Street Network (MSN).

Our next investigation concerns network throughput. The corresponding results are presented in tabular and graphical forms in Table 5.3 and Figure 5.14, respectively.
Table 5.3: Variation of Network Throughput with Normalized Number of Active User Pairs.

<table>
<thead>
<tr>
<th>Percentage of users communicating Actively (a ratio ≤ 1.00)</th>
<th>Network Utilization without using the Neural Arbiter and Buffer Stealer</th>
<th>Network Utilization with using the Neural Arbiter and the Buffer stealer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.05</td>
<td>0.25</td>
<td>0.45</td>
</tr>
<tr>
<td>0.10</td>
<td>0.30</td>
<td>0.57</td>
</tr>
<tr>
<td>0.20</td>
<td>0.35</td>
<td>0.73</td>
</tr>
<tr>
<td>0.40</td>
<td>0.37</td>
<td>0.80</td>
</tr>
<tr>
<td>0.60</td>
<td>0.39</td>
<td>0.83</td>
</tr>
<tr>
<td>0.80</td>
<td>0.40</td>
<td>0.84</td>
</tr>
<tr>
<td>1.00</td>
<td>0.40</td>
<td>0.84</td>
</tr>
</tbody>
</table>
Figure 5.14: Variation of Network Throughput with the offered Load in a 4x4 Manhattan Street Network (MSN).

The variation of the normalized network access delay with the number of active user pairs, i.e., the offered load, is as shown in Figure 5.15a. The probabilities of allowing and refusing local packets' access to the network are as shown in Figure 5.15b.
Figure 5.15a: Variation of the Network Access Delay with the Offered Load in a 4x4 Manhattan Street Network (MSN) when Neural Arbiters are Used.
Figure 5.15b: Variation of the Probabilities of Allowing and Denying Network Access with the Offered Load in a 4x4 Manhattan Street Network (MSN) when Neural Arbiters are Used.

It appears that with the neural arbiter and buffer stealer at node, the local packets are allowed to have access to the network even at moderately high (~63%) load. This is a noticeable amount of improvement compared to the local access schemes investigated elsewhere [79] in this type of networks.
We now investigate the effect of burstiness of the active users on the percentage of users who are allowed to be in the communicating mode within the network. As expected, with a decrease in the burstiness ratio (i.e., activity) of the users, more users can be allowed to have access to the network. These results are shown in Figure 5.16.

![Graph showing the relationship between burstiness ratio and percentage of users in active (burst) mode.]

**Figure 5.16:** Effect of Burstiness of the Active Users on the Maximum Allowable Number of Communicating Users in a 4x4 Manhattan Street Network (MSN) when Neural Arbiters are Used.
It is also found (Figure 5.16) that as the traffic sources become more and more bursty, i.e., the activity of the terminals reduces, we can allow more users to communicate with the same amount of available resources without severely degrading the quality of service. Figure 5.16 is drawn with the assumption that an amount of packet loss ≤ 5% is tolerable for the pre-specified GOS or QOS, as needed for packetized voice communication [7, 53, 54, 71, 131].

In our scheme, the operation of the network starts with the MSN (a small girth, multi-connected, and algebraically definable topology) as the initial (or seed) soft-topology, the nodes then learn to dynamically maintain the connection between the communicating source-destination pairs by implementing distributed cut-through of the intermediate nodes. That is, the arbiter learns to adapt the decision functions to the dynamics of the network by continuously monitoring network operation.

5.6 Conclusions

Although we have considered a 3x3 switch, the proposed method can be easily extended for efficient operation of larger switches as can be demanded by the emerging ATM (Asynchronous Transfer Mode) switching. It is not required to train the arbiter with larger number of switching patterns with an increase in the switch size as needed in the method proposed by Hiramatsu [62], and Marrakchi and Troudet [106]. This feature of our nodal operating system can be attributed to the on-line learning capability of the ART-1 based arbiter.

We also find that, in supra-high-speed packet-switched networks, the end-to-end flow control on the basis of dynamically maintained, as policed by the buffer stealer and
the arbiter, window size outperforms the *per-link flow control* method. The size of the window should depend on both the *source-destination separation* in the soft-topology and the *dynamics of the network*.

The avoidance of network congestion is achieved in our scheme because of the following reasons:

• **Very strict access assignment and control**, i.e., automatic throttling of local packets due to the *Buffer Stealer* in the nodes,

• **Fast and intelligent forwarding** of the transit packets on the basis of multiple criteria as implemented by the modified ART-1 based arbiter, and

• **Packet dropping** after they have travelled a pre-specified number of hops as policed by the pre-processor of the arbiter at the network nodes.

That is to say, we avoid network congestion by embedding the required controls in the network access and transit packet processing mechanisms. The probability of dropping transit packets substantially reduces because of *very strict access control*, and the transit packets are *almost never* blocked.
Chapter 6

SUMMARY, CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

This concluding chapter presents an assessment of the investigations conducted during the course of this dissertation. We also discuss future research topics which can be considered as natural extensions of the methodologies presented herein.

6.1 Contributions of this Dissertation

In this section we enlist the contributions of this dissertation with special emphasis on the novelties of our approaches and ideas.

A classification of the research efforts up to January 1992 for designing large supra-high-speed packet switching networks from the point of view of combined architectural and operational features is presented in the Chapter 2. While all other previous attempts [20, 41, 127, 130, 139, 141] for such categorization have considered either architectural or operational issues independently, our approach [81] of considering the combined effect in light of the future networking trends (described in the Chapter 1) appears to be more realistic.
Our next contribution is the development of a new analytical technique for evaluating network architectures. We have successfully [78, 85] applied this technique for evaluating the architecture of a highly promising (because of its small girth\(^\$\)) lightwave network structure called the Manhattan Street Network (MSN) [102-104] -- for different traffic distribution patterns. The advantages of using regular logical topologies have been discussed in Chapter 1. It is also shown in Chapter 3 how important a role traffic distribution patterns play in the architectural properties of a network.

Our third major contribution is analytical modeling of deflection routing in supra-high-speed regular digraphs. The applicability of the proposed method has been demonstrated by evaluating a deflection routing in two types of MSNs for UTD and ZTD patterns. The results agree well with our simulation observations as shown in Chapter 4. The little discrepancy between the simulation results and the analytical results can be explained due to the assumptions that (i) the total number of deflections are identically distributed among the intermediate or subdestination nodes, and (ii) uniform distribution of the probabilities of success (under Bernoulli-based approximation with independent trials) throughout the network. Despite these mis-matches the results are highly acceptable. This is because our analytical formulations can quickly provide reasonable estimations of the MPTT and throughput (or utilization) for networks with arbitrarily large number of nodes.

The fourth and final contribution of this research is regarding efficient operation of large supra-high-speed packet-switched networks under both certain and uncertain conditions using the Artificial Neural Network (ANN) based arbiters. We have employed Grossberg's ART-1 based ANN along with fuzzy logic based pre-processor for congestion-free operation of networks. This is achieved via dynamic reconfiguration of the

\(^\$\) Girth of a digraph is defined as the maximum over all nodes of the length of the shortest non-trivial path from a node back to itself.
soft-topology to match the demanded traffic matrix and pattern as closely as possible. Thus, fast and intelligent processing of transit packets and parsimonious access of newly generated packets allow us to achieve near-perfect network operation. The utility of our technique has been demonstrated via simulation in Chapter 5. Direct analytical models of such schemes are expected to be very complicated in order to be desirable and useful.

This concludes the summary of the contributions of this dissertation along with supporting references.

6.2 Suggestions for Future Research

As has always been the case, any novel approach to solve one problem sometimes opens gateway(s) to numerous other problems. Ours is no exception. Consequently, the author mentions few of the relevant research topics worthy of future investigations.

The new analytical technique [85] presented in Chapter 3 has been applied to regular networks only. Although this method allows one to determine the architectural properties of any regular network of arbitrarily large size, it has not been applied to any irregular network.

Our simulation experiments reported in the Chapters 4 and 5 show that there exists a direct interaction between the traffic distribution pattern and the packet routing preference (or rule) for a given soft-topology. It would be very interesting to see whether it is possible to find any direct analytical relationship in this regard. Although it is expected that the resulting relationship may be quite complex in nature.
We have not considered very large (e.g., a network with $\geq 1000$ TEs) MSNs in our investigation. In such a case, the network can be partitioned into communities of interest on the basis of traffic matrix. As an example, the communities of interest could first be optimised followed by an optimization of the interconnection(s) among the communities. Genetic algorithms or simulated annealing based algorithms [128] with a suitably defined annealing schedule can be employed for the purpose of optimization at both stages.

For large networks the information obtained from the neighboring nodes should be given higher importance (i.e., weight) than those extracted from the nodes which are further away. This may make the ART-based arbitration very useful in large high-capacity networks.

Our investigations have been limited to one type of traffic only. Simulation and analytical treatment of dynamic reconfiguration of soft-topology of networks with various types of traffic demands and patterns in an integrated environment (i.e., for a network supporting data, facsimile, video, and voice etc., types of traffic) are left as the materials for future investigations. The effect of traffic burstiness in integrated networks is another interesting arena which deserves further attention.

It is now clear [see e.g., 23, 43, 59, 84, 88, 99, 115] that the future lightwave networks will make extensive use of the emerging fine-grained WDMed channels for internode communications. The access assignment and transmission capacity allocation will be highly flexible (since the nodes can be taught [84]) in these networks which make them very attractive for future multi-service communications.
It would be very interesting to find out whether the network can learn its operating rules from the environment. Both off-line training and on-line (or background) learning which function like the human nervous system should be implementable.

This type of approach will definitely generate vigorous enthusiasm among the researchers in network design techniques, maintenance, and operations. Because the incorporation of natural intelligence in network operating system not only renders network management efficient but also helps to maintain the pre-specified level of Grade-Of-Service (GOS) even during the most unexpected network conditions. These conditions may include sudden load surge, as on Christmas day, isolated node or link failure, zonal network black-out due to an earthquake or any other natural disaster.
Appendix-A

SC-TC TRADEOFF IN THE MSNs
AND
EMBEDDING MSNs ONTO RING NETWORKS

In this Appendix, queueing-theoretic formulations for trading nodal Storage Capacity (SC) requirements with the incremental Transmission Capacity (TC) of the links in supra-high-speed packet-switched communication networks are presented. The performance criterion used is the mean nodal forwarding time. The results are useful for designing all-optical packet-switching telecommunication networks where the designer needs to compute the amount of Incremental TC (ITC) needed to eliminate a pre-specified number of nodal buffers (SC) while preserving the same mean nodal forwarding time. A single optical buffer can be easily implemented by using a simple optical delay line. An example is included using a very-high-speed regular di-graph as an architecture of macro-switching-fabric. [Note that the definitions of variables are local to this Appendix].

Ahmadi and Denzel [2] present an up-to-date survey of the fast packet switching fabrics and categorizes them into the following six groups: (i) Banyan and buffered banyan based fabrics, (ii) Sort-banyan based fabrics, (iii) Fabrics with disjoint path topology and output queues, (iv) Cross bar based fabrics, (v) Time division based fabrics with common packet memory, and (vi) Fabrics with shared medium. Note that the architecture of most of these fabrics can be used as logical (or virtual or soft) topology for the emerging large supra-high-speed packet switching networks. The basic difference is, of course, that in switching fabrics the 'propagation delay' constitutes a negligible portion of the inter-node packet transfer time, whereas the opposite is true for large networks (since the inter-node
distance is much higher, i.e., the links are longer). Consequently, the service time normalization constants are different.

A.1 Problem Statement And The Proposed Procedure

In this section, we present the problem statement, and a six-step procedure for investigating the proposed SC-TC tradeoff in supra-high-speed packet switching fabrics/networks. We investigate the trade-off between buffer size and the incremental transmission capacity. It is assumed that a feedback path exists (this is true for most of the emerging fast switching fabrics and networks) in the net, and it is required to find the incremental speed of the feedback path for eliminating buffer(s) in the forward path, keeping the mean forward path passage time the same. Note that, the basic difference between fabrics and networks is that in switching fabrics the ‘propagation delay’ constitutes a negligible portion of the inter-node packet transfer time, whereas the opposite is true for large networks (since the inter-node distance is much higher, i.e., the links are longer). Consequently, although the service time normalization constants are different, the same model can be applied to both fabrics and networks.

The possibility of trading the nodal storage capacity with the incremental transmission capacity can be attributed to the current availability of cost-effective fiber optics transmission media which offer abundant transmission bandwidth. Although this kind of trading could have been theoretically performed in the past, it would have been of no practical use because the electronic transmission media was expensive compared to the electronic storage and processing capability.
The proposed procedure involves the following steps:

**STEP 1:** Consider a two-dimensional segment of a high-speed packet-switching fabric or network. This segment must have\(^1\) some provision for deflecting the overflow packets rather than dropping them from the network because of limited waiting room at the nodes. The deflected packets are allowed to reenter the logically/virtually original entry point (i.e., the node) for the purpose of getting forwarded\(^2\) to the desired nearest neighbor. Then develop an exact or approximate - if the exact model is not a tractable one - equivalent queueing model for this network segment with a forward path and a feedback path (to accommodate the deflected packet). Both the paths are assumed to be lacking of any storage except for the packet which is being served currently, i.e., serving buffer only.

**STEP 2:** Analytically determine the mean nodal forwarding time, i.e., the mean system time \(E(S_d)\), for the queueing system developed in step 1. Then investigate how \(E(S_d)\) is affected by the packet arrival rate and the ratio of mean service rates of the servers in the feedback and feedforward paths of the system under consideration.

**STEP 3:** Now, consider a non-zero-size waiting buffer in the forward path instead of the zero buffer one in the queueing model developed in step 1. Then conduct all the necessary performance evaluations similar to those executed in step 2 for the modified system. This additional buffer is incorporated to calculate the delay-related performance gain, later on this buffer is traded off with the ITC in the feedback path while maintaining the same performance gain.

---

\(^1\) we realize that there exists some FPS fabrics - see e.g., [2], [26], and [30] - which do not have any feedback path at all, and that they tackle the congestion or overflow problem by dropping bits (or packets) or by controlling access via permits or tokens. Our investigation does not pertain to them.

\(^2\) it is true that because of the transmission speed range considered, some new arrivals - through the forward path during the intermittent period - might cause out-of-sequence disposals (from the segment under consideration) of the packets. The issue of resequencing out-of-order packets in these scenarios is presumably taken care of in an end-to-end manner, and is beyond the scope of this investigation.
STEP 4: Develop analytical relationships to compute the change in the parameters of the system considered in step 1 to have the same performance as that of the system constructed in step 3 in the sense of preserving the mean nodal forwarding time. These relationships must consider a wide range of offered load, i.e., the packet arrival rate. For example, we may vary $\lambda$ from 0.10 to 0.90 packets/slot (or unit time).

STEP 5: Utilize the relationships developed in step 4 to design buffer-free high-speed packet switching fabrics/networks whose delay related performance is the same as that of the non-zero buffer network segment considered in step 3.

STEP 6: Repeat steps 3 through 5 for a number of non-zero buffer cases.

A.2 Analysis of Zero And Non-Zero Buffer Cases

This section deals with steps 1, 2, and 3 of the procedure presented in section A.1. In particular, sub-section A.2.1 considers steps 1 and 2, and sub-section A.2.2 deals with the third step.

A.2.1 The Zero (Waiting) Buffer Case

We consider a regular mesh di-graph of the Manhattan street type as shown in Figure A-1. Such di-graphs have been proposed for high-speed metropolitan or wide area level packet switching networks in [103]. Since the deflection index of this network is four, the incremental gain in performance with more than four buffers -- at the input of the outgoing links of a node -- is not attractive. The topological properties of such architectures are discussed in [78], and the references cited there.
Figure A-1: A 4x4 Switching Fabric (or Network). The Encircled Segment is Used for Studying the Proposed SC-TC Trade-off (Adapted from [103]).

In this study, we focus on the encircled segment of Figure A-1. The exact queueing model for this segment is as shown in Figure A-2. The exact formulation needs to consider [103] that the inter-arrival times are exponentially distributed, the service times are deterministic or constant, and a limited waiting room in front of the server. We approximate the key features of Figure A-2 by the model shown in Figure A-3, where exponentially distributed service times have been assumed. The basis of our approximation is the fact that the M/M/1/K type queue provides a reasonable upper bound measure for the performance parameters of interest of an M/D/1/K type queue, as shown in Figure A-4. We also justify the approximation of Figure A-2 by Figure A-3 in section A.2.1.1.
Figure A-2: The Exact Queueing Model for the Encircled Segment of Fig.A-1. All the Queues are of M/D/1/K type, where K is the System Size in Number of Packets.

Figure A-3: A Highly Approximate Queueing Model for the Encircled Segment of Fig.A-1. All the Queues are of M/M/1/K type. A Proof of this Approximation is Presented in section A.2.1.1. The Same Results Can Also be Obtained by Applying Elementary Superposition Theorem. The Buffers in Front of the Servers A and B are only to Hold the Packets in Service, i.e., they are serving buffers.
Figure A-4: Demonstration of the Empirical Results that Delays Suffered in an M/D/1/K Queue are Always Upper-Bounded by those in an M/M/1/K Queue.

In Figure A-3, packets enter and proceed directly to receive service at A if A is not busy. In this case, there is no additional buffer (except for the one which is being served currently) in front of the server A. If A is busy, at most one packet is allowed to be recycled through the system, at any particular point in time, to try to return to server A later. This is modeled by the loop with the server B. The packet being recycled will eventually complete its service at B and return to server A to see whether A is idle or not. When server A is busy and there is a packet at server B, all other packets are blocked\(^3\) from the system. If server A is free and server B is busy, a new packet "may" or "may not" (no re-sequence problem will arise) go directly to server A. We consider the former strategy here. We control\(^4\) the system so that the proportion of packets blocked is kept very small.

Let \(n_A\) and \(n_B\) represent the number of packets in front of the server A and B, respectively. Note that both \(n_A\) and \(n_B\) can take on the value of 0 or 1. The loop in which B

---

\(^3\) from entering the segment under consideration.  
\(^4\) by restricting access, i.e., by tuning the input rate.
occurs represents the packets which recycle through the system in attempts to receive service from A. When a packet completes service at B, it enters service at A if there is a space available. If not, it recycles to B again and must complete another service at B in order to get another chance\(^5\) to enter the service at A. When a packet completes service at A, it leaves the node. Arrivals to the node will enter service at A if there is space available. If not, they will loop and enter service at B if there is space available. If there is no space at either A or B, then an arrival will be blocked from entering the node, and therefore must enter another neighboring node.

The states of the system consist of all possible pairs \(\{(n_A, n_B)\}\), i.e., \{(0, 0), (0, 1), (1, 0), (1, 1)\}. The state transition diagram is shown in Figure A-5. Assuming that the packet arrival follows a Poisson process with rate \(\lambda\) and service times at A and B are exponentially distributed with rates \(\mu_1\) and \(\mu_2\), respectively, the limiting probabilities of being in state \((i, j)\), i.e., \(p_{ij}\), can be obtained by solving the equilibrium equations; one for each of the states of Figure A-5. The balance equations are:

\[
\begin{align*}
\mu_1 p_{10} &= \lambda p_{00} & \text{for state (0,0)} \\
\lambda p_{00} + \mu_2 p_{01} &= (\lambda + \mu_1) p_{10} & \text{for state (1,0)} \\
\mu_1 p_{11} &= (\lambda + \mu_2) p_{01} & \text{for state (0,1)} \\
\lambda (p_{10} + p_{01}) &= \mu_1 p_{11} & \text{for state (1,1)}
\end{align*}
\]

And, finally, \(p_{00} + p_{01} + p_{10} + p_{11} = 1.0\)

\(^5\) we are assuming that the packets generated by the active sources are not dropped until a pre-specified lifetime is spent by them in the network.
The results are as follows.

\[ p_{10} = \frac{\lambda}{\mu_1} p_{00}, \quad p_{01} = \frac{\lambda^2}{\mu_1 \mu_2} p_{00}, \quad p_{11} = \frac{\lambda^2 (\lambda + \mu_2)}{\mu_1^2 \mu_2} p_{00}, \]

where \( p_{00} = \frac{\mu_2^2}{\lambda^3 + \lambda^2 (\mu_1 + \mu_2) + \lambda \mu_1 \mu_2 + \mu_1^2 \mu_2}. \)

The probability that a packet is blocked, \( P(\text{packet blocked}) = p_{11}, \) and the probability that the server A is idle, is \( P(\text{A is empty}) = (p_{00} + p_{01}). \)

![State Transition Diagram for the Queueing System of Fig.A-3.](image)

Figure A-5: State Transition Diagram for the Queueing System of Fig.A-3.

Now, assuming that \( S_0 \) be the service completion time of a packet which has just arrived at the system given that it is not blocked, the mean nodal forwarding time can be represented by:

\[ E(S_0) = \frac{[2\lambda^2 + \lambda (\mu_1 + 2\mu_2) + \mu_1 \mu_2]}{[\lambda^2 \mu_1 + \lambda \mu_1 \mu_2 + \mu_1^2 \mu_2]} . \]

The proof of the above results is given in section A.2.1.2. We now investigate the effect of the packet arrival rate, \( \lambda \) on \( E(S_0) \) for various values of the ratio \( [\mu_2/\mu_1] \) (we define this ratio as the parameter \( a \)), assuming that \( \mu_1 = 1.0 \). The results are presented in Figure A-6.

Note that, under this assumption, the utilization (defined as the ratio of arrival rate to service rate) of the server in the forward path is \( \lambda (1 - p_{11}) \) and \( a=\mu_2 \). Next, we find how \( E(S_0) \) is affected by the parameter \( a \). The results are presented in Figure A-7.
Figure A-6: Effect of Packet Arrival Rate, \( \lambda \), on the Normalized Mean Nodal Forwarding Time, \( E(S_\theta) \), for the Queueing System Presented in Fig.A-3.

Figure A-7: Effect of the Parameter, \( a \), on the Normalized Mean Nodal Forwarding Time, \( E(S_\theta) \), for the Queueing System Presented in Fig.A-3.
A.2.1.1 Justification of the Approximations Used in This Section

We now show the validity of the fact that the behavior of an M/D/1 queueing system can be safely approximated by that of an M/M/1 queue for the environment under consideration.

Let $p_a$ be the probability that a local node, i.e., a node of Figure A-2, is the final destination, $p_b$ be the probability that packet exits from the local node out of the loop, and $p_c$ be the probability that a packet originates at one of the 4 nodes, during the time interval when a packet is moving, for example, from buffer P to the buffer Q. Also, let $f(x)$ represents the probability density function for the time (measured from buffer P) for a packet to enter the buffer A.

Assume that each channel of the loop of Figure A-2 has an input buffer to hold at most K packets. Note that, generally, $p_a$ will be small relative to $p_b$ (depending on the number of nodes in the network), and $p_c$ will be small relative to $p_b$ (since we are assuming supra-high-speed (>10^8 bps) packet (e.g., 53 bytes) transmission, the time interval is very small compared to the inter-arrival times). Furthermore, the event that a local node is a final destination and the event that a local node generates a packet will offset each other probabilistically, for the uniform traffic distribution case. This can also be verified by simulation. Thus we will neglect $p_a$ and $p_c$, and consider only $p_b$. The value of $p_b$ will be at least 0.5, since at each node, a packet has 2 possible routes along which it could proceed.

If a packet exits at any particular node (and it does with probability $p_b$), we identify the next entering item at node 4 with the exiting item. The time interval between these two events is exponentially distributed since the interarrival times at node 4 are exponentially
distributed. This identification (or pairing) will give the same steady state distribution of the buffer occupancy at A. Under this identification (or *marking*) procedure, the time for the packet (or its paired equivalent) to enter buffer A, measured from buffer P has a distribution which can be described as follows.

Rather than being a sum of 4 sets (for the 4 channels or servers) of up to \( K \) deterministic times (since each queueing system is assumed to be of size \( K' \)), we now show that we obtain a linear combination of shifted exponential distributions. Let \( d \) be the deterministic service time (excluding waiting time) for a single packet at a channel. If, for example, a packet enters buffer P and must wait for 1 previous packet (with probability \( p_1 \)), the time to complete service at P would be \( 2d \) (\( d \) for the previous packet, and \( d \) for the current one). Assume that the packet immediately proceeds to an empty buffer Q. The probability of proceeding past node 1 is \( (1-p_0) \). The probability of the empty buffer Q is \( p_0 \). The time to complete service at Q is again \( d \), so we have a cumulative sum of \( 3d \) to this point. Assume that the packet exits at node 2 (with probability \( p_0 \)).

We then *identify* the exiting packet with an entering packet at node 4. But packets are generated or arrive at node 4 according to a Poisson process (with exponential interarrival times) so the total time involved has an exponential distribution shifted by \( 3d \) time units. The probability attached to this particular configuration, for \( t \geq 3d \), is:

\[
[p_1(1-p_0)p_0p_0e^{-\lambda(t-3d)}].
\]

Similar components can be obtained for all other configurations. The actual probability density will be the sum of these components. For the general case, assume the probability of \( i \) packets in the buffer of any channel is \( p_i \) for \( i=0,1,\ldots, K \).
Then the probability density of the time for a packet to reach \( A \) from \( P \) is:

\[
f(x) = \left[ \sum_{i=0}^{K} p_i \lambda e^{-\lambda(x - id)} \right] \\
+ (1 - p_b) \sum_{j=0}^{K} \sum_{i=0}^{K} p_i p_j \lambda e^{-\lambda(x - (i+j)d)} \\
+ (1 - p_b)^2 \sum_{k=0}^{K} \sum_{j=0}^{K} \sum_{i=0}^{K} p_i p_j p_k \lambda e^{-\lambda(x - (i+j+k)d)} \\
+ (1 - p_b)^3 \sum_{l=0}^{K} \sum_{k=0}^{K} \sum_{j=0}^{K} \sum_{i=0}^{K} p_i p_j p_k p_l \lambda e^{-\lambda(x - (i+j+k+l)d)} \right].
\]

Since the service time \( d \) for a single packet is negligible relative to the interarrival time, in the limiting case, corresponding to the supra-high-speed packet transmission, the density function \( f(x) \) can be reasonably approximated by an exponential density function. And, as shown in Figure A-4, the delay related performance of an \( M/D/1/K \) queue can be always safely approximated by that of an \( M/M/1/K \) queue, since the latter provides the upper bound values for the performance parameters of interest.

This is the justification and validation for replacing the exact model consisting of four deterministic channels (or servers, with buffers) by two servers -- one in the forward path and the other in the feedback path -- with exponential service times. This is a valid approximation for supra-high-speed packet transmission.

**A.2.1.2 Determination of the Expression for \( E(S_0) \)**

Let \( K \) be the amount of time for a packet to complete service if it encounters the \((1, 0)\) state when it enters the system. Then:
\[ E(S_0) = \frac{1}{1 - p_{11}} \left( \frac{1}{\mu_1} P(A \text{ empty}) + p_{10} E(K) \right) \]

where \( P(A \text{ empty}) = (p_{00} + p_{01}) \). The expression for \( E(K) \) is to be obtained now. In the following discussion, we assume that A and B refer to service completions at servers A and B, respectively, and N refers to a new arrival. Then:

\[
E(K) = \sum_i P(\text{event } i) E(K|\text{event } i)
\]
\[
= P(B)E(K|B) + P(AB)E(K|AB)
+ P(ANB)E(K|ANB) + P(ANAB)E(K|ANAB)
+ P(ANANB)E(K|ANANB) + \ldots.
\]

Now, since we have two independent Poisson processes with rates \( \mu_1 \) and \( \mu_2 \), then the expected time until the first of either event occurs is \( [1/(\mu_1 + \mu_2)] \). If we are given with the event which occurs first, then the expected time until that event occurs must still be is \( [1/(\mu_1 + \mu_2)] \). Using this analysis, we have:

\[
E(K|B) = \frac{1}{\mu_1 + \mu_2} + E(K)
\]
\[
E(K|AB) = \frac{1}{\mu_1 + \mu_2} + \frac{1}{\lambda + \mu_2} + \frac{1}{\mu_1}
\]
\[
E(K|ANB) = \frac{2}{\mu_1 + \mu_2} + \frac{1}{\lambda + \mu_2} + E(K)
\]
In general, we have

\[ E(K|\langle AN \rangle^i AB) = \frac{i + 1}{\mu_1 + \mu_2} + \frac{i + 1}{\mu_2 + \lambda} + \frac{1}{\mu_1} \quad i = 0, 1, \ldots \]

\[ E(K|\langle AN \rangle^i B) = \frac{i + 1}{\mu_1 + \mu_2} + \frac{i}{\mu_2 + \lambda} + E(K) \quad i = 0, 1, \ldots . \]

Also

\[ P((AN)^i AB) = \left( \frac{\mu_1}{\mu_1 + \mu_2} \right)^i \left( \frac{\mu_1}{\mu_1 + \mu_2} \right) \left( \frac{\mu_2}{\mu_2 + \lambda} \right)^i \quad i = 0, 1, \ldots \]

\[ P((AN)^i B) = \left( \frac{\mu_1}{\mu_1 + \mu_2} \right) \left( \frac{\lambda}{\mu_2 + \lambda} \right)^i \frac{\mu_2}{\mu_1 + \mu_2} \quad i = 0, 1, \ldots \]

So

\[ E(K) = \sum_{i=0}^{\infty} P((AN)^i AB) E(K|\langle AN \rangle^i AB) + \sum_{i=0}^{\infty} P((AN)^i B) E(K|\langle AN \rangle^i B) \]

\[ = \sum_{i=0}^{\infty} \left( \frac{\mu_1}{\mu_1 + \mu_2} \right)^i \left( \frac{\mu_1}{\mu_1 + \mu_2} \right) \left( \frac{\mu_2}{\mu_2 + \lambda} \right)^i \left( \frac{i + 1}{\mu_1 + \mu_2} + \frac{i + 1}{\mu_2 + \lambda} + \frac{1}{\mu_1} \right) \]

\[ + \sum_{i=0}^{\infty} \left( \frac{\mu_1}{\mu_1 + \mu_2} \right) \left( \frac{\lambda}{\mu_2 + \lambda} \right)^i \frac{\mu_2}{\mu_1 + \mu_2} \left( \frac{i + 1}{\mu_1 + \mu_2} + \frac{i}{\mu_2 + \lambda} + E(K) \right) \]

Simplifying gives

\[ E(K) = \frac{(\mu_1 + \mu_2 + \lambda)(2 \mu_2 + \lambda + \mu_1)\mu_2}{(\mu_1 \mu_2 + \mu_2^2 + \mu_2 \lambda)^2} + \frac{\mu_2^2 + \mu_2 \lambda}{\mu_1 \mu_2 + \mu_2^2 + \mu_2 \lambda} E(K). \]

Solving for \( E(K) \) gives

\[ E(K) = \frac{2 \mu_2 + \lambda + \mu_1}{\mu_1 \mu_2}. \]

Utilizing the above equation for \( E(K) \), we obtain the expression for the mean nodal forwarding time \( E(S_0) \).
A.2.2 A Non-Zero (Waiting) Buffer Case

In this section, we consider a non-zero, i.e., one, waiting-buffer (system size=2 packets; one serving buffer plus one waiting buffer) or storage capacity in the forward path of Figure A-3. This additional buffer is incorporated to calculate the delay-related performance gain, later on this buffer is traded off with the ITC in the feedback path while maintaining the same performance gain. The resulting queueing network is as shown in Figure A-8 with the corresponding state transition diagram in Figure A-9.

![Diagram](image)

**Figure A-8**: A Non-Zero Buffer Case. We Consider One Additional Buffer -- Compared to that in Fig.A-3 -- in the Forward Path. Here, the System Size, K=2, i.e., Only One Waiting Buffer, and the Other is a Serving Buffer, for the Forward Path. And the system size, K=1, i.e., No Waiting Buffer, for the Feedback Path.

![Diagram](image)

**Figure A-9**: State Transition Diagram for the Queueing System of Fig.A-8.
With the same assumptions, notations, and packet looping principle as used in sub-section A.2.1, we find that the allowable system states, i.e., \((n_A, n_B)\) are \((0, 0), (0, 1), (1, 0), (1, 1), (2, 0)\) and \((2, 1)\). As before, we can write down a set of flow-balance equations for these states, and solving them, the limiting probabilities can be found.

\[
\begin{align*}
p_{01} &= \frac{\lambda^3}{\mu_1 \mu_2 (2\lambda + \mu_1 \mu_2)} p_{00} \\
p_{10} &= \frac{\lambda}{\mu_1} p_{00} \\
p_{11} &= \frac{\lambda^3 (\lambda + \mu_2)}{\mu_1^2 \mu_2 (2\lambda + \mu_1 + \mu_2)} p_{00} \\
p_{20} &= \frac{\lambda^2 (\lambda + \mu_1 + \mu_2)}{\mu_1^2 (2\lambda + \mu_1 + \mu_2)} p_{00} \\
p_{21} &= \frac{\lambda^3 (\lambda + \mu_2)(\lambda + \mu_1 + \mu_2) - \lambda \mu_1}{\mu_1^3 \mu_2 (2\lambda + \mu_1 + \mu_2)} p_{00}
\end{align*}
\]

where

\[
p_{00} = \frac{\mu_1^3 \mu_2 (2\lambda + \mu_1 + \mu_2)}{\mu_1^2 \mu_2 (\lambda + \mu_1)(2\lambda + \mu_1 + \mu_2) + \lambda^3 (\lambda + \mu_1 + \mu_2)^2 + \lambda^2 \mu_1 \mu_2 (\lambda + \mu_1 + \mu_2) - \lambda^4 \mu_1}.
\]
Let $S_f$ be the service completion time or the nodal forwarding time for a packet which is not blocked from the network segment. Then the mean system time $E(S_f)$ can be obtained as follows.

(a) Let $L$ be the amount of time to complete service when an arrival encounters the state $(2,0)$,

$$E(L) = P(B)E(L|B) + T_1 + T_2 = P(B)E(L|B) + T_1 + T_{20} + T_{22}$$

$$= \frac{\mu_2}{(\mu_1 + \mu_2)^2} + T_1 + T_{20}$$

$$+ \sum_{i=0}^{\infty} \frac{\mu_1 \mu_2 \lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right)^i \left( \frac{1}{\mu_1 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i$$

$$\times \left\{ \frac{2}{\mu_1 + \mu_2} + \frac{i + 1}{\mu_1 + \mu_2 + \lambda} \right\}$$

$$+ \sum_{i=1}^{\infty} \frac{\mu_1 \mu_2 \lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right)^i \sum_{j=0}^{i+1} \frac{c_{ij}}{(\mu_1 + \lambda)^j(\mu_1 + \mu_2)^{i+1-j}}$$

$$+ E(L) \left\{ \frac{\mu_2}{\mu_1 + \mu_2} + \right.$$

$$\left. \sum_{i=0}^{\infty} \frac{\mu_1 \mu_2 \lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right)^i \left( \frac{1}{\mu_1 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i \right\}$$

where $T_1$ and $T_{20}$ are defined in the proof.

(b) $E(S_1) = \frac{1}{1 - p_{21}} \left( P(A \text{ is empty}) \frac{1}{\mu_1} + P(A \text{ has 1 packet}) \frac{2}{\mu_1} + p_{20}E(L) \right)$

where,

$$P(a \text{ packet is blocked}) = p_{21}$$

$$P(A \text{ is empty}) = \frac{\lambda^3 + \mu_1 \mu_2(2\lambda + \mu_1 + \mu_2)}{\mu_1 \mu_2(2\lambda + \mu_1 + \mu_2)} p_{00}$$

$$P(A \text{ has 1 packet}) = \frac{\lambda^2 \mu_2 + \lambda \mu_1 \mu_2(2\lambda + \mu_1 + \mu_2)}{\mu_1 \mu_2^2(2\lambda + \mu_1 + \mu_2)} p_{00}$$
The proofs for the above expressions are given in section A.2.2.1. From these equations, we can solve for $E(L)$ and use the equation for $E(L)$ to obtain an expression for $E(S_I)$. We can then numerically investigate the effects of various parameters of interest, e.g., $\lambda$, $[\mu_2/\mu_1]$ etc. on $E(S_I)$ over a wide range of the parameters. Figure A-10 and Figure A-11 show the results of such investigations.

![Graph](image.png)

**Figure A-10:** Effect of Packet Arrival Rate, $\lambda$, on the Normalized Mean Nodal Forwarding Time, $E(S_I)$, for the Queueing System Presented in Fig.A-8.
Figure A-11: Effect of the Parameter, $a$, on the Normalized Mean Nodal Forwarding Time, $E(S_f)$, for the Queueing System Presented in Fig.A-8.
A.2.2.1 Determination of the Expression for $E(S_1)$

(a) Let $L$ be the amount of time to complete service when an arrival encounters the state (2,0),

$$E(L) = P(B)E(L|B) + T_1 + T_2 = P(B)E(L|B) + T_1 + T_{20} + T_{22}$$

$$= \frac{\mu_2}{(\mu_1 + \mu_2)^2} + T_1 + T_{20}$$

$$+ \sum_{i=0}^{\infty} \frac{\mu_1 \mu_2 \lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right)^i \left( \frac{1}{\mu_1 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i$$

$$\times \left\{ \frac{2}{\mu_1 + \mu_2} + \frac{i + 1}{\mu_1 + \mu_2 + \lambda} \right\}$$

$$+ \sum_{i=1}^{\infty} \frac{\mu_1 \mu_2 \lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right)^i \sum_{j=0}^{i+1} \frac{c_{ij}}{(\mu_1 + \lambda)^i(\mu_1 + \mu_2)^{i+1-j}}$$

$$+ E(L) \left\{ \frac{\mu_2}{\mu_1 + \mu_2} + \right.$$}

$$\left. \sum_{i=0}^{\infty} \frac{\mu_1 \mu_2 \lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right)^i \left( \frac{1}{\mu_1 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i \right\}$$

where $T_1$ and $T_{20}$ are defined in the proof.

(b) $E(S_1) = \frac{1}{1 - p_{21}} \left( P(A \text{ is empty}) \frac{1}{\mu_1} + P(A \text{ has 1 packet}) \frac{2}{\mu_1} + p_{20} E(L) \right)$

where,

$P(a \text{ packet is blocked}) = p_{21}$

$P(A \text{ is empty}) = \frac{\lambda^3 + \mu_1 \mu_2 (2\lambda + \mu_1 + \mu_2)}{\mu_1 \mu_2 (2\lambda + \mu_1 + \mu_2)} p_{00}$

$P(A \text{ has } i \text{ packet}) = \frac{\lambda^3 (\lambda + \mu_2) + \lambda \mu_1 \mu_2 (2\lambda + \mu_1 + \mu_2)}{\mu_1^2 \mu_2 (2\lambda + \mu_1 + \mu_2)} p_{00}$
PROOF: (a) Using the same assumptions and notations as in section A.2.1, we obtain the following expression for $E(L)$.

$$E(L) = \sum_i P(\text{event } i)E(L|\text{event } i)$$

$$= P(B)E(L|B)$$
$$+ P(AB)E(L|AB)$$
$$+ P(AAB)E(L|AAB)$$
$$+ P(ANB)E(L|ANB)$$
$$+ P(AANB)E(L|AANB)$$
$$+ P(ANAB)E(L|ANAB)$$
$$+ P(AANAB)E(L|ANANB)$$
$$+ P(AANNB)E(L|AANNB)$$
$$+ P(ANAANB)E(L|ANAANB)$$
$$+ P(ANANAB)E(L|ANANAB)$$
$$+ P(AANANAB)E(L|AANANAB)$$
$$+ P(AANANNB)E(L|AANANNB)$$
$$+ P(AANNAAB)E(L|AANNAAB)$$
$$+ P(AANANNB)E(L|AANANNB)$$
$$+ P(ANAANAB)E(L|ANAANAB)$$
$$+ P(ANAANNB)E(L|ANAANNB)$$
$$+ P(ANANAANB)E(L|ANANAANB)$$
$$+ P(ANANANB)E(L|ANANANB)$$
$$+ P(ANANANNB)E(L|ANANANNB)$$
$$+ P(AANANAAAAB)E(L|AANANAAB)$$
$$+ P(AANANANB)E(L|AANANAB)$$
$$+ P(ANOAB)E(L|ANOAB)$$
$$+ \ldots$$
Here

\[ P(B) = \frac{\mu_2}{\mu_1 + \mu_2} \]

\[ E(L|B) = \frac{1}{\mu_1 + \mu_2} + E(L) \]

\[ P(AB) = \frac{\mu_1}{\mu_1 + \mu_2} \cdot \frac{\mu_2}{(\lambda + \mu_1 + \mu_2)} \]

\[ E(L|AB) = \frac{1}{\mu_1 + \mu_2} + \frac{1}{(\lambda + \mu_1 + \mu_2)} + \frac{2}{\mu_1} \cdot \]

\[ P(AAB) = \frac{\mu_1}{\mu_1 + \mu_2} \cdot \frac{\mu_1}{(\lambda + \mu_1 + \mu_2)} \cdot \frac{\mu_2}{\lambda + \mu_2} \]

\[ E(L|AAB) = \frac{1}{\mu_1 + \mu_2} + \frac{1}{(\lambda + \mu_1 + \mu_2)} + \frac{1}{\lambda + \mu_2} + \frac{1}{\mu_1} \cdot \]

\[ P(ANB) = \frac{\mu_1}{\mu_1 + \mu_2} \cdot \frac{\lambda}{(\lambda + \mu_1 + \mu_2)} \cdot \frac{\mu_2}{\mu_1 + \mu_2} \]

\[ E(L|ANB) = \frac{1}{\mu_1 + \mu_2} + \frac{1}{(\lambda + \mu_1 + \mu_2)} + \frac{1}{\mu_1 + \mu_2} + E(L) \cdot \]

\[ P(AANB) = \frac{\mu_1}{\mu_1 + \mu_2} \cdot \frac{\mu_2}{(\lambda + \mu_1 + \mu_2)} \cdot \frac{\lambda \mu_1}{\lambda + \mu_1 + \mu_2} \cdot \frac{1}{\lambda + \mu_2} \]

\[ E(L|AANB) = \frac{1}{\mu_1 + \mu_2} + \frac{2}{(\lambda + \mu_1 + \mu_2)} + \frac{1}{\lambda + \mu_2} + \frac{2}{\mu_1} \cdot \]

\[ P(ANAB) = \frac{\mu_1}{\mu_1 + \mu_2} \cdot \frac{\mu_2}{(\lambda + \mu_1 + \mu_2)} \cdot \frac{\lambda \mu_1}{\mu_1 + \mu_2} \cdot \frac{1}{\mu_1 + \mu_2} \]

\[ E(L|ANAB) = \frac{1}{\mu_1 + \mu_2} + \frac{2}{(\lambda + \mu_1 + \mu_2)} + \frac{2}{\mu_1} + \frac{1}{\mu_1 + \mu_2} \cdot \]

and so on.

All sequences (for example AANAB) after the first four in the expression for \( E(L) \) begin with AAN or AN. Let \( l_{ij} \) be the \( j \)-th sequence of length \( i \). We treat sequences of odd and even length separately. Then:

\[ E(L) = \sum_{i,j} P(\text{sequence } l_{ij})E(L|\text{sequence } l_{ij}) \]

\[ = P(B)E(L|B) + \sum_{i=1}^{\infty} \sum_{j} P(l_{2i,j})E(L|l_{2i,j}) \]

\[ + \sum_{i=1}^{\infty} \sum_{j} P(l_{2i+1,j})E(L|l_{2i+1,j}) \]

\[ := P(B)E(L|B) + T_1 + T_2 \cdot \]
where ':=' means 'is defined as'. We simplify our expressions for $T_1$ and $T_2$.

\[
T_1 = \left( \frac{\mu_1}{\mu_1 + \mu_2} \right) \left( \frac{\mu_2}{\mu_1 + \mu_2 + \lambda} \right) \left( \frac{1}{\mu_1 + \mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} + \frac{2}{\mu_1} \right) \\
+ \left( \frac{\mu_1}{\mu_1 + \mu_2} \right) \left( \frac{\mu_2}{\mu_1 + \mu_2 + \lambda} \right) \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right) \left( \frac{1}{\mu_1 + \mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right) \\
\times \left( \frac{1}{\mu_1 + \mu_2 + \lambda} + \frac{2}{\mu_1 + \mu_2} + \frac{2}{\mu_1} \right) \\
+ \left( \frac{\mu_1}{\mu_1 + \mu_2} \right) \left( \frac{\mu_2}{\mu_1 + \mu_2 + \lambda} \right) \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right) \left( \frac{1}{(\mu_2 + \lambda)^2} + \frac{1}{(\mu_1 + \mu_2)^2} \right) \\
+ \ldots.
\]

So

\[
T_1 = \sum_{i=0}^{\infty} \left( \frac{\mu_1}{\mu_1 + \mu_2} \right) \left( \frac{\mu_2}{\mu_1 + \mu_2 + \lambda} \right) \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right)^i \left( \frac{1}{\mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i \\
\times \left\{ \frac{1}{\mu_1 + \mu_2} + \frac{i + 1}{\mu_1 + \mu_2 + \lambda} + \frac{2}{\mu_1} \right\} \\
+ \sum_{i=1}^{\infty} \left( \frac{\mu_1}{\mu_1 + \mu_2} \right) \left( \frac{\mu_2}{\mu_1 + \mu_2 + \lambda} \right) \left( \frac{\lambda \mu_1}{\mu_1 + \mu_2 + \lambda} \right)^i \frac{i + 1}{\sum_{j=0}^{i} \frac{c_{ij}}{(\mu_2 + \lambda)^j(\mu_1 + \mu_2)^{i + 1 - j}}.
\]

where $c_{10} = 1$, $c_{11} = 0$, $c_{12} = 1$, $c_{20} = 2$, $c_{21} = 2$, $c_{22} = 2$, $c_{23} = 2$, $\ldots$.

In general, the expression for $c_{ij}$ can be shown to be

\[
c_{ij} = \begin{cases} 
\binom{i}{j}(i - j) + \binom{i}{j-1} & \text{for } j \leq i/2 \\
c_{i,i+1-j} & \text{for } j > (i + 1)/2 \\
2\binom{i}{j-1}(j-1) & \text{for } j = (i + 1)/2 \text{ and } j = 0, \ldots, i + 1.
\end{cases}
\]

Next we examine the summand $T_2$. To compute $T_2$, it must be divided into two parts depending on whether the sequences $l_{ij}$ end with zero customers or two customers in front of the server $A$ just before the completion by server $B$. The case with zero customers will give an expression similar to $T_1$, while the case with two customers will give an expression involving $E(L)$. We denote the sum involving sequences ending with zero customers by $T_{20}$ and the sum involving sequences ending with two customers by $T_{22}$. 
\[ T_{20} = P(AAB)E(L|AAB) + P(AANAB)E(L|AANAB) + P(ANAAAB)E(L|ANAAAB) + \ldots \]
\[ = \sum_{i=0}^{\infty} \frac{(\mu_1^2\mu_2)^i}{(\mu_1 + \mu_2)((\mu_1 + \mu_2 + \lambda)(\mu_2 + \lambda))} \left( \frac{\lambda\mu_2}{\mu_1 + \mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i \left( \frac{1}{\mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right) \]
\[ \times \left\{ \frac{1}{\mu_1 + \mu_2} + \frac{i + 1}{\mu_1 + \mu_2 + \lambda} + \frac{1}{\mu_2 + \lambda} + \frac{1}{\mu_1} \right\} \]
\[ + \sum_{i=0}^{\infty} \frac{(\mu_1^2\mu_2)^i}{(\mu_1 + \mu_2)((\mu_1 + \mu_2 + \lambda)(\mu_2 + \lambda))} \left( \frac{\lambda\mu_1}{\mu_1 + \mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i \sum_{j=0}^{i+1} \frac{c_{ij}}{(\mu_1 + \lambda)i(\mu_1 + \mu_2)^{i+1-j}} \]

Similarly

\[ T_{22} = P(ANB)E(L|ANB) + P(AANNB)E(L|AANNB) + P(ANANB)E(L|ANANB) + \ldots \]
\[ = \frac{\mu_1\mu_2\lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left\{ \frac{2}{\mu_1 + \mu_2} + \frac{1}{\mu_1 + \mu_2 + \lambda} + E(L) \right\} \]
\[ + \frac{\mu_1^2\mu_2^2}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)^2(\mu_2 + \lambda)} \left\{ \frac{2}{\mu_1 + \mu_2} + \frac{2}{\mu_1 + \mu_2 + \lambda} + E(L) + \frac{1}{\mu_2 + \lambda} \right\} \ldots \]
\[ = \sum_{i=0}^{\infty} \frac{\mu_1\mu_2\lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda\mu_1}{\mu_1 + \mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i \]
\[ \times \left\{ \frac{2}{\mu_1 + \mu_2} + \frac{i + 1}{\mu_1 + \mu_2 + \lambda} + E(L) \right\} \]
\[ + \sum_{i=0}^{\infty} \frac{(\mu_1\mu_2\lambda)}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda\mu_1}{\mu_1 + \mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i \sum_{j=0}^{i+1} \frac{c_{ij}}{(\mu_1 + \lambda)i(\mu_1 + \mu_2)^{i+1-j}} \]

Combining our results, and collecting terms gives

\[ E(L) = P(B)E(L|B) + T_1 + T_2 = P(B)E(L|B) + T_1 + T_{20} + T_{22} \]
\[ = \frac{\mu_2}{(\mu_1 + \mu_2)^2} + T_1 + T_{20} \]
\[ + \sum_{i=0}^{\infty} \frac{\mu_1\mu_2\lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda\mu_1}{\mu_1 + \mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i \]
\[ \times \left\{ \frac{2}{\mu_1 + \mu_2} + \frac{i + 1}{\mu_1 + \mu_2 + \lambda} \right\} \]
\[ + \sum_{i=0}^{\infty} \frac{\mu_1\mu_2\lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda\mu_1}{\mu_1 + \mu_2 + \lambda} \right)^i \sum_{j=0}^{i+1} \frac{c_{ij}}{(\mu_1 + \lambda)i(\mu_1 + \mu_2)^{i+1-j}} \]
\[ + E(L) \left\{ \frac{\mu_2}{(\mu_1 + \mu_2)^2} + \sum_{i=0}^{\infty} \frac{\mu_1\mu_2\lambda}{(\mu_1 + \mu_2)^2(\mu_1 + \mu_2 + \lambda)} \left( \frac{\lambda\mu_1}{\mu_1 + \mu_2 + \lambda} + \frac{1}{\mu_1 + \mu_2} \right)^i \right\} \]

(b) This result is clear.
A.3 Results

This section is concerned with the 4-th, 5-th, and 6-th steps of the procedure presented in section A.1. Here, we demonstrate the proposed SC-TC tradeoff by using the relationships developed in the previous sections. Our main objective is to trade the nodal feedforward path buffers, i.e., the SC with the ITC in the feedback path. We keep the mean nodal feed forwarding time the same as that of the system with a pre-specified number of buffers. To achieve this, we must develop explicit formulations for $E(S_0)$ and $E(S_f)$ as a function of the parameter $a (= \mu_2/\mu_1)$ over a wide range of the packet arrival rate, $\lambda$ (e.g., 0.00 to 0.90). Note that, without loss of generality, we can assume $\mu_1 = 1.0$, therefore, $a = \mu_2$ -- for the number of buffers under consideration -- and $\lambda$ is directly proportional to the server utilization.

Figure A-6 shows that for low values of $a$, the system tends to be unstable even at low load, e.g., at $a = 0.05$, the maximum allowable load is $\lambda=0.25$ packets/slot. For higher values of $a$, however, the mean system time versus offered load curve shows overdamped - and hence stable - type behavior. The mean nodal forwarding time smoothly reaches a steady state value for $a \geq 0.15$. Similar characteristics can also be found in Figure A-10 for the system with one buffer in the forward path. Now, since with an increase in $a$, the packets blocked in the feedback path get the chance to visit the server in the forward path more frequently, the mean system time decreases when $a$ increases. This feature is very well depicted in both Figure A-7 and Figure A-11.

From Figure A-7 and Figure A-11, we can develop simple analytical relationships between $E(S_0)$ and $a$, and between $E(S_f)$ and $a$, respectively. Using these relationships, we can investigate the variations of the ratio of the parameter $a$ (for zero buffer) to the
parameter $a$ (for one buffer) against the load $\lambda$. $E(S_1)$ is determined for a specific value of the parameter $a$ over a wide range of the offered load $\lambda$.

The resulting tradeoff curves are presented in Figure A-12 which shows how much ITC is needed at a particular load in the feedback path of Figure A-3 so that it will offer the same amount of mean nodal forwarding time as that of the system presented in Figure A-8 at that particular load. If it is intended to keep the probability of packet blocking as low as possible, we need to keep the parameter $a$ larger than or equal to unity. Otherwise, any sub-unity value for $a$ would suffice the requirement of using the network itself as effective buffer.

The impact of the load $\lambda$ on ITC -- as shown in Figure A-12 -- is as expected intuitively. As the load increases, a packet which is blocked in the feedback path must make attempts more often to increase its chance to find an idle server in the forward path. Thus, the number of attempts must increase (i.e., the speed of the channel must increase) as the buffer size in the forward path decreases.

Similar performance evaluation studies can be conducted for trading higher number (i.e., 2, 3, or 4) of buffers, i.e., SC with the ITC of the reverse or feedback path. This type of studies can be regarded as a natural extension of the investigations presented herein.

A.4 Remarks

The main objective of this work is to show that the nodal storage capacity (SC), i.e., the buffers can be traded with the Incremental Transmission Capacity (ITC) of the links in the feedback or reverse path. We have shown these results in Figure A-12.
Figure A-12: Demonstration of the Proposed SC-TC Trade-off. We consider the design of a fabric (or network) segment with zero buffer whose delay related performance is the same as that of the non-zero buffer system considered in Fig.A-8. Note that the ordinate represents the incremental TC needed in the feedback path to eliminate the additional buffer in the forward path of Fig.A-8.

It is expected that the results will be useful, especially, for designing all-optical fast packet switching (FPS) networks at both micro- and macro-level where the designers have enormous transmission capacity, i.e., bandwidth-on-hand. This transmission capacity can now be traded with other desirable -- but neither technologically available nor practically feasible -- network characteristics. Finally, the procedure presented in this paper can be easily extended to incorporate non-uniform feedback path in multi-input and multi-output nodal switching fabrics considered in [63] and [64].
A.5 Embedding MSNs Onto Ring Networks

In this section we present graphical embedding of the Manhattan Street Network (MSN) onto a particular hard-topology, the Ring network. Note that, a 2x2 MSN is the same as that of a two-channel ring network [136], as shown in Figure A-13 where one channel is carrying message packets in clock-wise direction, and the other in counter-clock-wise direction.

![Figure A-13: Embedding a 2x2 Manhattan Street Network (MSN) on-to a 4-Node Ring Network.](image)

Next, we show the embedding of a 4x4 MSN (Figure 5.2) in a 16-node multi-channel ring network (Figure A-14), where packets are flowing in each direction (clock-wise and anti-clock-wise) via three channels. The logical spare channels between the nodes 3-4, 7-8, 11-12, and 15-0 circulate around the ring as the soft-topology is reconfigured under the influence of the demanded traffic matrix and pattern.
Figure A-14: Embedding a 4x4 Manhattan Street Network (MSN) On-to a 16-Node, 6-Channel Ring Network.

Finally, note that the types of embedding we are considering are completely feasible [23, 43] using the Wave-length Division Multiplexing (WDM) technique in the optical fiber networks.
Appendix - B

DETERMINATION OF THE EXACT WAITING TIME FOR AN M/D/1 QUEUEING SYSTEM WITH FINITE (≤ 4) WAITING ROOM

For an M/D/1/K queue, the packet arrival process is Poisson with rate, e.g., \( \lambda \) packets/slot, the service time, \( 1/\mu \), is constant, e.g., one slot of time, and therefore, \( \rho = (\lambda/\mu) = \lambda \). The size of the waiting buffer, \( b = (K-1) \), and since the packet arrival process is Poisson, its distribution is

\[
a(n) = \frac{\lambda^n e^{-\lambda}}{n!}.
\]

Therefore, \( a(0) = e^{-\lambda} \), \( a(1) = \lambda e^{-\lambda} \), \( a(2) = \frac{\lambda^2 e^{-\lambda}}{2!} \), and so on.

Now, let \( P_n \) specify the probability of having exactly \( n \) packets in the system. The average waiting time in queue can now be determined for various values of \( \lambda \) e.g., from 0.0 to 0.95, using the following relationship (derived by using Little's formula \([46, 86, 110, 129, 142]\)).

\[
W_Q = \frac{1}{\lambda(1 - P_K)} \left( \sum_{n=0}^{K} [n \cdot P_n] - (1 - P_0) \right)
\]  

(B.1)

The values of \( P_n \) are now computed using the technique presented in \([32, 125, 129]\).

B.1 The Zero Buffer Case

For \( b = 0 \), i.e., \( K = (0+1) = 1 \), we have the following balance equations and normalization condition.
\[ P_0 = [(P_0 + P_1) \cdot a(0)] \]
\[ P_0 + P_1 = 1.0 \]

Solving the above two equations, we have \( P_0 = a(0) \) and \( P_1 = P_0 \left( \frac{1}{a(0)} - 1 \right) \). Therefore, we can now study the variation of \( W_Q \) with \( \lambda \) for \( b=0 \) using equation (B.1).

### B.2 The Two Buffer Case

For \( b=2 \), i.e., \( K=2+1=3 \), we have the following balance equations:

\[ P_0 = (P_0 + P_1) \cdot a(0) \]
\[ P_1 = (P_0 + P_1) \cdot a(1) + P_2 \cdot a(0) \]
\[ P_2 = (P_0 + P_1) \cdot a(2) + P_2 \cdot a(1) + P_3 \cdot a(0) \]

The boundary condition in this case is: \( \sum_{n=0}^{3} [P_n] = 1 \). Solving these equations, we obtain:

\[ P_0 = \frac{a(0)^3}{(1 - a(1))^2 - a(0)a(2)} \]
\[ P_1 = P_0 \left( \frac{1}{a(0)} - 1 \right) \]
\[ P_2 = P_0 \left( \frac{1 - a(0) - a(1)}{a(0)^2} \right) \]
\[ P_3 = P_0 \left( \frac{1 - a(1))(1 - a(0) - a(1))}{a(0)^3} \right) - a(0)a(2) \]
Given the values of \( P_0 \) through \( P_3 \), we can determine \( W_Q \) for various values of \( \lambda \) for \( K=3 \) using equation (B.1).

### B.3 The Four Buffer Case

For \( b=4 \), \( K=(4+1)=5 \), we have the following balance equations:

\[
\begin{align*}
P_0 &= (P_0 + P_1) a(0) \\
P_1 &= (P_0 + P_1) a(1) + P_2 a(0) \\
P_2 &= (P_0 + P_1) a(2) + P_2 a(1) + P_3 a(0) \\
P_3 &= (P_0 + P_1) a(3) + P_2 a(2) + P_3 a(1) + P_4 a(0) \\
P_4 &= (P_0 + P_1) a(4) + P_2 a(3) + P_3 a(2) + P_4 a(1) + P_5 a(0)
\end{align*}
\]

The boundary condition in this case is: \( \sum_{n=0}^{5} [P_n] = 1 \). Solving these equations, we obtain:

\[
P_0 = \frac{a(0)^5}{\text{Denominator}}
\]

where \( \text{Denominator} = (1 - a(1)) [a(3) a(0)^3 - 2 a(3) a(0)^2 - a(0)^2 a(2)^2] \\
+ a(0) (1 - a(1))^2 [1 - a(1) - 3 a(2)] \\
+ (1 - a(1))^3 [1 - a(0) - a(1)] \\
- a(0)^3 [a(3) + a(4)]
\]

\[
P_1 = P_0 \left( \frac{1}{a(0)} - 1 \right)
\]
\[ P_2 = P_0 \frac{(1 - a(0) - a(1))}{a(0)^2} \]

\[ P_3 = P_0 \frac{(1 - a(1))(1 - a(0) - a(1)) - a(0)a(2)}{a(0)^3} \]

\[ P_4 = P_0 \frac{(1 - a(1))^2[1 - a(0) - a(1)] - 2a(0)a(2)(1 - a(1)) + a(0)^2[a(2) - a(3)]}{a(0)^4} \]

\[ P_5 = 1 - \sum_{n=0}^{4} \frac{P_n}{a(0)^n} \text{Numerator.} \]

where, Numerator = [1 - a(1)] [(1 - a(1))^2 - a(0)a(2)] [1 - a(0) - a(1)]

\[ + a(0)^2 a(3) [1 - a(0) - a(1) - a(2)^2] \]

\[ + a(0)^2 (1 - a(1)) [a(2) - a(3)] \]

\[ - 2 a(0) a(2) (1 - a(1))^2 \]

\[ + a(0)^3 a(4). \]

Now, we can study the effect of \( \lambda \) on \( W_Q \) for the case when \( K=5 \) using equation (B.1).

Since in our investigation we need to consider a maximum of four buffers at the input of each outgoing link [45, 104], we do not continue the development of analytical formulations for buffer sizes > 4 packets.
Appendix-C

SWITCHING PATTERNS OF A 3X3 SWITCH

In this section we derive analytical expressions for all possible Switching Patterns (SPs) or Connection Configurations (CCs) of an n-input-m-output switch. Helpful references for this type of developments are [12] and [65]. Pictorial representations of the feasible SPs of a 3x3 switch (Figure C-1) are as shown in Figure C-2 through Figure C-4.

![Diagram of a 3x3 switch](image)

Figure C-1: An nxm Switch, where for a 3x3 switch, n=3, and m=3.

For an nxm switch, i.e., the switch has n input terminals and m output terminals, when only one input terminal is busy, there are \( \binom{m}{1} \) different ways that this input terminal can be connected to an output terminal. But, since the input terminal can be any one of the n inputs, the total number of 1-in-1-out CCs is 'm times n', which is \( n^2 \) for an nxn switch. Again, since the \( L_1 \) terminal does not send any information to the \( L_0 \) terminal (see Figure 5.4), the total number of feasible 1-in-1-out CCs are:

\[
\left[ mn - 1 \right]
\]  \hspace{1cm} (C.1)
or, \((n^2 - 1)\) iff \(m=n\).

Next, if two of the \(n\) incoming terminals are busy, then there are \(\binom{n}{2}\) different ways to choose two input terminals. For each of these choices, there are \(\binom{m}{2}\) different ways to select two possible output terminals. These two input terminals can be connected to the two output terminals in either straight (=) or exchange (X) fashion. Thus, the total number of CCs become: \([2p_2 \times \binom{n}{2} \times \binom{m}{2}]\). Here again, the interconnection between \(L_i\) and \(L_o\) is not desirable, and the fraction of CCs in which the \(L_i\) appears to be connected to the \(L_o\) is:

\[
2 \times \frac{1}{2n} \times \frac{\binom{m}{2}}{\binom{m}{2} - \binom{m-1}{2}} = \frac{2}{m \times n}
\]

Consequently, the total number of feasible 2-in-2-out CCs are:

\[
\left[ 2 \times \binom{n}{2} \times \binom{m}{2} \right] \times \left[ 1 - \frac{2}{m \times n} \right]
\]

Similarly, when 3 out of \(n\) input terminals have packets to transfer, there are \(\binom{n}{3}\) different ways to choose 3 inputs and \(\binom{m}{3}\) different ways to select the output terminals.

Once, one of the three inputs is connected to any one of the \(m\) outputs (\(m\) different choices), it is required to choose 2 output links out of (\(m-1\)), i.e., \(\binom{m-1}{2}\) different choices, and for each of these choices, there can be two different configurations (straight and exchange). Thus the total number of CCs are:

\[
\left[ \binom{n}{3} \times m \times \binom{m-1}{2} \times 2p_2 \right]
\]
Now, if \( L_1 \) is one of the input terminals whose packet transfer request is to be fulfilled, then since \( L_1-L_0 \) connection pattern is prohibited, we end up with the following number of feasible 3-in-3-out CCs:

\[
\left[ 2 \times \left( \binom{n}{3} \times m \times \left( \binom{m-1}{2} \right) \right) \times \left( 1 - \frac{m}{m \times n} \right) \right]
\]  

(C.3)

Therefore, for a 3x3 switch, the total number of feasible 1-in-1-out CCs is 8, 2-in-2-out CCs is 14, and 3-in-3-out connections is 4, as can be easily computed using equations (C.1) through (C.3). Thus, the total number of feasible CCs for a 3x3 switch is \((8+14+4)=26\). Now, if a virtual outgoing link is added to the switch, as shown in Figure 5.4, the total number of feasible CCs increases to \((11+30+16)=57\). This increases the number of feasible connection patterns by more than 200%. It substantially reduces the probability of network congestion not only by reducing the probability of packet dropping but also by policing the network access.

![Diagram of 1-in-1-out SPs]

**Figure C-2:** All Possible 1x1 Switching Patterns (SPs) of a 3x3 Switch.
2-in-2-out SPs

Figure C-3: All Possible 2x2 Switching Patterns (SPs) of a 3x3 Switch.
Figure C-4: All Possible 3x3 Switching Patterns (SPs) of a 3x3 Switch.
Appendix-D

LOCAL BUFFER STEALER

The local buffer stealer (Figure D-1) is a very simple priority arbiter. It allows the transmit buffer of the local source to be temporarily utilized by the transit packets under the following conditions:

• Both the incoming transit packets are only two hops away from the destination,
• Both of them satisfy all the criteria to get routed through the same logical outgoing link, and
• None of them is destined to the local sink.

The motivations for incorporating the local buffer stealer at the nodes are: (i) the size of the local source's transmit buffer is usually kept higher than that at the input of the outgoing link of a node, and (ii) in supra-high-speed packet switching networks the size of a time slot is very small (of the order of micro-seconds or lower) and hence, it may be possible to steal few slots equivalent of local transmit buffers' occupancy either without user's notice or without causing much of an annoyance to him/her. Our simulation results supports these statements [83].

The priority arbiter can be a simple maximum selection network (described in Appendix-E) and the transit packet is always given higher priority over the local sources' packets. Note that under heavy load conditions, the transit packets can be allowed to obtain exclusive possession of the local source's transmit buffer using a threshold function of the type shown in Figure 5.9.
Thus, the buffer stealer implicitly enforces the throttling of local packets’ access to the network. This mechanism also partially helps to avoid network congestion.

Figure D-1: The Buffer Stealer.
Appendix-E

MAXIMUM SELECTION, MINIMUM SELECTION, AND SORTING USING ARTIFICIAL NEURAL NETWORKS (ANNs)

In this section, we present a detailed description of maximum selection, minimum selection and sorting using Artificial Neural Networks (ANNs).

E.1 Maximum Selection Network

The architecture of the maximum selection network along with the threshold functions which are controlling the operation of the neurons is as shown in Figure E-1.

The operation of this network can be described as follows. Let $x_0$ and $x_1$ are two inputs. Now if $x_0 > x_1$ then:

$$v_0 = \Phi_{\text{tp}}(x_0 - x_1) = \begin{cases} [x_0 - x_1] & \text{if } (x_0 - x_1) \leq \vartheta \\ \vartheta & \text{if } (x_0 - x_1) \geq \vartheta \end{cases}$$

and $v_1 = \Phi_{\text{tp}}(x_1 - x_0) = 0.0$.

Therefore, the output of the network is:
Max(\(x_0, x_1\)) = \(\Phi_{tt}(0.5 (v_0 + x_0 + x_1 + v_1))\)

\[= \Phi_{tt}(x_0) = \begin{cases} x_0 & \text{if } |x_0| \leq |\theta| \\ \pm \theta & \text{if } |x_0| \geq |\theta| \end{cases} \]

Thus, the network of Figure E-1 selects the maximum of two inputs.

Figure E-1: The Maximum Selection Network.
E.2 Minimum Selection Network

The architecture of the minimum selection network is exactly same as that of the maximum selection network, except that here the threshold function used to generate the intermediate outputs, i.e., $v_0$ and $v_1$, is different as shown in Figure E-2.

The operation of this network can be described as follows. Let $x_0$ and $x_1$ are two inputs. Now if $x_0 > x_1$ then:

$$v_1 = \Phi_{tn}(x_1 - x_0) = \begin{cases} \lceil x_1 - x_0 \rceil & \text{if } (x_1 - x_0) \geq -\theta \\ -\theta & \text{if } (x_1 - x_0) \leq -\theta \end{cases}$$

and $v_0 = \Phi_{tn}(x_0 - x_1) = 0.0$.

Therefore, the output of the network is:

$$\text{Min}(x_0, x_1) = \Phi_{tt}(0.5 (v_0 + x_0 + x_1 + v_1))$$

$$= \Phi_{tt}(x_1) = \begin{cases} x_1 & \text{if } |x_1| \leq |\theta| \\ \pm \theta & \text{if } |x_1| > |\theta| \end{cases}$$

Thus, the network of Figure E-2 selects the minimum of two inputs.
Figure E-2: The Minimum Selection Network.
E.3a Sorting Network (Type-1)

In the type-1 sorting networks, the basic building block consists of a maximum selection and minimum selection networks as described in the previous two sections. The basic building block along with a four input sorter is shown in Figure E-3a.

Note that, since $\binom{4}{2} = 6$, we need at most six 2-input-2-output network (basic sorter) to construct a four input sorting network.

![Diagram of Sorting Network](image)

**Figure E-3a: Type-1 Sorter Using Artificial Neural Networks.**
E.3b Sorting Network (Type-2)

In this type of sorter, the outputs of the neurons in the intermediate layer are used to compute the weights of the synapses from the inputs to the neurons in the highest layer whose output is either maximum or minimum of the two inputs. The architecture of the sorter along with the threshold functions of each layer are as shown in Figure E-3b (i and ii).

![Diagram](image)

**Figure E-3bi:** Threshold Functions for Sorting (Type-2) Using Artificial Neural Networks.
Figure E-3bii: Sorting (Type-2) Using Artificial Neural Networks.
The operation of this network is as follows. Let the two inputs are \( x_0 \) and \( x_1 \), then if \( [x_0 > x_1] \), \( v_{L1} = 1 \) and \( v_{R1} = 0 \), therefore,

\[
Z_R = \Phi_{tt}(x_0 \cdot v_{L1} + x_1 \cdot v_{R1}) = \begin{cases} x_0 & \text{if } x_0 \in \{0\} \\ \emptyset & \text{if } x_0 \notin \{0\} \end{cases} = \text{Max}(x_0, x_1)
\]

Similarly, since \( v_{L2} = 0 \) and \( v_{R2} = 1 \), therefore,

\[
Z_L = \Phi_{tt}(x_0 \cdot v_{L2} + x_1 \cdot v_{R2}) = \begin{cases} x_1 & \text{if } x_1 \in \{0\} \\ \emptyset & \text{if } x_1 \notin \{0\} \end{cases} = \text{Min}(x_0, x_1)
\]

Consequently, the network of Figure E-3b can be used as a sorter. For multi-input sorting, we can use the network of Figure E-3b as the basic building block to come up with a structure similar to that of the bottom half of Figure E-3a.
Appendix-F

LEARNING IN THE ADAPTIVE RESONANCE THEORY (ART) BASED ARTIFICIAL NEURAL NETWORKS (ANNs)

In this appendix, we present a very brief overview of the type-1 ART network, i.e., ART-1, along with the basic equations which govern learning in this type of networks. For details, the interested readers may consult Reference [52].

F.1 Description of the ART-1 Machines

ART-1 dictates a two-layered (Comparison layer or C-Layer and Recognition layer or R-Layer), deterministic or quasi-stochastic, self-controlled, self-scaled, self-organized and self-stabilized massively parallel ANN architecture. This network can recognize codes (for pattern recognition and image processing applications) in response to arbitrarily large number of arbitrarily complex input patterns presented in arbitrary sequence.

ART-1 machines are capable of extracting the critical feature patterns from the environment (environment is the teacher for the ART machines) using, (i) Attentional Priming, (ii) Automatic Gain Control (AGC), (iii) Attentional Vigilance, and (iv) Intermodal Competition. Of these, the Attentional Priming and AGC contribute towards code matching and stabilization activities of the ART-1 nets. When the R-layer is active, the Attentional Priming mechanism delivers excitation-specific learned template pattern to the
C-layer. The AGC is an auxiliary mechanism required to distinguish between Top-Down (TD) and Bottom-Up (BU) inputs. It has an inhibitory non-specific unlearned effect on the sensitivity with which the C-layer responds to the top-down template pattern as well the other patterns received by the lower layer. Attentional Vigilance defines the perfect matching range and Intermodal Competition helps in selecting the necessary input features competitively.

Consequently, the ART-1 nets are capable of circumventing the effects of noise, saturation, capacity limits, the input orthogonality requirements and the linear predictability constraints of the pattern recognition mechanisms or models.

F.2 Major Properties of the ART-1 Machines

The following are the major properties of the ART-1 based ANNs.

(i) **Plasticity**: It means that the new critical features are allowed to overlap the old ones when it (the new one) overmatches the stored pattern (connection configuration or switching pattern).

(ii) **Stability**: New steady-states (or patterns or classes or categories) are dynamically created to match the input from the environment.

(iii) **Stability-Plasticity Dilemma**: All the familiar patterns (or events) are processed as usual (i.e., as in ordinary stored-pattern matching based decision), and for an unfamiliar event new classes are generated.
(iv) **Role of Attention in Learning:** The activities are as described by Attentional Priming, AGC, Attentional Vigilance and Intermodal Competition in the previous section.

(v) **Complexity:** ART-1 dynamically recognizes the input codes (or patterns) to preserve its stability-plasticity balance as its internal representation become increasingly complex and discriminated through learning, i.e., ART-1 nets always maintain an acceptable level of balance among plasticity, stability, and complexity.

Thus, we see that the ART-1 nets are:

- **Self-scaling,** i.e., their capability to withstand mismatches with the stored (i.e., learned) patterns increases with the increase of the degree of mismatch.
- **Self-stabilizing,** i.e., the machine does not go through infinite number of iterations to find a match with the stored patterns. It rather triggers off the search process and creates a new stored-pattern when the degree of mismatch exceeds a preset threshold at a prescribed vigilance level.
- **Capable of circumventing noise, saturations, and the constraints like capacity limit, orthogonality and linear predictability etc. of the traditional pattern recognition mechanisms.**

These features of the ART-1 machine motivate Grossberg [52] to claim that this ANN architecture is the ultimate massively parallel structure for pattern recognition applications.

**F.3 Operational Features of the ART-1 Machines**

The following are the four basic working properties of the ART-1 nets:

(i) **Self-Scaling Computation:** This means that the ART nets use context- and learning-history dependent definitions of signals and noise.
(ii) Self-Adjusting Memory Search: The knowledge structure evolves due to the learning in an environment-specific fashion. Usually it starts with a little randomness in the initial values. This is the most useful attribute of the ART-1 machine which makes it feasible for efficient operation in uncertain environments.

(iii) Direct Access to Learned Codes: Learned codes become globally self-consistent and predictably accurate. Therefore, the search process becomes automatically triggered off when a familiar input pattern is presented to the ART-1 machines. Only the unfamiliar patterns can create new classes of patterns in the R-layer.

(iv) Attentional Vigilance (Environment as a Teacher): In the ART-1 nets the environment mediates the learning process and thereby carry out the role of a teacher. Although the number of feature detectors in the ART-1 nets is fixed, it seems that the capability of the detectors can be enhanced via teaching by modulating the vigilance.

(v) Matching (the 2/3 rule): The 2/3 rule states that to activate (i.e., to generate supra-threshold output signal) a neuron in the lower-layer (i.e., the feature detection and extraction layer or C-Layer), the two of its three input signal sources must be activated.

Other operational features of the ART-1 nets include the following:

(a) Bottom-Up (BU) Adaptive Filtering and Contrast Enhancement in the Short Term Memory (STM) System

The neurons in the C-layer receive the input \( I \) from the environment, extract the necessary set of features which is a kind of STM, produce an output signal pattern \( S \) using the 2/3 rule. Finally, the C-layer sends an environmental expectation, i.e., a weighted
sum of the components of \( S \) or Long Term Memory (LTM)-gated signal to each of the target nodes (a pattern) of the upper layer (or R-layer).

The received vector, \( I \) in the R-layer is transformed to the vector \( Y \) via interactions, i.e., extensive lateral inhibitions as in Hamming Net [52, 117, 128], to select the winner neuron. This contrast-enhanced pattern, \( Y \) is stored in the STM by the R-layer nodes. Note that the LTM weights in the pathways, i.e., the connection weights or synaptic weights, do not change during the recognition event.

(b) Top-Down (TD) Template Matching And Stabilization of Code Learning

The STM activity pattern, \( Y \) in the R-layer produces a top-down signal pattern, \( U \). The components of \( U \) are LTM-gated and added to produce the input vector \( Y \), i.e., the top-down template or learned expectation to the C-layer neurons. The original STM activity pattern \( \Delta \) is now updated to \( \Delta^* \) by using the combined effect of the input pattern \( I \) and the signal vector \( Y \). Note that \( \Delta \) was produced by \( I \) alone, because the output of the gain control unit was unity at that time. Also, the LTM weights do not change during the recognition event.

(c) Interaction Between Attentional and Oriental Systems (STM Reset and Search)

If the top-down template, i.e., the learned expectation \( Y \), differs form the input pattern \( I \) beyond a pre-specified threshold, \( \Delta \) will be changed to \( \Delta^* \). This change initiates a vigilance (a kind of threshold) dependent inhibitory signal towards the R-layer, and thereby resets all the activated neurons in that layer. This is done by the reset block of the ART-1
net. Thus, the STM pattern $\mathbf{V}$ is removed from the R-layer, and the original STM activity pattern $\mathbf{A}$ is re-instated in the C-layer. The cycle $\mathbf{I} \rightarrow \mathbf{A} \rightarrow \mathbf{S} \rightarrow \mathbf{T} \rightarrow \mathbf{Y}^* \rightarrow \mathbf{U}^* \rightarrow \mathbf{V}^*$ continues.

Thus, a series of STM matching and reset events occur and the search for a stable LTM configuration (a spatial configuration which is the desired connection configuration) continues. This process sequentially activates all the relevant nodes (i.e., patterns) in the R-layer by using the novelty-sensitive orienting sub-system. A new recognition category, i.e., a new class or pattern, is created when the search process fails to find a near-match (as dictated by the vigilance parameter) stored (or learned) pattern.

F.4 Learning in the ART-1 Machines

Learning or tuning of the synaptic weights (or connection strengths) occur in the ART-1 net according to the following membrane equations [52]. A conscious reader may easily visualize the resemblance between these equations and the birth-death process equations of queueing theory [86].

The STM activity $\lambda_i$ of a neuron $v_i$ in the C-layer is represented by:

$$
\epsilon \frac{d\lambda_i}{dt} = -\lambda_i + (1 - A_1 \lambda_i) [I_i + D_1 \sum_j f(\mu_j) z_{ij}] - (B_1 + C_1 \lambda_i) [\sum_j f(\mu_j)] \quad (F.1)
$$

where the range of i and j are as shown in Figure F-1. Note that in order to simplify the notations we write $\lambda_i$, $\mu_j$, $I_i$, $I_j$, $z_{ij}$, $z_{ji}$ to mean $\lambda_i(t)$, $\mu_j(t)$, $I_i(t)$, $I_j(t)$, $z_{ij}(t)$, $z_{ji}(t)$, and so on, i.e., we omit the argument function which stands for the time instant, t. In equation
(F.1), $I_i$ is the raw input at the $i$-th feature detector of the C-layer, $\lambda_i$ is the input stimulus (or trace) extracted from $I_i$.

![Diagram](image)

Figure F-1: Architecture of the Type-1 Adaptive Resonance Theory (ART) Based Artificial Neural Networks (ANNs) \([N=\text{Number of Neurons in the Comparison Layer}=\text{Maximum Size of the (Input) Feature Vector}, M=\text{Number of Neurons in the Recognition Layer}=\text{Number of Patterns Stored}]\).

Also note that $0 \leq I_i, \lambda_i \leq 1.0$. $A_1$, $B_1$ and $C_1$ are non-negative rate constants that govern learning, $\text{Max}(1, D_1) < B_1 < (1 + D_1)$ as dictated by the 2/3 rule, $C_j << B_1$ and $0 < \varepsilon << 1.0$. $z_{ij}$ is the Bottom-Up association strength (or LTM or connection weight) from the $i$-th neuron in the C-layer to the $j$-th pattern (switching pattern) in the R-layer. $z_{ji}$ is the top-down association strength (or LTM or connection weight) from the $j$-th pattern (switching pattern) in the R-layer to the $i$-th neuron in the C-layer. And,

$$f(\mu_j) = \begin{cases} 1 & \text{if the } j\text{-th SP of the R-layer has the highest activation} \\ 0 & \text{otherwise.} \end{cases}$$ (F.2)
Since the ART-1 based arbiter of Figure 5.4 (modified nodal structure of the MSN) is designed to recognize an SP based on the current connection request and the CC of the immediate past, in our case $\sum_j f(\mu_j) = 1$, unless the switch was inactive or idle in the immediate past time slot.

Now, we consider the STM activity $\mu_j$ of a pattern $v_j$ in the R-layer when the J-th node or pattern (i.e., the SP or CC) is active, i.e.,

$$ \varepsilon \frac{d\mu_j}{dt} = -\mu_j + (1 - A_1 \mu_j)[I_j + D_1 z_{j1}] + (B_1 + C_1 \mu_j) \quad (F.3) $$

And, when the R-layer is inactive, all top-down signals are in the sub-threshold region, i.e., $f(\mu_j)=0$ for all values of j. Therefore, we obtain:

$$ \varepsilon \frac{d\mu_j}{dt} = -\mu_j + (1 - A_1 \mu_j) I_j \quad (F.4) $$

The LTM traces from the neurons in the C-layer to the SPs in the R-layer, i.e., the connection strengths or the synaptic weights are computed using the following differential equations.

In the Bottom-Up case, the connection strength from neuron $v_i$ to the pattern $v_j$ is tuned according to the following differential equation.

$$ \frac{dz_{ij}}{dt} = K_{BU} [-E_{ij} z_{ij} + h(\lambda_i)] f(\mu_j) \quad (F.5) $$
In order to satisfy the Weber learning law and competitive learning mechanism [52], the following condition must be satisfied.

\[ E_{ij} = h(\lambda_i) + \frac{h(\lambda_k)}{L} \]

\[ \sum_{k \neq i} \]  \( (F.6) \)

where \( L \) is a non-zero positive integer representing the learning parameter, \( f(\mu_j) \) is same as defined earlier in this section, and \( h(\lambda_i) \) is as defined below.

\[ h(\lambda_i) = \begin{cases} 
1 & \text{if } \lambda_i > 0 \\
0 & \text{otherwise.} 
\end{cases} \quad (F.7) \]

Thus, we obtain:

\[ \frac{dz_{ij}}{dt} = K[(1 - z_{ij}) L h(\lambda_i) + z_{ij} (L \| \lambda_l - 1)] \]

\[ \quad (F.8) \]

where \( K = \frac{K_{BU}}{L} \) and \( h(\lambda_i) = 1 \) if the i-th neuron in the C-layer is active, i.e., it has a supra-threshold input value. At steady state, \( \frac{dz_{ij}}{dt} = 0 \), therefore,

\[ z_{ij} = \frac{L}{(L + \| \mathbb{I}_l - 1) = \frac{L}{(L + \| \mathbb{X}_i - 1)} \quad \forall i \text{ and } \forall j. \]

\[ \quad (F.9) \]

In general:

\[ \frac{dz_{ij}}{dt} = \begin{cases} 
K[(1 - z_{ij}) L + z_{ij} (L \| \lambda_l - 1)] & \text{if both } v_i \text{ and } v_j \text{ are active} \\
-K \| \mathbb{I}_l z_{ij} & \text{if } v_i \text{ is inactive and } v_j \text{ is active} \\
0 & \text{if } v_j \text{ is inactive.} 
\end{cases} \quad (F.10) \]
In the top-down case, the connection strength from the pattern \( v_j \) to neuron \( v_i \) follows the differential equation:

\[
\frac{dz_{ji}}{dt} = K_{TD} [-E_{ji}z_{ji} + h(\lambda_j)] f(\mu_j)
\]  

\( (F.11) \)

Assuming that \( K_{TD} = 1 = E_{ji}, \forall i \) and \( \forall j \), we find that:

\[
\frac{dz_{ji}}{dt} = \begin{cases} 
- z_{ji} + 1 & \text{if both } v_i \text{ and } v_j \text{ are active} \\
-z_{ji} & \text{if } v_i \text{ is inactive and } v_j \text{ is active} \\
0 & \text{if } v_j \text{ is inactive}
\end{cases} 
\]  

\( (F.12) \)

Here again at steady state, \( \frac{dz_{ji}}{dt} = 0 \), and when we assume that the next connection configuration or switching pattern is derived using the current switching pattern and the input request we find that \( z_{ji} = 1, \forall i \) and \( \forall j \).

Finally, since we are interested in fast learning and direct access to the learned SPs in order to cope with the speed of the supra-high-speed transmission links, we need the Bottom-Up-LTMs as:

\[
z_{ij}(0) = \begin{cases} 
\frac{L}{L + (1 \text{ or } X \text{ or } 1)} - 1 & \text{if } i \in X \\
0 & \text{if } i \notin X
\end{cases}
\]  

\( (F.13) \)

And the Top-Down -LTMs as:

\[
\begin{cases} 
1 & \text{if } i \in X \\
0 & \text{if } i \notin X
\end{cases}
\]  

\( (F.14) \)
For a 3x3 or 3x4 switch the value of \( X \) or \( I \) is 3 and during our simulation experiments (chapter 5 and [83]) we find that the value of \( L=2 \) yields highly desirable results. If the arbiter can not be trained with all possible switching patterns, it is useful to set \( L \) high at the beginning. Note that larger the initial value of \( L \), larger the bias toward novel SPs. Therefore, for an \( NxN \) switch where \( N \) is a very large integer, \( L \) should be set large initially.

F.5 A Step-Wise Procedure for Developing ART-1 Model for a Given Binary Pattern Recognition Problem

In this section, we present a step-wise procedure for developing the ART-1 model for a given pattern recognition problem. Note that, ART-1 deals with binary patterns only. The operation of the ART-1 Net consists of the following FIVE phases: Initialization, Recognition, Comparison, Search, and Training.

STEP-1: Initialization: For an \( N \)-feature (i.e., the number of neurons in the lower or C-layer is \( N \)) \( M \)-pattern (i.e., the number of patterns already stored in the ART-1 Net is \( M \), i.e., no. of patterns stored in the R-layer is \( M \)) ART Net at time \( t=0 \), the bottom-up weight, \( z_{ij}(t) \), is:

\[
0.0 < z_{ij}(t) \leq \frac{L}{L - 1.0 + (\text{No. of 1s in the pattern stored in the } j\text{-th neuron})}
\]

where \( i = 1, 2, ..., N \), \( j = 1, 2, ..., M \), and \( L \) is a constant, \( L > 1.0 \), and typically, \( L=2.0 \) for fast learning. \( L \) determines the degree of network's biasness towards uncommitted neurons -- irrespective of the value of the vigilance parameter, \( \nu \) -- in the
Recognition layer (R-layer) in response to a novel input pattern. Larger the value of \( L \), larger the bias.

The top-down weight, \( z_{ji}(t) \), is: \( z < z_{ji}(t) < 1.0 \), where \( z = [(B - 1.0)/D] \), and \( \text{Max}(1, D) < B < (1+D) \), where \( D \) is any integer. For fast learning, \( z_{ji}(0) \) can be assumed to be equal to unity. Note that, if \( z_{ji}(t) \ll 1.0 \), it will result in no matches in the Comparison layer (C-layer).

The vigilance parameter, \( \nu \) is: \( 0.0 < \nu \leq 1.0 \). \( \nu \) indicates the allowable level of mismatch between a stored (or learned) pattern and an input vector. Higher value of \( \nu \) implies finer categorization and the lower value indicates coarse distinctions. (Note: It is customary to start with a low value of \( \nu \) making coarse distinctions, then gradually increase the vigilance level to produce accurate categories at the end).

The outputs of all neurons in the R-layer and the output of the reset block are set to zero. Finally, at \( t=0 \), the output of the gain control blocks GAIN-1 and GAIN-2 (see Figure F-1) are set to unity, assuming that at least one of the features in the input vector is unity.

**STEP-2:** Bottom-up recognition or computation of the environmental expectation (or template). Apply the new input pattern \( I = \{I_i\}_{i=1..N} \), then extract the input (to the C-layer) vector, \( X = \{X_i\}_{i=1..N} \). This sets the output of the gain control blocks, GAIN-1 and GAIN-2 to unity iff at least one of the elements of \( X \) is unity. Then compute the following vectors in the sequence as indicated below:

\[
\begin{align*}
\mathbf{X} \quad & \quad \text{2/3 rule} \quad \mathbf{S} \quad & \quad \text{Weighted Sum} \quad \mathbf{T} \quad & \quad \text{Matching Score} \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad 
\end{align*}
\]
where \( X_i = \begin{cases} 1 & \text{if } I_i > \text{pre-specified threshold} \\ 0 & \text{otherwise} \end{cases} \)

Applying the 2/3 rule, compute the signal vector, \( \mathbf{S} \):

\[
\mathbf{S} = \begin{cases} \mathbf{X} & \text{iff all neurons in R-layer are inactive} \\ \mathbf{X} \land \mathbf{Y}^{(j)} & \text{if R-layer neuron J is activate} \end{cases}
\]

[Note: " \land " indicates logical AND operation]. Since initially (i.e., at \( t=0 \) \( \mathbf{Y}=0 \), and output of GAIN-1 is unity, any component of \( \mathbf{X} \) that is unity will provide the 2nd input to satisfy the 2/3 rule. Therefore, only the corresponding neuron will fire. Hence, we get \( \mathbf{S} = \mathbf{X} \). Now, \( \mathbf{T} \) is computed as the weighted sum of the components of \( \mathbf{X} \), as follows:

\[
T_j = \sum_{i=1}^{N} z_{ij} X_i, \quad \text{for } j = 1, 2, ..., M.
\]

Then compute \( Y_j \) from \( T_j \). Note that, the \( Y_j \) is simply the pattern stored in the R-layer corresponding to the matching score \( T_j \).

**STEP-3**: Top-down recognition or computation of the top-down template or the learned expectation. Compute the signal pattern, \( \mathbf{U} \) from the vector \( \mathbf{T} \) as follows:

\[
\mathbf{U} = \{ f(T_1), f(T_2), ......., f(T_M) \}.
\]

where

\[
f(T_j) = \begin{cases} 1 & \text{if } T_j = \text{Max} \{ T_k : k=1 \ldots M \} \\ 0 & \text{otherwise} \end{cases}
\]
That is, the best matching exemplar (i.e., the neuron with the highest matching score) is selected using extensive lateral inhibition as in Hamming network [128]. Then compute the top-down template pattern or learned expectation vector, \( \mathbf{\lambda} = \{ \lambda_1, \lambda_2, \ldots, \lambda_N \} \), where

\[
\lambda_i = D \cdot \sum_{j=1}^{M} f(T_j) \cdot \lambda_{ji}(t), \quad i = 1, 2, \ldots, N.
\]

(Note: The parameter \( D \) is obtained from step-1). Thus, for the above definition of \( f(T_j) \), we obtain:

\[
\lambda_i = \begin{cases} 
D \cdot \lambda_{ji}(t) & \text{iff the } J\text{-th neuron in R-layer is active} \\
0 & \text{otherwise}
\end{cases}
\]

Therefore, we obtain: \( \mathbf{\lambda} = \{ D \cdot t_{j1}(t), D \cdot t_{j2}(t), \ldots, D \cdot t_{jN}(t) \} \).

**STEP-4**: Vigilance test. At this stage, since the neuron with the highest matching score from the R-layer has fired, the feedback signal (i.e., the top-down signal or the learned expectation), \( \mathbf{\lambda} \) is no longer a zero vector. Consequently, the output of the gain control block GAIN-1 becomes inhibited. Now, because of the 2/3 rule, only those neurons in the C-layer will fire which receive simultaneous activations from the input vector \( \mathbf{X} \) and the learned expectation vector, \( \mathbf{\lambda} \). The signal vector, \( \mathbf{S} \) is now computed as the logical and (i.e., \( \land \)) of the vectors \( \mathbf{X} \) and \( \mathbf{\lambda}^{(J)} \), assuming that the J-th neuron primed from the R-layer. Therefore, \( \mathbf{S} = \mathbf{X} \land \mathbf{\lambda}^{(J)} \), when neuron J is active in R-layer. Now, compute the parameter, \( n \) as defined below:

\[
n = \sum_{i=1}^{N} t_{ji} \cdot x_i = \text{Number of } 1\text{s in the vector } \mathbf{S}.
\]
Then compute the parameter, \( d \) as defined by:

\[
d = \sum_{i=1}^{N} X_i = \text{Number of 1s in the input vector, } \mathbf{X}.
\]

Now compute the degree of similarity between the input vector, \( \mathbf{X} \) and the signal vector, \( \mathbf{S} \) by computing the ratio, \( r \) (where \( r = n/d \)). Then compare the ratio, \( r \) with the vigilance parameter, \( \nu \).

- If \( r < \nu \), execute STEP-5,
- If \( r = \nu \), execute STEP-6,
- If \( r > \nu \), execute STEP-7.

**STEP-5:** The case \( r < \nu \). If \( r < \nu \), the reset block generate an activation signal to disable (i.e., force the output to zero) the firing the neuron in the R-layer for the duration of the current classification (that neuron no longer takes part in the maximization process of STEP-3).

Thus, the gain control block GAIN-1 becomes activated, its output becomes unity and a new R-layer neuron is selected for priming the C-layer neurons by going back to STEP-3. After all the committed neurons of the R-layer has been tested and disabled (because none of them match the input pattern at the current vigilance level) STEP-8 is executed.

**STEP-6:** The \( r = \nu \) case. If \( r = \nu \), a neuron from the R-layer perfectly matches the input pattern corresponding to \( 1 \) at the current vigilance level. Therefore, STEP-9 is to be executed next.
STEP-7: The case \( r > \nu \). In this case, the ART machine enters a training cycle to adapt its stored (i.e., learned) pattern to the matching exemplar by computing the logical and between the stored pattern and the current input pattern and modifying both the Top-Down and Bottom-Up weight vectors associated with the firing R-layer neuron (i.e., the J-th neuron). Thus, we obtain:

\[
    z_{ji}(t+1) = z_{ji}(t)X_i, \text{ and } z_{ij}(t+1) = \frac{L \cdot z_{ji}(t) \cdot X_i}{N} \left[ L - 1 + \sum_{i=1}^{N} z_{ji}(t) \cdot X_i \right]
\]

STEP-8: Creation of a new pattern in R-layer. A previously uncommitted (i.e., unallocated) neuron in the R-layer is assigned to this novel input pattern and the weight vectors \( z_i(M+1) \) and \( z_i(M+1) \) corresponding to this new class are set to match the input pattern. The components of \( z_i(M+1) \) are all unity, i.e., their initial values. The 2/3 rule will make the signal vector, \( S \) identical to the input vector \( X \). The ratio, \( r \) becomes unity and the vigilance threshold is either satisfied or exceeded. (Note: Each additional neuron in the R-layer needs \( 2N \) connections for computing the matching score and \( M+1 \) connections are needed for lateral inhibitions of the \( M+1 \)-th neuron).

STEP-9: The classification of the current input pattern is complete. Next available input pattern is fed to the ART-1 machine and the process is repeated by going back to STEP-2.
F.6 A Numerical Example

In this section, we consider a four-feature (i.e., \( i=1, ..., 4 \)) three-pattern (i.e., \( j=1, ..., 3 \)) ART-1 Net for numeral recognition. Since we are considering four-feature patterns, the size of the input vectors \( \mathbf{I} \) and \( \mathbf{X} \) are four. The stored patterns are:

\[
\begin{align*}
Y_1 &= \{ 1, 1, 1, 1 \} \quad \text{... representing the numeral } 0. \\
Y_2 &= \{ 0, 0, 0, 1 \} \quad \text{... representing the numeral } 1. \\
Y_3 &= \{ 0, 1, 0, 1 \} \quad \text{... representing the numeral } 7.
\end{align*}
\]

Let us consider that a novel input pattern \( \mathbf{I} \) is applied to the ART-1 Net. The corresponding input pattern is \( \mathbf{X} = \{ 0, 0, 1, 1 \} \). We also assume the situation where fast learning occurs (i.e., \( \lambda=2 \) is assumed). Now we apply the nine-step procedure developed in section F-5 for recognizing the novel pattern at a vigilance level of 0.5 [i.e., \( \nu = 0.5 \)].

**STEP-1:** Initialization. The bottom-up weights are as follows:
\[
\begin{align*}
\{ z_{1i} \}_{i=1}^{4} &= \{ 2/5, 2/5, 2/5, 2/5 \}.
\{ z_{2i} \}_{i=1}^{4} &= \{ 0, 0, 0, 1 \} \\
\{ z_{3i} \}_{i=1}^{4} &= \{ 0, 2/3, 0, 2/3 \}
\end{align*}
\]

All the top-down weights are assumed to be unity, i.e.,
\[
z_{ji} = 1.0 \quad \text{for} \quad j = 1, 2, 3, \text{and} \quad i = 1, 2, 3, 4.
\]

The vigilance parameter, \( \nu \) is set to 0.5. And the output of the gain control blocks GAIN-1 and GAIN-2 will be unity.

**STEP-2:** Computation of environmental (bottom-up) expectation. As mentioned before, we have \( \mathbf{X} = \{ 0, 0, 1, 1 \} \). Since all the neurons of the R-layer are presently inactive,
output of GAIN-1 is unity and the signal vector, \( \mathbf{s} = \mathbf{x} \). The bottom-up expectation is now computed by taking the weighted sum of the components of \( \mathbf{x} \) (since now \( \mathbf{s} = \mathbf{x} \)). Thus we obtain:

\[
T_1 = [0 + 0 + (2/5) + (2/5)] = 4/5 = 0.8.
T_2 = [0 + 0 + 0 + 1] = 1.0.
T_3 = [0 + 0 + 0 + (2/3)] = 2/3 = 0.67.
\]

Since the pattern corresponding to the highest score (i.e., the pattern for the score \( T_2 \)) will be activated, the pattern \( Y_2 \) becomes activated.

**STEP-3**: Computation of learned (top-down) expectation. Since the 2nd neuron in the R-layer has the highest score, it is the only activated neuron in that layer. Therefore, the signal vector, \( \mathbf{u} \) becomes: \( \mathbf{u} = \{0, 1, 0\} \). The top-down expectation vector, \( \mathbf{v} \) becomes: \( \mathbf{v} = \{0, 0, 0, 1\} \).

Now the output of the gain control block GAIN-1 becomes zero and the new signal vector, \( \mathbf{s} \) is computed by taking logical and of the input vector, \( \mathbf{x} \) and the top-down expectation, \( \mathbf{v} \) as shown below: \( \mathbf{s} = \mathbf{x} \land \mathbf{v} = \{0, 0, 1, 1\} \land \{0, 0, 0, 1\} = \{0, 0, 0, 1\} \).

**STEP-4**: Vigilance test. Since \( \mathbf{x} = \{0, 0, 1, 1\} \), the number of 1s in \( \mathbf{x} \) is 2. Therefore, \( d = 2 \). And since \( \mathbf{s} \), as found in step-3, is \( \{0, 0, 0, 1\} \), the number of 1s in \( \mathbf{s} \) is 1. Therefore, \( n = 1 \). Hence the ratio, \( r = (n/d) = 1/2 = 0.5 \).

Since the vigilance threshold, \( \nu \) has been set to 0.5 in step-1, in this case \( r \) becomes equal to \( \nu \). Hence, step-6 is executed next.
STEP-6: The input pattern, $X = \{0, 0, 1, 1\}$ has been identified as the stored pattern $Y_2$ (i.e., the numeral 1) at a vigilance level of 0.5. Step-9 is executed next.

STEP-9: The next input pattern is fed to the ART-1 machine for the purpose of recognition.
BIBLIOGRAPHY


VITA AUCTORIS

NAME: Bhumip Khasnabish.

PLACE OF BIRTH: Kishoreganj, Bangladesh.


Gurudyal College, Kishoreganj, Bangladesh, 1975 - 1977, H.S.C.


University of Waterloo, Waterloo, Ontario, Canada, 1984 - 1986, M.A.Sc.

McMaster University, Hamilton, Ontario, Canada, 1986 - 1989, Research and Part-time Teaching Assistant.

University of Windsor, Windsor, Ontario, Canada, 1989 - todate, Ph.D. Research.