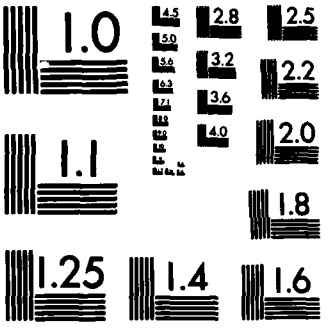
AD-A122 995 DESIGN CONSIDERATIONS FOR LARGE COMPUTER COMMUNICATION NETWORKS(U) CALIFORNIA UNIV LOS ANGELES SCHOOL OF ENGINEERING AND APPLIED SCIENCE F KAMOUN APR 76 1/5

UNCLASSIFIED UCLA-ENG-7642 DAHC15-73-C-0368 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ARPA CONTRACT NO. DAHC-15-73-C-0368

DESIGN CONSIDERATIONS FOR LARGE COMPUTER COMMUNICATION NETWORKS

F. KAMOUN

AD A122995

COMPUTER SCIENCE DEPARTMENT
School of Engineering and Applied Science
University of California
Los Angeles

83 12 23 078

COMPUTER SYSTEMS MODELING AND ANALYSIS GROUP
REPORT SERIES

Turn, R., "Assignment of Inventory of a Variable Structure Computer," January 1963, UCLA-ENG-6305.

Martin, D.F., "The Automatic Assignment and Sequencing of Computations on Parallel Processor Systems," January 1966, UCLA-ENG-6604 (AEC/ONR).

Coffman, E.G., "Stochastic Models of Multiple and Time-Shared Computer Operations," June 1966, UCLA-ENG-6638 (AEC/ARPA/ONR) AD No. 636-976.

Bovet, D.P., "Memory Allocation in Computer Systems," June 1968, UCLA-ENG-6817 (AEC/ARPA/ONR).

Baer, J.L., "Graph Models of Computations in Computer Systems," October 1968, UCLA-ENG-6846 (AEC:UCLA-10P14-51/ARPA/ONR) AD No. 678-753.

Russell, E.C., "Automatic Program Analysis," March 1969, UCLA-ENG-6912 (AEC:UCLA-10P14-72/ARPA/ONR) AD No. 686-401.

Koster, R., "Low Level Self-Measurement in Computers," December 1969, UCLA-ENG-6957 (AEC:UCLA-10P14-84).

Cerf, V.G., "Measurement of Recursive Programs," May 1970, UCLA-ENG-7043 (AEC:UCLA-10P14-90/ARPA).

Volansky, S.A., "Graph Model Analysis and Implementation of Computational Sequences," June 1970, UCLA-ENG-7048 (AEC:UCLA-10P14-93).

Cole, G.D., "Computer Network Measurements: Techniques and Experiments," October 1971, UCLA-ENG-7165 (ARPA) AD No. 739-344.

Hsu, J., "Analysis of a Continuum of Processor-Sharing Models for Time-Shared Computer Systems," October 1971, UCLA-ENG-7166 (ARPA) AD No. 739-345.

Ziegler, J.F., "Nodal Blocking in Large Networks," October 1971, UCLA-ENG-7167 (ARPA) AD No. 741-647.

Cerf, V.G., E. Fernandez, K. Gostelow and S. Volansky, "Formal Control-Flow Properties of a Model of Computation," December 1971, UCLA-ENG-7178 (AEC:UCLA-10P14-105).

Gostelow, K.P., "Flow Control, Resource Allocation, and the Proper Termination of Programs," December 1971, UCLA-ENG-7179 (AEC:UCLA-10P14-106).

Cerf, V.G., "Multiprocessors, Semaphores, and a Graph Model of Computation," April 1972, UCLA-ENG-7223 (AEC:UCLA-10P14-110).

Fultz, G.L., "Adaptive Routing Techniques for Message Switching Computer Communication Networks," July 1972, UCLA-ENG-7252 (ARPA).

Fernandez, E., "Activity Transformations on Graph Models of Parallel Computations," October 1972, UCLA-ENG-7287 (AEC:UCLA-10P14-116).

Gerla, M., "The Design of Store-and-Forward (A/F) Networks for Computer Communications," January 1973, UCLA-ENG-7319 (ARPA).

Talan, R., "Optimal Control of Tandem Queues," May 1973, UCLA-ENG-7333 (AFOSR).

Postel, Jonathan B., "A Graph Model Analysis of Computer Communications Protocols," January 1974, UCLA-ENG-7410 (ARPA).

Opderbeck, H., "Measurement and Modeling of Program Behavior and its Applications," April 1974, UCLA-ENG-7418 (ONR).

Lam, Simon S., "Packet Switching in a Multi-Access Broadcast Channel with Application to Satellite Communication in a Computer Network," April 1974, UCLA-ENG-7429 (ARPA).

Yavne, M., "Synthesis of Properly Terminating Graphs," May 1974, UCLA-ENG-7434, (AEC: UCLA-34P214-2).

Webster, F., "An Implementation of the Burroughs D-Machine," June 1974, UCLA-ENG-7449, (AEC: UCLA-34P214-6).

Sylvain, P., "Evaluating the Array Machine," August 1974, UCLA-ENG-7462, (NSF).

Kleinrock, L., "Computer Network Research Final Technical Report," August 1974, UCLA-ENG-7467 (ARPA).

Tobagi, F.A., "Random Access Techniques for Data Transmission over Packet Switched Radio Networks," December 1974, UCLA-ENG-7499 (ARPA).

White, D.E., "Fault Detection Through Parallel Processing in Boolean Algebra," March 1975, UCLA-ENG-7504 (ONR).

Wong, J.W-N., "Queueing Network Models for Computer Systems," October 1975. UCLA-ENG-7579 (ARPA).

Loomis, M.E., "Data Base Design: Object Distribution and Resource — Constrained Task Scheduling," March 1976, UCLA-ENG-7617 (ERDA: UCLA-34P214-26).

Kamoun, F., "Design Considerations for Large Computer Communication Networks," April 1976, UCLA-ENG-7642 (ARPA).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A122995

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

DESIGN CONSIDERATIONS FOR LARGE COMPUTER COMMUNICATION NETWORKS

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

UCLA-ENG-7642

7. AUTHOR(s)

Farouk Kamoun

8. CONTRACT OR GRANT NUMBER(s)

DAHC-15-73-C-0368

9. PERFORMING ORGANIZATION NAME AND ADDRESS

School of Engineering and Applied Science
University of California
Los Angeles, California 90024

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Advanced Research Projects Agency (ARPA)
1400 Wilson Blvd.
Arlington, Virginia 22209

12. REPORT DATE

April 1976

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

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

DTIC
ELECTE
DEC 29 1983
S D
H

18. SUPPLEMENTARY NOTES

Ph.D. Dissertation, March 1976

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

Large Computer Communication Networks
Packet Switching
Store-and-Forward
Distributed Computer Networks
ARPANET
Resource Sharing
Adaptive Routing
Deterministic Routing

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

This research deals with the specification, analysis and evaluation of some routing and topology design procedures for large store and forward packet switched computer communication networks. The procedures studied are an extension of present techniques and rely on a hierarchical clustering of the network nodes.

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19. KEY WORDS (Cont.)

Hierarchical Adaptive Routing
Performance Evaluation and Efficiency
Nodal Storage Limitation
Storage Sharing Schemes
Routing Updates
Routing Table
Clustered Routing
Simulation of Hierarchical Routing
Optimal Hierarchical Clustering Structure
Clustering Technique for Routing
Hierarchical Design of Large Networks
Topological Design
Computational Design Cost
Optimization
Average Message Delay
Queueing Models
Blocking
Loss System
Torus Networks
Average Path Length

20. Hierarchical adaptive routing schemes are investigated in the context of large networks. In particular, optimal clustering structures are determined so as to minimize the length of the routing tables used by the routing function. The effect of this reduction of routing information on the message path length is evaluated. Queueing models are developed in order to evaluate the delay-throughput performance of the hierarchical routing policies and to compare them with present schemes. The models prove the infeasibility of present routing procedures for large networks and demonstrate the remarkable efficiency of hierarchical routing schemes.

These queueing models represent an extension of Kleinrock's model for networks in that they consider nodal storage requirements and line overhead due to routing updates. Furthermore, several buffer sharing schemes are proposed and analyzed in order to optimally utilize nodal storage.

A hierarchical design methodology is also presented as a solution for the reduction of the computational cost involved in the topological design of large networks. A computational cost model is developed in order to determine that clustering structure which achieves a minimal cost. Such optimal structures are found which lead to very significant savings in computation.

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A

DTIC COPY INSPECTED

UCLA-ENG-7642
April 1976

Design Considerations for Large Computer Communication Networks

by

Farouk Kamoun

School of Engineering and Applied Science
University of California
Los Angeles

ACKNOWLEDGEMENTS

I wish to express my appreciation to my doctoral committee consisting of Professors David Cantor, Glenn Graves, James Jackson, Leonard Kleinrock (Chairman), Richard Muntz and Jacques Vidal. I am especially grateful to my advisor, Professor Leonard Kleinrock, for his continued guidance and friendship and the innumerable helpful discussions during the course of this research.

A debt of gratitude is owed to Mr. Moktar Latiri, former director of the School of Engineering of Tunis (ENIT), Tunisia, and Miss Leila Mogannam, Agency of the International Development (USAID) in Tunis, for their assistance in making my program of studies in the United States possible.

Gratitude is also owed to the personnel of the Computer System Modelling and Analysis Group in the Computer Science Department at UCLA for providing the stimulating atmosphere in which this research was performed. Thanks also go to Brent Ellerbrock, Chaim Kaltgrad and Frank Heath for their programming help. I am also indebted to Diana Skocypec for her continued friendship and administrative help and to George Ann Hornor for her patient assistance in the typing and preparation of this dissertation.

This research was supported in part by the Advanced Projects Agency of the Department of Defense under Contract Number DAHC 15-73-C-0368 and by a Ph.D. scholarship from the Agency of International Development, U.S. Department of State.

ABSTRACT

This research deals with the specification, analysis and evaluation of some routing and topology design procedures for large store and forward packet switched computer communication networks. The procedures studied are an extension of present techniques and rely on a hierarchical clustering of the network nodes.

Hierarchical adaptive routing schemes are investigated in the context of large networks. In particular, optimal clustering structures are determined so as to minimize the length of the routing tables used by the routing function. The effect of this reduction of routing information on the message path length is evaluated. Queueing models are developed in order to evaluate the delay-throughput performance of the hierarchical routing policies and to compare them with present schemes. The models prove the infeasibility of present routing procedures for large networks and demonstrate the remarkable efficiency of hierarchical routing schemes.

These queueing models represent an extension of Kleinrock's model for networks in that they consider nodal storage requirements and line overhead due to routing updates. Furthermore, several buffer sharing schemes are proposed and analyzed in order to optimally utilize nodal storage.

A hierarchical design methodology is also presented as a solution for the reduction of the computational cost involved in the topological design of large networks. A computational cost model is developed in order to determine that clustering structure which achieves a minimal cost. Such optimal structures are found which lead to very significant savings in computation.

CONTENTS

		Page
LIST OF FIGURES	. . . . .	xiii
LIST OF TABLES	. . . . .	xvix
CHAPTER 1	INTRODUCTION	1
1.1	Routing for Packet Switching Networks	3
1.2	The Design of Computer Networks	5
1.3	Summary of Results	7
CHAPTER 2	HIERARCHICAL ROUTING PROCEDURES FOR LARGE COMPUTER NETWORKS	12
2.1	Important Aspects of Adaptive Routing Schemes	13
	2.1.1 The Routing Problem in Networks . . .	13
	2.1.2 Adaptive Routing Policies	14
2.2	m-level Hierarchical Routing (MHR) Schemes .	17
2.3	Minimum Routing Information	21
	2.3.1 Tree Representation for the MHC and Problem Statement	22
	2.3.2 Real-Valued Solution of the Optimization Problem	26
	2.3.3 Integer Solution	34
	2.3.4 Optimality with No "Self-Entries" in the Routing Table	43
2.4	Conclusions	50
CHAPTER 3	STATIC EVALUATION OF HIERARCHICAL ROUTING: BOUNDS ON THE INCREASE IN NETWORK PATH LENGTH.	52
3.1	Path Characteristics for Hierarchical and Non-Hierarchical Adaptive Routing Policies .	52
	3.1.1 Non-Clustered Routing (NCR)	54
	3.1.2 Specification of the MHR Schemes . .	55

CONTENTS (continued)

		Page
	3.1.3 Path Characteristics	67
3.2	Bounds on the Increase in Path Length . . .	74
3.3	Static Performance Evaluation of the MHR Schemes for a Family of Networks	83
	3.3.1 A Family of Large Distributed Networks	83
	3.3.2 Limiting Performance of the MHR Schemes	86
	3.3.3 Static Performance Evaluation for the MHR's: Numerical Applications . . .	90
3.4	Conclusion	106
CHAPTER 4	STOCHASTIC PERFORMANCE EVALUATION OF THE HIERARCHICAL ROUTING	107
4.1	General Considerations	108
	4.1.1 Delay Analysis in S/F Computer Networks: Kleinrock's Model	108
	4.1.2 A Class of Symmetrical Networks . .	112
	4.1.3 Continuous Modelling of the Hierarchical Routing	117
4.2	A Queueing Model with No Updates and No Storage Limitation	120
	4.2.1 Degradation in Throughput at Constant Delay (Primal Scaling)	121
	4.2.2 Degradation of Delay at Constant Throughput per Node (Dual Scaling) .	126
4.3	A Queueing Model with Updates and No Storage Limitation	126
	4.3.1 Priority Model for a Channel	128
	4.3.2 Network Model	132

CONTENTS (continued)

		Page
	4.3.3 Performance Evaluation of the Hierarchical Routing	133
4.4	A Queueing Model with No Updates and with Storage Limitation	144
	4.4.1 A Loss-Queueing Model for Symmetrical Networks	147
	4.4.2 Performance Evaluation of Hierarchical Routing	168
4.5	A Queueing Model with Updates and Storage Limitation	191
4.6	Conclusion	194
CHAPTER 5	CLUSTERING ISSUES AND SIMULATION OF THE MHR SCHEMES	195
5.1	Clustering Issues	195
	5.1.1 Characterization of a "Good" Clustering	196
	5.1.2 Nearness Measures	198
	5.1.3 Clustering Techniques	200
5.2	Simulation of the MHR Schemes	206
	5.2.1 The Simulator	206
	5.2.2 Simulation of the ARPANET	210
	5.2.3 Simulation of a 64-Node Torus . . .	214
5.3	Conclusion	218
CHAPTER 6	TOPOLOGY DESIGN CONSIDERATIONS FOR LARGE COMPUTER COMMUNICATION NETWORKS	219
6.1	Introduction	219
6.2	The Topology Design Problem	221

CONTENTS (Continued)

		Page
6.2.1	Delay Analysis	221
6.2.2	The Communication Cost	222
6.2.3	Traffic Requirement	223
6.2.4	Reliability	223
6.2.5	Topology Design Problem	224
6.3	The m-level Hierarchical Topology (MHT) Design Procedure	226
6.3.1	Step a: MHC of the Set of Nodes . .	227
6.3.2	Decomposition Step	233
6.3.3	Design and Evaluation Steps	233
6.4	Optimal Clustering Structure, General Case .	234
6.4.1	Expression of the Computational Cost and Problem Statement	235
6.4.2	Optimal Solution	239
6.5	Optimal Clustering Structure with Uniform Design Strategy and Gate Assignment	243
6.5.1	Optimal Expressions Given m	244
6.5.2	Global Optimum	248
6.5.3	Irreducibility	255
6.5.4	Variations and Limiting Behavior of the Optimal Solution with Respect to the Design Variables	257
6.5.5	Comparison of the Optimal Solution with Two Feasible Solutions	276
6.5.6	Suboptimal Integer Solution	279
6.6	Application to Other Special Cases	287
6.7	Delay Expression for Hierarchical Networks .	288

CONTENTS (continued)

		Page
6.8	Conclusion	291
CHAPTER 7	CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH	293
7.1	Conclusions	293
7.2	Future Research	294
BIBLIOGRAPHY		296
APPENDIX A	PATH LENGTH IN GRID AND TORUS NETWORKS	302
A.1	Definitions	302
A.2	Average Path Length of a Grid	303
A.3	Average Path Length of a Torus	306
A.4	Distribution of Path Lengths in a Square Torus	308
APPENDIX B	ANALYSIS OF SHARED STORAGE IN A COMPUTER NETWORK ENVIRONMENT	312
B.1	Complete Partitioning (CP)	315
B.2	Complete Sharing (CS)	318
	B.2.1 General Case	318
	B.2.2 Limiting Behavior	324
	B.2.3 Special Case: Equal ρ_i's	329
	B.2.4 Comparison of CP and CS	335
B.3	Sharing with Maximum Queue Lengths (SMXQ)	339
	B.3.1 General Case	340
	B.3.2 Special Case: Equal ρ_i's	346
B.4	Sharing with Minimum Allocations (SMA)	349
	B.4.1 General Case	349

CONTENTS (continued)

		Page
	B.4.2 Special Case: $\rho_i = \rho$, $a_i = a$ . .	355
B.5	Sharing with Maximum Queue Length and Minimum Allocations: SMQMA.	358
B.6	Further Numerical Results and Comparisons .	359
B.7	Conclusion	360
APPENDIX C	PROPOSITION 6.1 AND ITS PROOF: APPLICATION TO SOME SPECIAL CASES	364
C.1	Optimal Clustering Structure	364
C.2	Application to Other Special Cases	380
	C.2.1 Uniform Design Strategy, Variable Gate Assignment	381
	C.2.2 Proportional Gate Assignment	385

LIST OF FIGURES

		Page
2.1	A 3-Level Clustered 24-Node Network	19
2.2	A Tree Representation of a 3-Level Clustered Net . .	20
2.3	Routing Table of Node 1.1.1	20
2.4	Tree Structure of the Hierarchical Clustering	23
2.5	Minimum Relative Table Length, ℓ/N, Given m	35
2.6	Ratio of Table Lengths at Optimality Given m, and at Global optimality, ℓ/ℓ_*	36
2.7	Minimum Relative Table Length, ℓ/N, versus the Number of Nodes	37
2.8	Integer Minimum Relative Table Length, Given m . . .	44
2.9	Slack in Size Constraint with the Integer Optimization	45
3.1	Routing Table at Node s, with an m-Level Hierarchical Clustering	57
3.2	A 2-Level Clustered Network	61
3.3	Equivalent Representation of Cluster $C_{k-1}(t)$	71
3.4	Relative Bound on the Increase in Path Length at Global Minimum Table Length	91
3.5	Relative Bound on the Increase in Path Length, E, versus the Relative Table Length, ℓ/N	93
3.6	E versus ℓ/N, Linear Axis	94
3.7	Lower Bound on the Ratio of Path Lengths Without and with Clustering, LB(h/hc)	96
3.8	LB(h/hc) versus ℓ/N, Linear Axis	97
3.9	Decrease in Table Length for a Given Maximum Increase in Path Length	98
3.10	Continuous Number of Levels, m	99
3.11	Relative Bounds on the Increase in Path Length with a Continuous and a Discrete m, $1 \leq m \leq 2$	102
3.12	Relative Bounds on the Increase in Path Length with a Continuous and a Discrete m, $1 \leq m \leq 3$	104
4.1	Degradation in Throughput at Constant Delay, $LB(\Gamma_c/\Gamma)$; Model with no Updates and no Storage Limitation	123
4.2	$LB(\Gamma_c/\Gamma)$ versus ℓ/N, Linear Axis	124

LIST OF FIGURES (continued)

		Page
4.3	Degradation in Throughput at Constant Relative Table Length	125
4.4	Degradation in Delay at Constant Throughput per Node; Model with no Updates and no Storage Limitation	127
4.5	State of the Queue as seen by an Arriving Data Message	130
4.6	Throughput at Constant Delay; Model with Updates, $\lambda_u = \lambda_u^0$	139
4.7	Throughput at Constant Delay; Model with Updates $\lambda_u = N^{1/4}\lambda_u^0/2$	140
4.8	Throughput at Constant Delay; Model with Updates, $\lambda_u = N^{1/2}\lambda_u^0/2$	141
4.9	Table Length Associated with the Lower Bound Envelope on the Throughput; $\lambda_u = \lambda_u^0$	142
4.10	Delay at Constant Throughput per Node; Model with Updates, $\lambda_u = \lambda_u^0$	145
4.11	Delay at Constant Throughput per Node; Model with Updates, $\lambda_u = N^{1/4}/2\ \lambda_u^0$	146
4.12	Model of the S/F Function of a Node	149
4.13	Solution of Eq. (4.49)	159
4.14	Normalized Throughput ρ_s, versus Normalized Load ρ, for a 121-Node Torus with Storage Limitation	162
4.15	ρ_s versus ρ; semi-log Representation	163
4.16	Normalized Delay $\mu C_0 T$ versus Load ρ, for a 121 Node Torus with Storage Limitation	164
4.17	Normalized Delay $\mu C_0 T$, versus Throughput ρ_s, for a 121-Node Torus with Storage Limitation	165
4.18	Probability of Loss for a 121-Node Torus with Storage Limitation	166
4.19	Relative Path Length of Successful Traffic, n_s/h; for the 121-Node Torus	167
4.20	Buffer Scaling: Normalized Throughput versus Normalized Load	170

LIST OF FIGURES (continued)

		Page
4.21	Buffer Scaling: Maximum Normalized Throughput versus Network Size	171
4.22	Buffer Scaling: Probability of Success	172
4.23	Buffer Scaling: Normalized Delay and Average Path Length of Successful Traffic	173
4.24	Throughput ρ_s, versus Load ρ, with Different Degrees of Clustering	180
4.25	Maximum Throughput Obtained with the Hierarchical Routing Model with No Updates and With Storage Limitation	183
4.26	Network Delay at Maximum Throughput with the MHR	184
4.27	Probability of Success at Maximum Throughput.	185
4.28	Behavior of the Number of S/F Buffers per Node, with the MHR	186
4.29	Behavior of the Relative Increase in Path Length with the MHR	187
4.30	Maximum Throughput with a Delay Constraint, Performance of the MHR: Model with No Updates and With Storage Limitation	188
4.31	Network Delay at Maximum Throughput with a Delay Constraint	189
4.32	Probability of Success at Maximum Throughput with a Delay Constraint	190
4.33	Maximum Lower Bound Throughput Obtained with the Hierarchical Routing; Model with Updates and Storage Limitation	193
5.1	ARPANET, Planar Map as of June 1974	204
5.2	Minimum Nearness (Threshold) in the "Measured" Clustering of the ARPANET	205
5.3	Sample Outcome of the Clustering Technique Based on "Measured" Nearness Measure; $\ell = 17$	207
5.4	Sample Outcome of the Clustering Technique Based on "Static" Nearness Measure; $\ell = 17$	208
5.5	Simulation of the ARPANET as Operated with an MHR Scheme	213
5.6	Clustered 64-Node Torus Network	215

LIST OF FIGURES (continued)

		Page
5.7	Simulation of a 64-Node Torus as Operated with the MHR Scheme	217
6.1	A 3-Level Hierarchical Network	230
6.2	Minimum Computational Cost $G(m, \alpha, \beta,)$, Given m; $\alpha = 3$, $\beta = 1$	260
6.3	Ratio of Computational Cost at Optimality Given m, and at Global Optimality, G/G_*; $\alpha = 3$, $\beta = 1$	261
6.4	Minimum Computational Cost $G(m, \alpha, \beta)$ Given m; $\alpha = 3$, $\beta = 3$	262
6.5	Ratio of Computational Cost at Optimality Given m, and and at Global Optimality; $\alpha = 3$, $\beta = 3$..	263
6.6	Minimum Computational Cost $G(m, \alpha, \beta)$, Given m: $\alpha = 5$, $\beta = 1$	264
6.7	Ratio of Computational Cost at Optimality Given m, and at Global Optimality, G/G_*; $\alpha = 5$, $\beta = 1$	265
6.8	Minimum Computational Cost $G(m, \alpha, \beta)$, Given m; $\alpha = 5$, $\beta = 3$	266
6.9	Ratio of Computational Cost at Optimality Given m, and at Global Optimality, G/G_*; $\alpha = 5$, $\beta = 3$	267
6.10	Root of dn_k/dm	268
6.11	Variations of the Optimal Degrees with Respect to m	270
6.12	Optimal Number of Levels in the Hierarchical Design	273
6.13	Comparison of the Optimal Solution with Two Feasible Solutions; $N = 10^4$, $\alpha = 3$	280
6.14	Comparison of the Optimal Solution with Two Feasible Solutions; $N = 10^4$, $\alpha = 5$	281
6.15	Suboptimal Integer Solution, Given m; $\alpha = 3$, $\beta = 3$	283
6.16	Ratio of Suboptimal Integer Solution, Given m, to the Real-Valued Global Optimal Solution; $\alpha = 3$, $\beta = 3$	284
6.17	Suboptimal Integer Solution, Given m; $\alpha = 5$, $\beta = 1$	285
6.18	Ratio of the Sub-optimal Integer Solution, Given m, to the Real Valued Global Opimum Solution; $\alpha = 5$, $\beta = 1$	286
6.19	Illustration of the Tree Path of class-k Traffic	289
A.1	p.q Grid	304

LIST OF FIGURES (continued)

		Page
A.2	Torus Nets	304
A.3	Square Torus	309
B.1	Storage Sharing Schemes	314
B.2	State-Transition-Rate Diagram	319
B.3	Average Number of Customers in the System, CS Scheme with Asymetric Input Rates	328
B.4	Probability of Blocking; CS Scheme	332
B.5	Normalized Throughput, CS Scheme	333
B.6	Normalized Delay; CS Scheme	334
B.7	Comparison of CP and CS: Blocking	336
B.8	Comparison of CP and CS: Utilization	337
B.9	Comparison of CP and CS: Delay	338
B.10	Comparison of the Four Schemes: Blocking	361
B.11	Comparison of the Four Schemes: Utilization	362
B.12	Comparison of the Four Schemes: Delay	363
C.1	Comparative Behavior of the Two Sides of Eq. (C.16)	385

LIST OF TABLES

		Page
3.1	RT at Node 12	62
3.2.a	RT's at the Exchange Nodes	63
3.2.b	RT's at the Exchange Nodes	64
3.3	Aggregation of Routing Information	65
3.4	Intermediate Update	65
3.5	Final Outcome	66
3.6	Matching the Address Vectors	67
4.1	Critical Values of N	143

CHAPTER 1

INTRODUCTION

One of the main reasons for the great interest in computer networks is the considerable economy that can be achieved through resource sharing [ROBE 70]. Among such resources we include computer power for load sharing, specialized hardware, specialized software, data banks, etc.

Such networks are called distributed computer communication networks and they made their first appearance with the ARPANET [HEAR 70], [KLEI 70], [FRAN 70], [CARR 70], [ROBE 70].

Computer networks are also emerging as very efficient means for data communication between remote locations. The first commercial data carrier, TELENET, is already operational.

At the root of this booming demand for computer networks is the ever increasing need for computer and data communication powers. It is projected that by 1980 approximately a quarter million terminals will be in use in Europe [EURO 73], [PETE 73] and as many as four or five million in the United States [KLEI 76].

A very important component of the network is the communication subnetwork. This includes the hardware and software specifically dedicated to the transfer of data from node to node. Many alternative communication schemes can be implemented at the subnet level. Among these are: circuit switching [PORT 71], packet switching (a form of store-and-forward communication) [KLEI 64], radio broadcasting [ABRA 70],

satellite communication [LAM 74], or any combination of the above, etc.

The selection of the best scheme is a difficult problem and depends very much on the nature of the traffic to be handled by the network [CLOSS 72A, 72B], [MIYA 75]. The bursty nature of computer traffic, as well as the continuously decreasing cost of computer hardware [ROBE 74], very much favor the packet switching as the technology for us to consider.

The basic concepts and the first packet switching computer network were developed by the United States Department of Defense Advanced Research Projects Agency (ARPA). This network (ARPANET), in operation since 1969, has been an enormously successful demonstration of the packet switching technique. It has resulted in the appearance of a multitude of other networks throughout the world (the NPL network in England, CYCLADES in France, etc.).

Present computer networks may be characterized as small to moderate sized networks (57 nodes for the ARPANET as of December 1975). The predictions mentioned above indicate that, in fact, large networks of the order of hundreds (or even possibly thousands) of nodes are imminent.

In the course of developing the ARPA network, a design methodology has evolved which is quite suitable for the efficient design of small and moderate sized networks [FRAN 72], [GERL 73A]. Unfortunately the cost of conducting the design is prohibitive if these same techniques are extrapolated to the large network case. Indeed, not only does the cost of design grow exponentially with the network size, but also the cost of a straightforward adaptive routing procedure becomes

prohibitive. Other design and operational procedures (techniques) must be found which handle the large network case and such techniques form the subject of this dissertation.

1.1 Routing for Packet-Switching Networks

In a packet switching network, messages are partitioned into segments called packets which then are transmitted through the network using the store-and-forward switching. That is, a packet traveling from source S to destination D is received and "stored" in queue at any intermediate node K while awaiting transmission, and is then sent "forward" to node P, the next node on the route from S to D, when channel (K,P) permits.

The selection of the next node P is made by a well-defined decision rule referred to as the routing policy. Several classification schemes have been devised to characterize routing policies [KLEI 64], [FULT 72], [GERL 73B], [MCCO 75], [MCQU 74]. Generally speaking, routing policies may be divided into two main classes: deterministic and adaptive. While deterministic routing is more attractive to use at the design phase, adaptive policies are essential for the operation of the network. A main objective of this dissertation is to specify and evaluate adaptive routing policies for large networks.

The major goal of adaptive routing policies is to sense changes in the traffic distribution and network status and then to route messages such that the conjested and damaged areas of the network are avoided. As a consequence, they adjust to load fluctuations and node or branch failures. The most commonly used adaptive policies base their decision

on routing information (delay estimate, excess capacity estimate, hop number estimate) stored in tables at each node of the network. These tables identify the output line to select for each destination in the network. They are updated periodically or asynchronously, or a combination of both, using routing information collected internally (at each node) and/or provided from neighboring nodes.

For large computer networks (on the order of many thousands of nodes or more), the length of the routing table (which directs the traffic through each node) will grow lineraly (one entry per node) with the number of nodes, and therefore the storage required to contain this list in each node will be extremely costly. Also, as a direct consequence of these large table lengths, the cost of interchanging routing information among the network nodes will also grow and will represent a significant burden on the communication lines themselves. All these considerations suggest that a reduction of routing table length of some sort is called for.

The main idea in reducing the table length is to provide at any node's (say i) routing table, one entry per destination node for those nodes which are close to i (in terms of a hop distance or some nearness measure) and one entry per <u>set</u> of nodes for those nodes located further away from i. The size of this set may increase with the (average) distance from i to the set of nodes. This partitioning of nodes into sets may be realized through a hierarchical clustering of the nodes. Consider a large distributed network and assume that we can realize the grouping of nodes into clusters, clusters into superclusters, etc. Indeed, let us assume that there is to be an m-level hierarchy. Provided such a

clustering structure, a routing table at any node (say i) will contain an entry for each node in the same cluster as i, and an entry per cluster in the same supercluster as i, etc., thus achieving a reduction of routing information.

It is such a scheme that we intend to define, analyze, and discuss. Fultz [FULT 72], McQuillan [MCQU 74] and others proposed similar schemes but did not provide any quantitative or experimental analysis as we do here.

1.2 The Design of Computer Networks

In our previous considerations, we were given a large distributed computer communication network and the problem was to devise an appropriate adaptive routing scheme which would operate efficiently with a fairly small amount of routing information. Now, going one step back, we are interested in designing the topology of a large network under some cost and performance constraints.

Several different formulations of the design problem related to the communication subnetwork can be found in the literature [FRAN 72], [GERL 73A]. Generally they correspond to different choices of performance measures, design variables and constraints. Here, we select the following very general formulations.

Given: Node locations

Channel capacity options

Minimize: Total communication cost

Over: Topology

Channel capacities

Routing policies

Subject to: Delay constraint

Reliability constraint

Traffic requirement

Along those formulations, several solutions have been proposed and applied to ARPA-like network designs. However, for networks with more than a few hundred nodes, we recognized that present procedures become prohibitively expensive because of the large amount of computer time and storage needed to perform the optimization step. Design procedures, based also on a hierarchical clustering of the network nodes, have been proposed [FRAN 73], [GERL 73B] [COVI 74] to substantially reduce the design cost. Again consider a set of nodes, initially not connected and assume that we wish to cluster the nodes into groups which themselves will be clustered into higher level groups, and so on up to a specified number of hierarchical levels. Once the hierarchical classes are defined, then the previously developed network design techniques for moderate sized networks may be used to design each cluster level separately.

Several questions arise as to the optimal clustering structure, the decomposition of the global performance variables and requirements which then lead to a set of smaller design problems. Frank and others [FRAN 72], [FRAN 73] showed from a feasibility study of a 1000 node network that indeed, hierarchical structures are desirable for the design of large networks. They also posed the same questions concerning the

clustering structure, but failed to answer them for the general case of an arbitrary number of hierarchical levels. Such questions will be addressed in this dissertation.

1.3 Summary of Results

The above considerations demonstrate the need for some new design and operational (routing) procedures for the large network case. They also suggest that methods based on some sort of decomposition scheme appear to be desirable for both the routing and the design of large networks.

The goal of this dissertation is twofold. First, we develop analytic models with which we can predict and optimize the performance of a hierarchical routing in large distributed networks. Second, we define and optimally specify the decomposition step in a hierarchical design of large networks.

Chapter 2 is primarily concerned with the introduction of the hierarchical routing schemes and their underlying hierarchical clustering structure as solutions to the reduction of the routing information and its associated overhead. An optimal clustering structure is found to minimize the length of the routing table and consequently it results in a minimum cost routing scheme. Enormous gains can be achieved whereby the table length may be reduced from N (N = number of nodes) to e ln N. However, a shortcoming of the reduction in routing information is the increase in the path length of a message in the network.

In Chapter 3, we examine the effect of hierarchical routing on network path length. Bounds are derived to evaluate the maximum increase

in path length for a given table reduction. The bounds demonstrate a major result; in the limit of a very large network, enormous table reduction may be achieved with no significant increase in network path length.

The reduction in table length means that more channel capacity and storage are available for the transmission of data traffic in the network. However, this gain in communication power may have to be partially or completely paid back to handle the excess traffic caused by longer network paths.

In Chapter 4, we address the above issue. In particular, we are interested in the trading relations among the table reduction, the nodal storage, the channel capacity, the network size, the throughput and the delay. Several queueing models are developed to capture and exhibit the interrelationships among these variables. The models demonstrate that for some reasonable cost and performance constraints, and for a class of symmetrical and distributed networks, routing procedures under their present form (non-hierarchical) become infeasible for a network size beyond some "critical" value; on the other hand, hierarchical routing with an optimally selected table length, preserves a remarkably good network performance for a phenomenal range of the network sizes. The particular numerical examples show that the transition point where hierarchical routing becomes certainly better than a non-hierarchical one occurs for a network size between 100 and 200 nodes.

The queueing models developed in this chapter represent, in a major part, an extension of Kleinrock's model for networks [KLEI 64]. They provide us with some new results concerning the effect of updates

and storage on network performance.

In Chapter 5 we address the following two issues: (i) the assignment of nodes to clusters, clusters to superclusters, etc., given an arbitrary network and clustering structure; (ii) the evaluation of hierarchical routing as applied to a more general network environment such as the ARPANET. In this chapter we mainly lay the framework and introduce a methodology in defining nearness measures between pairs of nodes. A clustering technique (Complete Linkage Method) has been modified in order to fit the constraints of our specific network environment. The nearness measures and the clustering technique are utilized to achieve the clustering of the June 1973 ARPANET.

Furthermore, the simulation of a 64-node torus net confirms our theoretic results and shows that even for such a moderate sized network, an appropriately selected hierarchical routing begins to exhibit improvements in network performance as compared to a non-hierarchical scheme.

In Chapter 6 we address some issues related to the hierarchical design of large networks. The emphasis is on the determination of a clustering structure to be used in the design phase and which minimizes the computational cost of the design. Such a cost is assumed to grow exponentially with the number of nodes in the subnet to be designed. We present optimum results both for the number of clusters, superclusters, etc., and the number of hierarchical levels when the same design strategy (technology) is considered at all levels. Optimal clustering structures are also determined when different design strategies are considered, provided that the number of levels is given. An expression of the average delay of a message in such a hierarchical network is also

provided in terms of the average delays in the layer subnets composing the network. This decomposition should consequently lead to the design of smaller subnetworks for which we can utilize present design strategies.

In Chapter 7 we give some concluding remarks and suggest topics for further research.

Some intermediary, but general, results were required which proved to be quite useful in the analytic performance evaluation of hierarchical routing in large networks (Chaper 4). These results form the object of Appendixes A and B.

In Appendix A we derive a closed form expression for the average shortest path length in grid-type networks. Also the distribution and the corresponding z-transform of path lengths in a torus network are defined and determined.

In Appendix B we consider various schemes for sharing a pool of buffers among a set of communication channels in a single network node environment [KAMO 76]. Several sharing schemes are examined and the results of the analysis are presented and displayed in a fashion which permits one to establish the tradeoffs among blocking probability, utilization, throughput and delay. The study shows that no one scheme is always optimal, and that the selection of a particular scheme should fit the particular application considered. In general, we find under normal operating conditions that sharing, with some restrictions on the contention of space, is certainly desirable, especially when little storage is available.

In Appendix C we present some proofs and complementary results for Chapter 6.

Even though this research addressed some design issues which arise in large computer communication networks, the methodology developed may be applicable to more general and large systems. In particular we mention a possible application to packet-radio networks, where clustering of terminals in groups may alleviate some of the difficulties encountered in performing the routing function.

In summary, we list below the major contributions of this research.

(1) We characterize and evaluate the performance of hierarchical routing for large packet switched computer networks; models are developed to determine clustering structures which lead to a minimal routing cost (storage, capacity) and their effect on the network path length. More importantly, we can determine the effect of table reduction in terms of network throughput and delay. As a result, we are able to demonstrate that hierarchical routing is remarkably efficient (in fact, necessary) for large distributed networks.

(2) A general methodology is specified for the design of large hierarchical networks. A model is developed to determine a partitioning structure so as to minimize the computational cost involved in the design of such networks.

(3) Models are developed to study the effect of storage in a single node and in a network environment. Several storage sharing schemes are specified and analyzed in order to make best use of such a resource (storage).

CHAPTER 2

HIERARCHICAL ROUTING PROCEDURES FOR LARGE COMPUTER NETWORKS

The most commonly used distributed adaptive routing policies base their decisions on routing information (delay, hop numbers) stored in tables at each node of the network. These tables identify the output line to select for each destination in the network. They are also regularly updated using routing information collected internally (at each node) and/or provided from neighboring nodes.

For large computer networks, on the order of many thousands of nodes, the routing information may become excessively costly in terms of nodal storage required to keep the tables, CPU time needed to perform the updates, and channel utilization incurred by the exchange of routing information. These considerations make current adaptive routing policies unattractive for large networks.

In this chapter, first, we will review the important aspects of adaptive routing procedures for computer networks, then we will discuss general problems associated with the impact of the network's size on those procedures. Next, routing schemes, based on hierarchical partitioning of the network, are presented as a solution for reducing the amount of routing information and the associated overhead. The focus of this chapter is to determine the clustering (partitioning) structure which results in a minimum amount of routing information and, consequently, in a minimum cost routing scheme. Unfortunately, the gains obtained from such schemes are accompanied with an increase in the path

length of a message in the network which results in a degradation of performance. This issue will be the object of the next chapter.

2.1 Important Aspects of Adaptive Routing Schemes

2.1.1 The Routing Problem in Networks

In a packet switching network, messages are partitioned into smaller segments called packets which then are transmitted through the network using the store-and-forward technique. That is, a packet traveling from source S to destination D is received and "stored" in queue at any intermediate node K, while awaiting transmission, and is then sent "forward" to node P, the next node on the route from S to D, when channel (K,P) permits [KLEI 64], [FULT 72], [KAHN 72], [MCQU 74] [KLEI 76].

The selection of the next node, P, is made by a well-defined decision rule referred to as the routing policy. Routing policies may be divided into two main classes: deterministic and adaptive [GERL 73C], [FULT 72]. A deterministic policy is time invariant. It may be selected to provide the optimal routing for a network in steady state, which has a given traffic requirement governed by a certain probabilistic distribution (see Chapter 6 for more details), and under the assumptions that no failures can occur. The optimal policy may be determined analytically [FRAT 73] and, therefore, it is attractive to use it in the design phase [GERL 73A]. An adaptive policy is time varying and bases its decision on some measure of the observed traffic. It can adjust to load fluctuations and node or branch failures. For

such a reason it is more attractive for real network operation than a deterministic policy. However, the performance evaluation of an adaptive policy requires time-consuming simulations, which makes it inadequate for use in the network design phase.

2.1.2 Adaptive Routing Policies

The major goal of an adaptive routing procedure is to sense changes in the traffic distribution and network status and then to route messages such that the congested areas of the network are avoided. It is very important for those procedures to adapt to line and node failures in order to maintain a good grade of service for the network. Such policies base their decisions on the measured values, at given times, of a set of time varying variables (number of messages enqueued, number of hops, etc.) which describe the salient features of the state of the network (traffic, topology, etc.). Such information is referred to as routing information. A central node could provide the routing information (centralized control) and distribute it to all the nodes in the network, or the nodes could collaborate in computing the routing information directly (distributed control) [KLEI 64], [FULT 72], [KAHN 72].

Specifically, routing information is stored in tables at each node and is used to identify the output line for each destination. More detailed classifications of the routing policies can be found in [FULT 72], [GERL 73C] and [MCQU 74]. In this study, we limit our considerations to the distributed routing policies which base their decisions on routing information contained in routing tables

individually maintained at each node. The tables are updated periodically or asynchronously or a combination of both [FULT 72] using routing information collected internally and provided from neighboring nodes. Such a scheme is used to operate the ARPANET [MCQU 74].

Typically, in a network with N nodes, each node ("IMP" in the ARPANET terminology) i, (i = 1, 2, ..., N), has a routing table (to be denoted by RT) which is composed of N entries. Each entry, say k, is subdivided into three (or more) fields. The "delay" field indicates the estimated minimal delay from node i to node k. The "next-node" field indicates the next node a message must be forwarded to on its way to node k, using the "minimal" delay path. The "hop" field represents the minimum number of line hops to node k. The purpose of the hop-field is to allow the detection of line or node failures in the network.

Each node periodically (i.e., every .64 sec in the ARPANET, for a heavily loaded 50 kilobit line) sends to and receives update messages from neighboring nodes; these updates are not synchronized among nodes. Upon reception of an update, a node updates its own routing table, using the delays measured on its output lines and the delay information found in the update message. An example of an updating rule is provided in Chapter 5 and is used in a program developed to simulate computer networks.

To summarize, let us say that, fundamental to the operation of the distributed adaptive routing schemes is the storage, the propagation and the updating of routing tables. Also, it is important to note that in such schemes, the routing tables must contain a number of

entries equal to the number of nodes in the network.

For large distributed networks, on the order of hundreds or thousands of nodes, the adaptive routing policies in their present form become unsuitable. This comes about for the following reasons:

i. The excessive amount of storage required at each node to store the routing tables will substantially reduce the storage available for other functions such as store-and-forward, flow control and reassembly functions. Either that or extra nodal cost is incurred if more storage must be provided.

ii. The large CPU overhead required for the maintenance of the routing tables (frequent computations of the best routes) will induce either large delays in the forwarding of messages or higher nodal cost for faster CPU's.

iii. If no added capacity is provided, the channel capacity required for the transmission of the routing messages could become so demanding as to significantly reduce the throughput (load carried) of the network.

iv. The large delays incurred in the propagation of the routing information throughout the network will reduce the degree of adaptability of the routing scheme to changes in the network. In order to maintain a good response time in avoiding congested areas it is necessary to increase the rate of updates which, consequently, will require more CPU and channel capacity.

In view of the above considerations, routing schemes based on a hierarchical clustering of the network are proposed as solutions to keep the amount of routing information, hence the storage, the CPU and

the line overhead associated with it, within reasonable limits without deteriorating the performance of the network. With respect to (iv), because of the smaller routing tables, it is possible to increase the update rate without much extra cost. Fultz [FULT 72] and McQuillan [MCQU 74] proposed similar schemes but did not provide any quantitative analysis.

2.2 m-level Hierarchical Routing (MHR) Schemes

Because of the underlying hierarchical partitioning of the network, these schemes are referred to as the MHR schemes.

The main idea in reducing the table length (routing information) is to keep at any node, say i, complete routing information about nodes which are close to i (in terms of a hop distance or some nearness measure), and lesser information about nodes located further away from i. This can be realized by providing one entry per destination for the closer nodes, and one entry per set of destinations for the remote nodes. The size of this set may increase with the (average) distance from i to the nodes in the set. A similar concept underlies the mechanisms of large information systems with pyramidal structures, in which information is more and more aggregated as we move up to the higher levels in the hierarchical organization. Aggregation of information or variables is commonly introduced when dealing with large systems [MESA 70], [CHUR 68], [SCHO 71].

For the routing in large nets, the aggregation of the routing information is achieved through a hierarchical partitioning of the network. Basically, an m-level hierarchical clustering (MHC) of a set

of nodes consists of grouping the nodes (which we shall define as 0th level clusters) into 1st level clusters, which in turn are grouped into 2nd level clusters, etc. This operation continues in a bottom up fashion until the grouping of the m-2nd level clusters into m-1st level clusters whose union constitute the mth level cluster. The mth level cluster is the highest level cluster and as such it includes all the nodes of the network. The MHC will be described more formally in Section 2.3.1.

Since MHR schemes are based on an m-level hierarchical clustering of the nodes of the network, only <u>one</u> entry in the routing table, at any node, say i, is provided for each node in the same 1st level clusters as i, and for each 1st level cluster (set of nodes) in the same 2nd level cluster as i, and in general for each k-1st level cluster in the same k^{th} level cluster as i (k = 1, 2., ..., m). The structure of this scheme can be best understood by an example. Fig. 2.1 shows a 3-level hierarchical clustering imposed on a 24 node network. The clustering leads to the tree representation shown in Fig. 2.2, where nodes are identified using the Dewey notation [KNUT 69]. To each node is now associated a reduced routing table. Fig. 2.3 shows the entries of node 1.1.1's routing table; the number of entries is now 10 instead of 24 without clustering. As an example, the routing of a packet from node 1.1.1 to node 3.2.2 may proceed as follows: Node 1.1.1 recognizes from the address of the destination node, 3.2.2, that it has to use entry 3, of the 2nd level cluster entries, to decide upon the next node to which the packet must be forwarded. When the packet reaches a node, say 3.1.1, in the 2nd level cluster 3, then that node will in turn use the

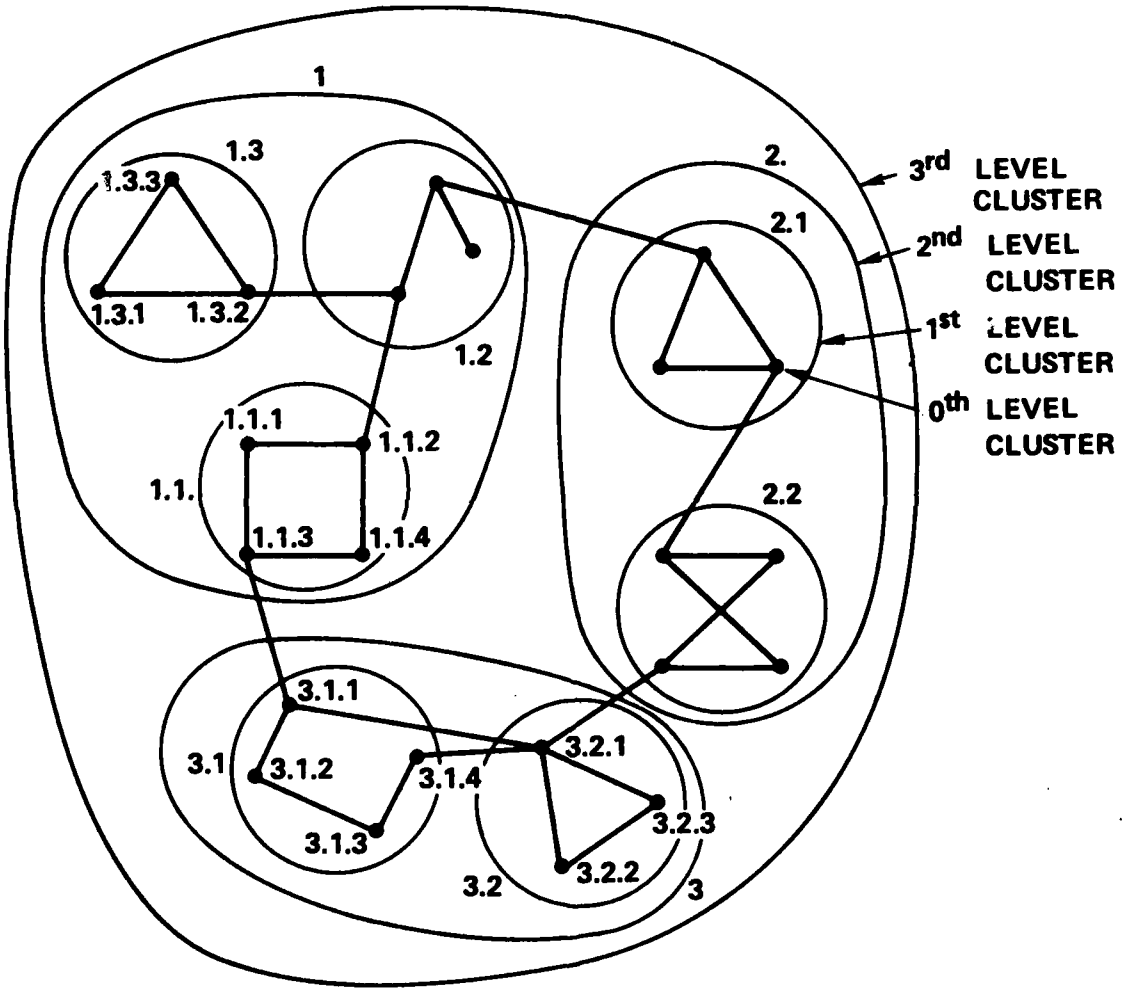


Figure 2.1. A 3-Level Clustered 24-Node Network.

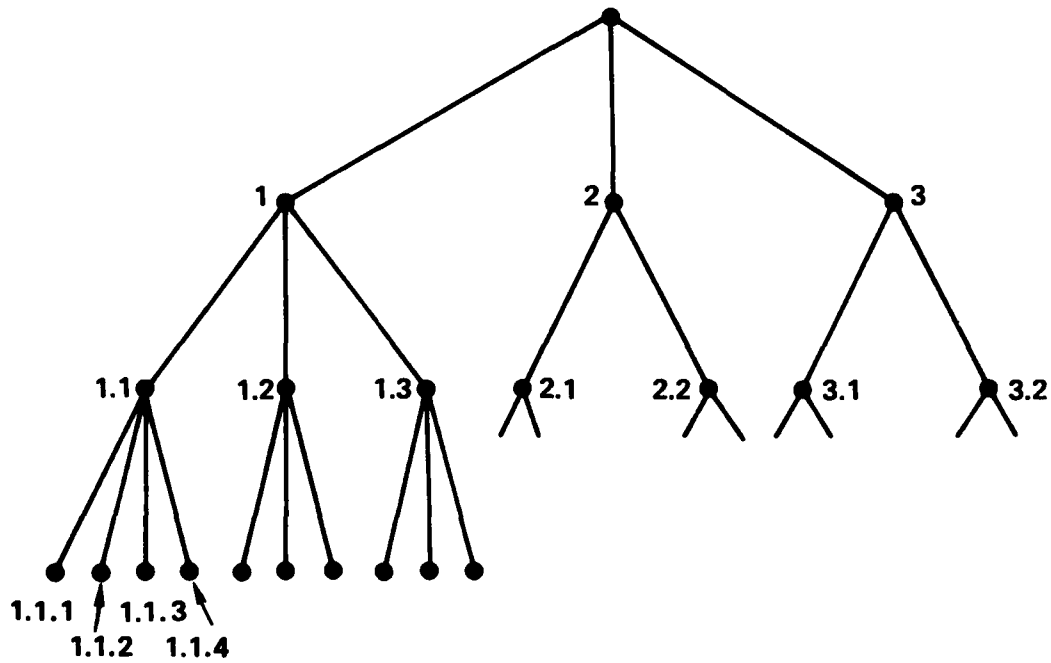


Figure 2.2. A Tree Representation of a 3-Level Clustered Net.

DESTINATION		NEXT NODE	DELAY	HOP NUMBER		
NODES IN SAME CLUSTER	1.1.1				*	0th LEVEL CLUSTER ENTRIES
	1.1.2					
	1.1.3					
	1.1.4					
CLUSTERS IN SAME SUPERCLUSTER	1.1.				*	1st LEVEL CLUSTER ENTRIES
	1.2.					
	1.3.					
SUPERCLUSTERS	1.				*	2nd LEVEL CLUSTER ENTRIES
	2.					
	3.					

* = SELF ENTRY

Figure 2.3. Routing Table of Node 1.1.1.

second entry among the 1st level cluster entries. Finally, when the packet enters the destination cluster, 3.2, the routing will be done using the 0th level cluster entry number 2. Notice that it was assumed that the MHC results in connected subgraphs.

From the above considerations, the following problems must be resolved at the outset:

i. The determination of an appropriate clustering structure, i.e., the size of the clusters at all levels and the number of levels

ii. The definition of an aggregate routing variable for the clusters at all levels; the specification of new updating rules, if necessary

iii. The assignment of the nodes (0th level clusters) to 1st level clusters, 1st level clusters to 2nd level clusters, etc.

The rest of this chapter will focus on (i). (ii) and (iii) are respectively treated in Chapters 3 and 5.

Finally, it is also necessary to evaluate the performance of the MHR schemes as compared with the present non-clustered schemes. This question will be examined in various ways in Chapters 3, 4, and 5.

2.3 Minimum Routing Information

The hierarchical partitioning of the network has as an objective the reduction of the size of the routing table. It is then important to determine the specific clustering structure (cluster sizes, number of levels) that will result in a minimum table length. The optimal sizes found will serve as the input parameters to the clustering techniques (see Chapter 5) whose function is to assign nodes to clusters,

clusters to superclusters, etc. In what follows, we will introduce the tree representation for the MHC and formally pose the problem of finding an optimal clustering structure. We will then proceed with the derivation of the optimal solution and the study of its characteristics.

2.3.1 Tree Representation for the MHC and Problem Statement

Any hierarchical classification scheme lends itself to a tree representation [KNUT 69]. The tree structure has already been introduced in Fig. 2.2, to represent the 3-level hierarchical clustering of the 24-node network in Fig. 2.1. The representation of an m-level hierarchical clustering is shown in Fig. 2.4.

2.3.1.1 Notation and Definitions

Definitions:

A kth level cluster, C_k, is defined recursively as a set of $k\text{-}1^{st}$ level clusters. It corresponds to a node at level k, in the tree representation of Fig. 2.4.

A kth level cluster is identified, similar to the Dewey notation [KNUT 69], by a vector of predecessors, $\underset{\sim}{i}_{k+1} = (i_m, i_{m-1}, \ldots, i_{k+1})$ which can subsequently serve as an address of C_k. The index, i_m, indicates the $m\text{-}1^{st}$ level cluster, say $C_{m-1}(i_m)$, to which C_k belongs; i_{m-1} indicates the $m\text{-}2^{nd}$ level cluster in $C_{m-1}(i_m)$ to which C_k belongs; etc. The notation, $C_k(i_m, i_{m-1}, \ldots, i_{k+1})$ or $C_k(\underset{\sim}{i}_{k+1})$, will be used when there is a need to identify C_k.

Notice that a leaf in the tree representation corresponds to a node (0th level cluster) in the network, and to any node is associated

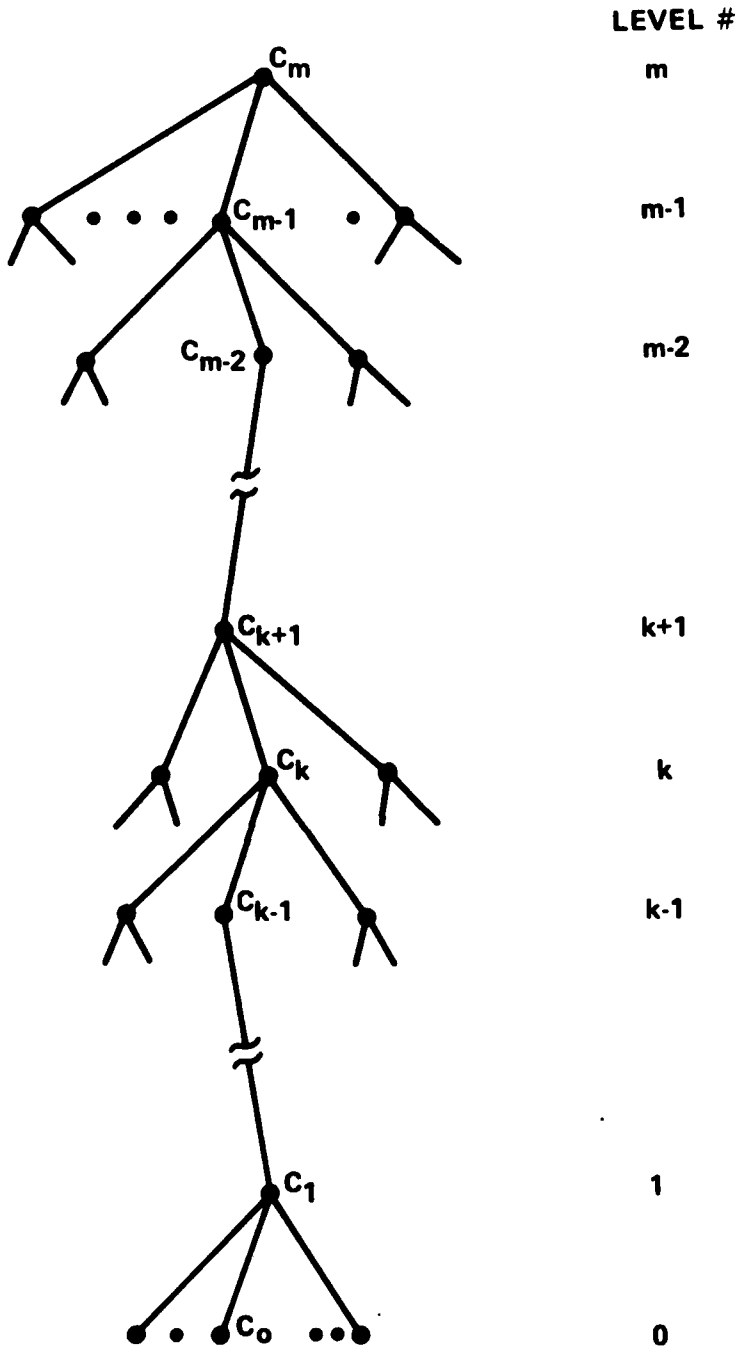


Figure 2.4. Tree Structure of the Hierarchical Clustering.

an address vector $\underset{\sim}{i}_1$ which will be used for the routing of messages.

The <u>degree</u> of a kth level cluster, C_k, is defined as the number of $k\text{-}1^{st}$ level clusters included in C_k. It also indicates the downward degree of the corresponding node in the tree.

<u>Notation and further definitions</u>

RT = Routing Table

MHC = m-level Hierarchical Clustering

MHR = m-level Hierarchical Routing

$S \triangleq$ Set of nodes in the network

N = size of the above set

m = number of levels in the hierarchy

C_k = k^{th} level cluster, represented by a node at level k in the tree structure

C_0 = 0th level cluster, network node

$\underset{\sim}{i}_{k+1} = (i_m, i_{m-1}, \ldots, i_{k+1})$ = vector of predecessors

$C_k(\underset{\sim}{i}_{k+1}) = C_k(i_m, \ldots, i_{k+1})$ = completely identified k^{th} level cluster

$n_k(\underset{\sim}{i}_{k+1}) = n_k(i_m, \ldots, i_{k+1})$ = degree of the above cluster

n_m = degree of the mth level cluster - (root of tree)

$\underset{\sim}{n}_k \triangleq \left\{n_k(\underset{\sim}{i}_{k+1})\right\}_{\underset{\sim}{i}_{k+1}}$ vector of degrees of all k^{th} level clusters

$\underset{\sim}{n} \triangleq (\underset{\sim}{n}_1, \underset{\sim}{n}_2, \ldots, \underset{\sim}{n}_m)$ degree vector

2.3.1.2 <u>Expressions for the length of the Routing Table and the Size Constraint</u>

The summation of the degrees of all the 1st level clusters

gives the total number of nodes in the network (i.e., the total number of leaves in the tree structure, Fig. 2.4). Hence

$$N = \sum_{i_m=1}^{n_m} \cdots \sum_{i_k=1}^{n_k(i_m, \ldots, i_{k+1})} \cdots \sum_{i_2=1}^{n_2(i_m, \ldots, i_3)} n_1(i_m, \ldots, i_2) \qquad (2.1)$$

Eq. (2.1) will generally serve as a constraint over the choice of the optimal degree vector $\underset{\sim}{n}$, and it will be referred to as the <u>size constraint</u>.

Let $\ell[C_0(\underset{\sim}{i}_1)]$ be the length of the RT at node $C_0(\underset{\sim}{i}_1)$; it is defined as the number of entries in that table. Then

$$\ell[C_0(\underset{\sim}{i}_1)] = \sum_{k=1}^{m} n_k(i_m, \ldots, i_{k+1}) \qquad (2.2)$$

The <u>assumption</u> is: each node of the network, $C_0(\underset{\sim}{i}_1)$, contains an RT with an entry for each $k\text{-}1^{st}$ level cluster in the same k^{th} level cluster as $C_0(\underset{\sim}{i}_1)$ (there are $n_k(i_m, \ldots, i_{k+1})$ such entries), and this for $k = 1, 2, \ldots, m$.

In order to simplify the manipulation of the RT's, we will also assume that <u>equal</u> length tables are provided at all nodes. Consequently, if ℓ is that length, it must accomodate the number of entries in the RT of any node.

Hence

$$\ell(m, \underset{\sim}{n}) \overset{\Delta}{=} \max_{\left\{\begin{matrix}\text{over all}\\ \text{nodes}\end{matrix}\right\}} \left\{ \sum_{k=1}^{n} n_k(i_m, i_{m-1}, \ldots, i_{k+1}) \right\} \qquad (2.3)$$

2.3.1.3 Problem Statement

$$\begin{cases} \text{given :} & N \\ \text{minimize :} & \ell(m, \underset{\sim}{n}) \qquad \text{[see Eq. (2.3)]} \\ \text{over :} & m \text{ and } \underset{\sim}{n} \\ \text{subject to :} & \text{size constraint} \qquad \text{[see Eq. (2.1)]} \\ & m \text{ a positive integer variable} \\ & \underset{\sim}{n} \text{ a vector of positive integer variables} \end{cases} \tag{2.4}$$

2.3.2 Real-Valued Solution of the Optimization Problem

We proceed to first solve Problem 2.4, with the assumption that $\underset{\sim}{n}$ is a real valued vector. This is in order to obtain an explicit analytical expression for the optimal solution. As a consequence of this assumption, a summation of the type $\sum_{i_2=1}^{n_2} n_1(i_2)$ becomes meaningless if n_2 is not at integer, unless all the $n_1(i_2)$'s are equal (to n_1) then the summation becomes, $n_2 n_1$. In fact, the solution of the optimization problem will show that clusters of the same level must be of equal degree, hence, all the summations in Eq. (2.1) will become meaningful a posteriori.

2.3.2.1 Optimality for a fixed m

Proposition 2.1

Given m, the number of levels in the hierarchy, and assuming that $\underset{\sim}{n}$ is a real valued vector, the solution of Problem 2.4 is such that:

(a) All clusters at all levels, $k = 1, \ldots, m$, are composed of the same number of lower level clusters,

$$\begin{cases} n_k(i_m, \ldots, i_{k+1}) = n_k & \forall\ (i_m, \ldots, i_{k+1}) \text{ and } k \\ n_k = N^{1/m} & k = 1, 2, \ldots, m \end{cases} \tag{2.5}$$

(b) With this optimum assignment the minimum table length is

$$\bar{\ell}(m) = mN^{1/m} \tag{2.6}$$

<u>Proof</u>:

The proof proceeds by induction on the number of levels, m. First, we start by showing that Proposition 2.1 is true for

For <u>$m = 2$</u>, the problem becomes:

$$\begin{cases} \min : \max\limits_{\substack{\text{over } i_2 \\ 1 \le i_2 \le n_2}} \{n_1(i_2) + n_2\} \\ \text{over} : \underset{\sim}{n}_1 = \{n_1(i_2)\}_{i_2}, \quad n_2 \\ \text{s.t.} : \sum\limits_{i_2=1}^{n_2} n_1(i_2) = N \\ \qquad\qquad \underset{\sim}{n}_1, n_2 \text{ positive.} \end{cases} \tag{2.7}$$

The formulation above is equivalent to

$$\begin{cases} \min : \eta \\ \text{over} : \eta, \underset{\sim}{n}_1, n_2 \\ \text{s.t.} : \eta \ge n_1(i_2) + n_2 \qquad \forall\ i_2 = 1, \ldots, n_2 \\ \qquad \sum\limits_{i_2=1}^{n_2} n_1(i_2) = N \qquad \underset{\sim}{n}_1, n_2 \text{ positive} \end{cases} \tag{2.8}$$

Let n_2 be fixed. Then, summing the first constraint in Problem 2.8, with respect to n_2, we find

$$n_2\eta \geq \sum_{i_2=1}^{n_2} n_1(i_2) + n_2^2 \tag{2.9}$$

If $\underset{\sim}{n}$ is feasible, i.e., if it satisfies Eq. (2.1), then Eq. (2.9) becomes

$$\eta \geq \frac{N}{n_2} + n_2 \qquad \forall\, \underset{\sim}{n} \text{ feasible.}$$

The above equation provides a lower bound on the optimal solution for a fixed n_2. Consequently, if a feasible solution achieves that lower bound, then it must be optimal. Such a solution is

$$n_1(i_2) = \frac{N}{n_2} \qquad i_2 = 1, 2, \ldots, n_2 \tag{2.10}$$

If we now let n_2 be a variable, the problem reduces to

$$\begin{cases} \min \eta = \dfrac{N}{n_2} + n_2 \\ \text{over } n_2,\ n_2 \geq 0 \end{cases}$$

whose optimum is achieved for $n_2 = N^{1/2}$ which, combined with Eq. (2.10) proves that Proposition 2.1 is true for $m = 2$.

Assuming that Proposition 2.1 is true for up to $m - 1$ levels, let us show that it is true for <u>m levels</u>. The size constraint, Eq. (2.1), is equivalent to the following set of constraints:

$$\begin{cases} \displaystyle\sum_{i_{m-1}=1}^{n_{m-1}(i_m)} \cdots \sum_{i_2=1}^{n_2(i_m, \ldots, i_3)} n_1(i_m, \ldots, i_2) = p(i_m) \qquad i_m = 1, \ldots, n_m & (2.11) \\ \displaystyle\sum_{i_m=1}^{n_m} p(i_m) = N & (2.12) \end{cases}$$

Let us <u>fix</u> the variables n_m and $p(i_m)$, $i_m = 1, \ldots, n_m$, such that Eq. (2.12) is satisfied. Problem 2.4 becomes decomposable into n_m subproblems, corresponding each to a given value of the index i_m. Such a subproblem, for a given i_m, is

$$\begin{cases} \text{min:} & \displaystyle\max_{\substack{\text{over all nodes} \\ \text{in same } m\text{-}1^{st} \\ \text{level cluster } C_{m-1}(i_m)}} \left\{ n_m + \sum_{k=1}^{m-1} n_k(i_m, i_{m-1}, \ldots, i_2) \right\} \\ \text{over:} & \underset{\sim}{n}_1, \underset{\sim}{n}_2, \ldots, \underset{\sim}{n}_{m-1} \\ \text{s.t.:} & \text{Eq. (2.11) must be satisfied.} \end{cases}$$

From the induction hypothesis, the solution of the above problem is such that

$$n_k(i_m, i_{m-1}, \ldots, i_{k+1}) = [p(i_m)]^{\frac{1}{m-1}} \qquad \forall\, (i_m, i_{m-1}, \ldots, i_{k+1}) \quad i_m \text{ fixed, and } \forall\, k = 1, 2, \ldots, m-1 \tag{2.13}$$

with such an assignment the minimum objective is equal to

$$(m - 1)[p(i_m)]^{\frac{1}{m-1}} + n_m .$$

If we now let the $p(i_m)$'s be variable but we keep n_m fixed, the problem becomes

$$\begin{cases} \text{min:} & \max\limits_{1 \le i_m \le n_m} \left\{ n_m + (m-1)[p(i_m)]^{m-1} \right\} \\ \text{over:} & p(i_m) \\ \text{s.t.:} & \text{Eq. (2.12) must be satisfied .} \end{cases}$$

The above problem is similar to Problem 2.7; hence, for a fixed n_m, the solution

$$p(i_m) = \frac{N}{n_m} \qquad i_m = 1, \ldots, n_m \tag{2.14}$$

is optimal. Consequently, we are left with

$$\begin{cases} \min \ell = (m-1) \left(\frac{N}{n_m}\right)^{\frac{1}{m-1}} + n_m \\ \text{over } n_m, \qquad n_m > 0 \end{cases}$$

Differentiating ℓ with respect to n_m, we find

$$\frac{d\ell}{dn_m} = 1 - N^{\frac{1}{m-1}} (n_m)^{-m/m-1}$$

From the above equation, we determine that ℓ is minimum for

$$n_m = N^{1/m} \tag{2.15}$$

and the minimum value of ℓ is

$$\bar{\ell} = mN^{1/m} \tag{2.16}$$

Substituting Eq. (2.15) into Eq. (2.14), then Eq. (2.14) into Eq. (2.13) we arrive at Eq. (2.5). Also, Eq. (2.16) is exactly equal to Eq. (2.6)

Q.E.D.

2.3.2.2 Global Optimality

So far we have solved for the optimal clustering when m, the number of levels is fixed. We now intend to let m vary and consequently solve for the global optimum. In other words, we intend to solve Problem 2.4 in its entirety except that the components of the degree vector $\underset{\sim}{n}$ and the number of levels m, are assumed to be real variables.

Proposition 2.2

Under the conditions of Proposition 2.1 and m being a real variable, the global optimum clustering is achieved for a number of levels

$$m_* = \ln N \tag{2.17}$$

and a degree vector n*

$$n_k^* = e = 2.718... \qquad k = 1, 2, ..., m_* \tag{2.18}$$

The corresponding minimum table length is

$$\ell_* = e \ln N \tag{2.19}$$

Proof:

Proposition 2.1 gave us the optimal clustering, for a fixed m. Consequently we are left with the minimization of $\overline{\ell}(m)$, Eq. (2.6), with respect to m. Differentiating Eq. (2.6) with respect to m, we find

$$\frac{d\bar{\ell}(m)}{dm} = \left(\frac{m - \ln N}{m}\right) N^{1/m}$$

The above equation shows that $\bar{\ell}(m)$ is minimum for m as given in Eq. (2.17). Substituting Eq. (2.17) into Eqs. (2.5) and (2.6), we arrive at Eqs. (2.18) and (2.19).

2.3.2.3 Duality

A simpler formulation of problem 2.6 may be obtained by directly imposing that all k^{th} level clusters are of equal degree, n_k $(k = 1, \ldots, n)$. Hence Problem 2.6 becomes

$$\begin{cases} \text{given:} & N \\ \text{min:} & \ell = \sum_{k=1}^{m} n_k \\ \text{over:} & \underset{\sim}{n} = (n_1, \ldots, n_m), \quad m \\ \text{s.t.:} & \prod_{k=1}^{m} n_k = N \\ & \underset{\sim}{n}, m \quad \text{positive integer valued.} \end{cases} \tag{2.20}$$

It can be (directly) shown by induction, that the real valued solution of the above problem verifies Propositions 2.1 and 2.2.

The dual formulation of Problem 2.20 is

$$\begin{cases} \text{given:} & \ell \\ \text{max:} & N = \prod_{i=1}^{m} n_k \\ \text{over:} & \underset{\sim}{n}, m \\ \text{s.t.:} & \sum_{k=1}^{m} n_k = \ell \\ & \underset{\sim}{n}, m \quad \text{positive interger valued.} \end{cases} \tag{2.21}$$

The objective, in the formulation above, is to find the maximum number of nodes, N , such that there exists an MHC whose application results in a routing table of length ℓ. The dual propositions to 2.1 and 2.2 are respectively,

Proposition 2.3

For a fixed m and ℓ, the real valued solution of Problem 2.21 is such that

$$n_k = \frac{\ell}{m} \qquad k = 1, 2, \ldots, m$$

with this assignment

$$N = \left(\frac{\ell}{m}\right)^m$$

Proposition 2.4

The real valued global optimum of Problem 2.21 is such that

$$\begin{cases} m_* = \dfrac{\ell}{e} \\ n_k^* = e \qquad k = 1, \ldots, m_* \\ N^* = e^{\ell/e} \end{cases}$$

Proposition 2.3 can be directly proven by induction on the number of levels in the hierarchy. Proposition 2.4 is a direct consequence of Proposition 2.3.

2.3.2.6 Numerical examples

Figs. 2.5 and 2.6, respectively, illustrate the behavior of $\bar{\ell}/N$ and $\bar{\ell}/\ell_*$ (see Eqs. (2.6) and (2.19) with respect to m and for several values of N. The plots exhibit a flat area around the minimum.

They also show an initial fast decrease of $\bar{\ell}$ toward a value close to the minimum. This last property is better illustrated in Fig. 2.7 where $\bar{\ell}/N$ is plotted with respect to N for $m \in \{1, 2, \ldots, \ln N\}$; this indicates that most of the table reduction can be obtained with hierarchical clustering whose number of levels is quite a bit smaller than m_* (Eq. (2.17)).

2.3.3 Integer Solution

In this paragraph we intend to solve the integer optimization problem as formulated in 2.20, except for the size constraint which is changed to an inequality, the problem becomes

$$\begin{cases} \text{min:} \quad \ell = \sum_{k=1}^{m} n_k \\ \text{over:} \quad \underset{\sim}{n},\ m \quad \text{integer valued} \\ \text{s.t.:} \quad \prod_{k=1}^{m} n_k \geq N \end{cases} \tag{2.22}$$

The above modification is introduced to avoid dealing with empty feasible sets of vectors $\underset{\sim}{n}$, for some values of m and N. A solution $\underset{\sim}{n}$, such that $\prod_{k=1}^{m} n_k > N$, only implies that the degree of any k^{th} level cluster, $k = 1, \ldots, m$, must be less than or equal to n_k. Practically this means that there will be unused entries in some of the routing tables.

Recall that the global optimum real-valued solution is such that all the components n_k's are equal to e, therefore in the integer case we are not surprised that the following proposition holds true.

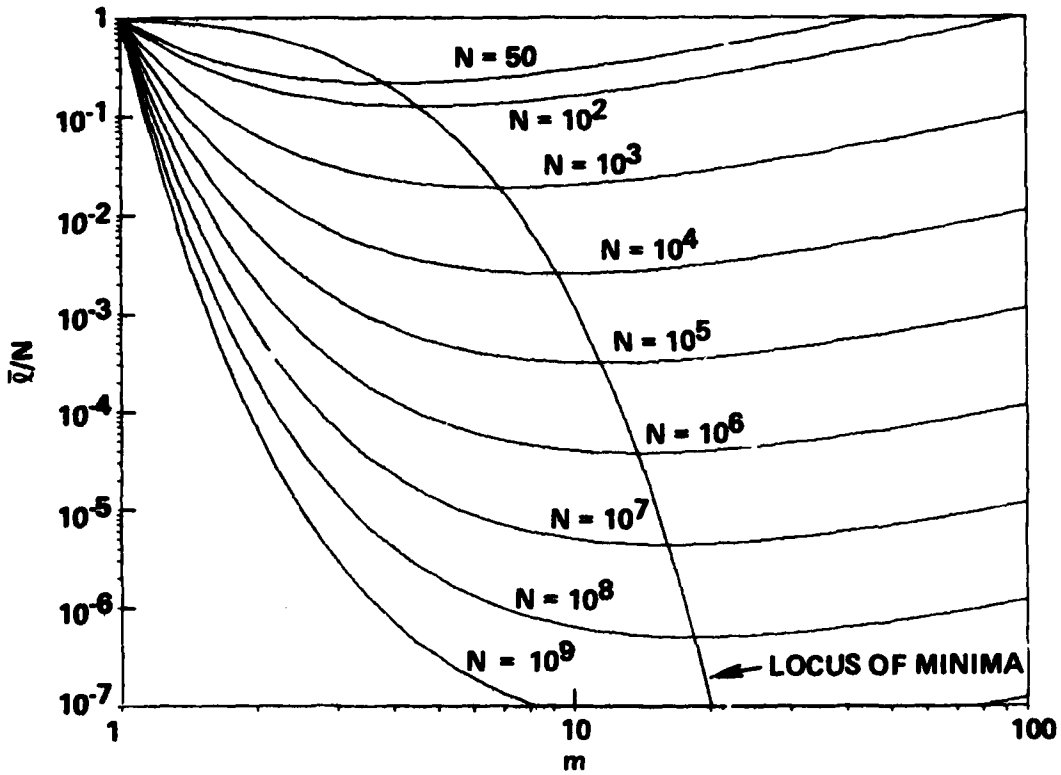


Figure 2.5. Minimum Relative Table Length, ℓ/N, Given m.

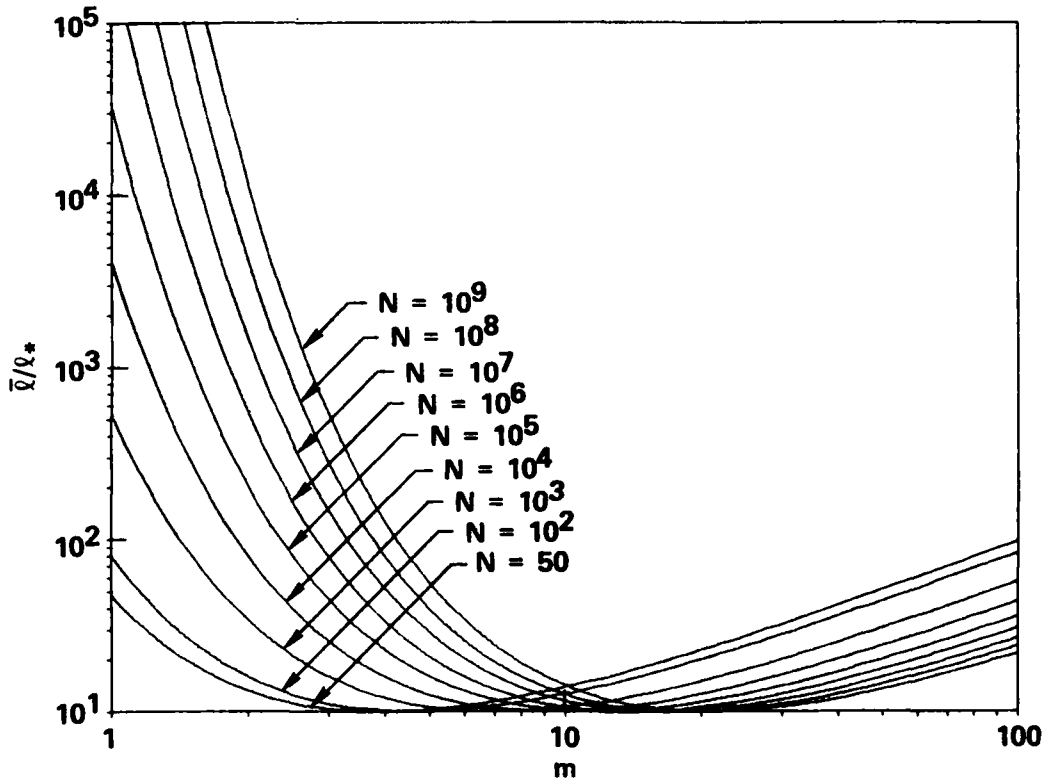


Figure 2.6. Ratio of Table Lengths at Optimality Given m, and at Global optimality, ℓ/ℓ_*.

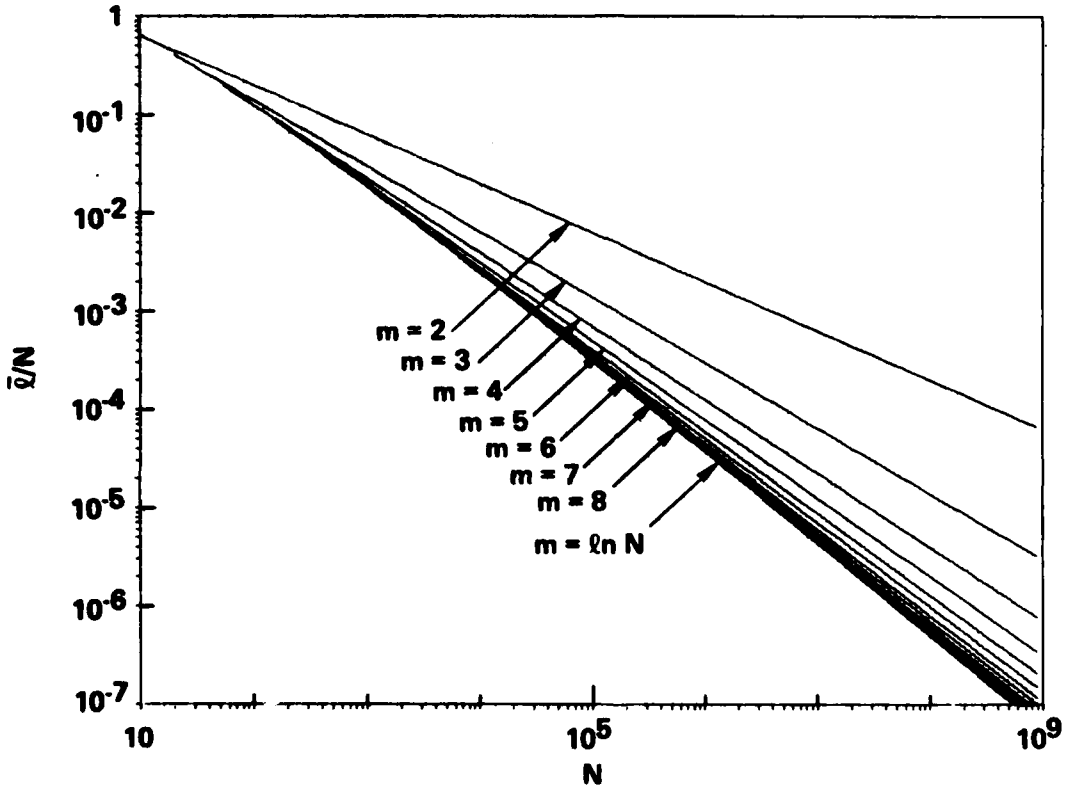


Figure 2.7. Minimum Relative Table Length, $\bar{\ell}/N$, versus the Number of Nodes.

Proposition 2.5

There exists a global optimum vector $\underset{\sim}{n}^*$ which is composed of zero, one or two components equal to 2, with all the others equal to 3.

Proof

The idea[1] is that any number (component of $\underset{\sim}{n}$) or set of numbers can be replaced by a set of 2's or 3's which results in the same sum but a higher product. Hence the new set is at least as good as the original. As an example, we list typical transformations.

Original Numbers	Transformation	Sum	Original Product	New Product
4	2, 2	4	4	4
5	2, 3	5	5	6
6	3, 3	6	6	9
7	2, 2, 3	7	7	12
2, 2, 2	3, 3	6	8	9

Let us now proceed with the proof by showing that we can always derive from an optimal solution which does not satisfy Proposition 2.5, one which does. Let $\underset{\sim}{n}^*$, m_* be a solution which does not satisfy Proposition 2.5; then, at least one of the following conditions is true: (1) at least one component is equal to 1, (2) at least one component is greater than 3, finally (3) at least three components are equal to 2.

With respect to (1) let us prove by contradiction that this situation cannot occur. Assume that $n_{m_*} = 1$, then consider the

[1] This idea was suggested by Dr. D. Cantor, Mathematics Department, UCLA.

vector $\underset{\sim}{n}'$ composed of the other $m_* - 1$ components of $\underset{\sim}{n}_*$, i.e.,

$$n'_k = n_k \qquad k = 1, \ldots, m_* - 1$$

consequently
$$\sum_{k=1}^{m_*-1} n'_k = \sum_{k=1}^{m_*} n_k - 1$$

and
$$\prod_{k=1}^{m_*-1} n'_k = \prod_{k=1}^{m_*} n_k$$

Hence $\underset{\sim}{n}'$ is a feasible vector which results in a better objective; that is a contradiction.

<u>With respect to (2)</u>, assume first that $n_{m_*} \geq 5$, then consider the vector $\underset{\sim}{n}'$ composed of $m_* + 1$ elements which are such that

$$\begin{cases} n'_k = n_k & k = 1, \ldots, m_* - 1 \\ n'_{m_*} = 3 \\ n'_{m_*+1} = n_{m_*} - 3 \end{cases}$$

With the above transformation the objective function is obviously unchanged. The product is such that

$$n'_{m_*} n'_{m_*+1} = n_{m_*} + 1 + 2n_{m_*} - 10$$

hence
$$n_{m_*} \geq 5 \Rightarrow n'_{m_*} n'_{m_*+1} \geq n_{m_*} + 1$$

$$\prod_{k+1}^{m_*+1} n'_k = \left(\prod_{k=1}^{m_*-1} n_k\right) n'_{m_*} n'_{m_*+1} > \prod_{k=1}^{m_*} n_k$$

which means that $\underset{\sim}{n}'$ is also feasible, hence optimal.

If $n_{m_*} = 4$ then we choose $n'_{m_*} = n'_{m_*+1} = 2$ and still obtain a feasible vector $\underset{\sim}{n}'$. The above operations can be repeated several times on the constructed vectors $\underset{\sim}{n}'$ in order to transform all the

components which are greater than 3.

<u>With respect to (3)</u>, assume that $n_{m_*-2} = n_{m_*-1} = n_{m_*} = 2$; then consider the vector $\underset{\sim}{n}'$ composed of $m_* - 1$ elements which are such that,

$$\begin{cases} n'_k = n_k & k = 1, \ldots, m_* - 3 \\ n'_{m_*-2} = n'_{m_*-1} = 3 \end{cases}$$

With the above transformation the objective fuction is unchanged. The product of the components is increased by 1. As a result $\underset{\sim}{n}'$ is also feasible, hence optimal.

The above operation can be repeated several times on the constructed vectors $\underset{\sim}{n}'$, until we reach a solution with no more than two 2's.

Finally, we can see that the repeated application of the transformations shown for (2) and (3) will eventually generate a vector $\underset{\sim}{n}'$ which satisfies Proposition 2.5.

As a consequence of Proposition 2.5 the search for the optimal number of levels is reduced to three possibilities. From Problem 2.22, the optimal m, must be such that

$$3^{m-x}2^x \geq N$$

where $$x \in \{0, 1, 2\}$$

Hence, the three possible values of m are:

1. $x = 0 \Rightarrow m_0 = \left\lceil \frac{\ln N}{\ln 3} \right\rceil$

2. $x = 1 \Rightarrow m_1 = \left\lceil \frac{\ln N/2}{\ln 3} \right\rceil + 1$

3. $x = 2 \Rightarrow m_2 = \left\lceil \frac{\ln N/4}{\ln 3} \right\rceil + 2$

Finally the optimal m, m_* , is the solution of

$$\begin{cases} \min : & \ell = 3n - x \\ \text{over} & (m,x) \in \{(m_0, 0), (m_1, 1), (m_2, 2)\} \end{cases}$$

Notice that the optimal pair (m,x) gives the composition of the optimal vector $\underset{\sim}{n}^*$.

<u>Proposition 2.6</u>

Given m, there exists an optimal vector $\underset{\sim}{n}$ which is such that no two components differ by more than 1, and which is given by,

$$\begin{cases} n_m = \lceil N^{1/m} \rceil \\ n_k = \left\lceil \left(\dfrac{N}{\prod_{i=k+1}^{m} n_i} \right)^{1/k} \right\rceil \quad k = 2, 3, \ldots, m \end{cases} \tag{2.23}$$

or any other permutation of the above solution.

<u>Proof</u>

Let us show that, given any two numbers which differ by more than 1, we can replace them by <u>exactly</u> two numbers which do not differ by more than 1 and which result in the same sum but in a better product. Two cases to consider depend on whether the difference between the two numbers is odd or even:

1/ (even), pick n and n + 2p.

Their sum is 2n + 2p.

Their product is $n^2 + 2pn$.

Let us replace them by n + p and n + p; then the sum is still the same, but the product is

$$n^2 + 2pn + p^2$$

which represents an improvement of p^2. A similar proof can apply for the odd case.

From the above property, we conclude that any n_k $k = 1, \ldots, m$, is equal to a given number, either a or a + 1. *If we let x represent* the number of components equal to a + 1, then the problem reduces to

$$\begin{cases} \min \ell = (m - x)a + x(a + 1) = ma + x \\ \text{over} \quad (a, x) \\ \text{s.t.} \quad a^{m-x}(a + 1)^x \geq N \end{cases}$$

a positive integer; $x \leq m$, positive integer.

Let us show that there exists at least one component, say n_m, equal to $N^{1/m}$. From the constraint above, the optimal a is such that

(i) $x = 0 \Rightarrow a^m \geq N \Rightarrow a = \lceil N^{1/m} \rceil$

(ii) $x \neq 0 \Rightarrow (a + 1)^m \geq (a + 1)^x a^{m-x} \geq N \Rightarrow a + 1 = \lceil N^{1/m} \rceil$

Knowing that $n_m = \lceil N^{1/m} \rceil$, Problem 2.22 can be reduced to m - 1 variables, with N replaced by N/n_m. From the same considerations as above, we know that the optimal vector $(n_1, n_2, \ldots, n_{m-1})$ for the new problem has at least one component, say n_{m-1}, equal to

$$\left\lceil \left(\frac{N}{n_m}\right)^{\frac{1}{m-1}} \right\rceil$$

Repeating the same operation, we arrive at Eq. (2.23).

Numerical examples

Similar curves as in Fig. 2.5 are plotted in Fig. 2.8. They

illustrate the behavior of $\bar{\ell}$ (ℓd) versus m for the integer case. They exhibit the properties described in Section 2.3.2.4, mainly the enormous table reduction for small values of m. By comparing Fig. 2.5 with Fig. 2.8 we notice that the integer valued solution is extremely close to the real-valued solution. Consequently, we will limit any further considerations to the simple real-valued solution.

Fig. 2.9 illustrates the behavior of N_d/N, where

$$N_d = \prod_{k=1}^{m} n_k$$

Notice that the slack in the size constraint is relatively small for large values of N.

2.3.4 Optimality with No "Self-Entries" in the Routing Table

In the previous model, at each routing table, one entry (to be called self-entry) is reserved for the node which contains that table, and one for each of the kth level clusters k = 1, 2, ..., m - 1, to which that node belongs. For some MHR schemes (e.g., those defined in Chapter 3) and/or with some extra CPU overhead, the updating algorithm can operate without those self-entries. Consequently, the new length ℓ' of the RT's is,

$$\ell' = \ell - m \tag{2.24}$$

where ℓ is given by Eq. (2.3).

The optimal clustering structure is the solution of Problem 2.4 where ℓ is replaced by ℓ'.

Real Valued Solution

For a fixed m, Eq. (2.5) still holds true. Hence the minimum

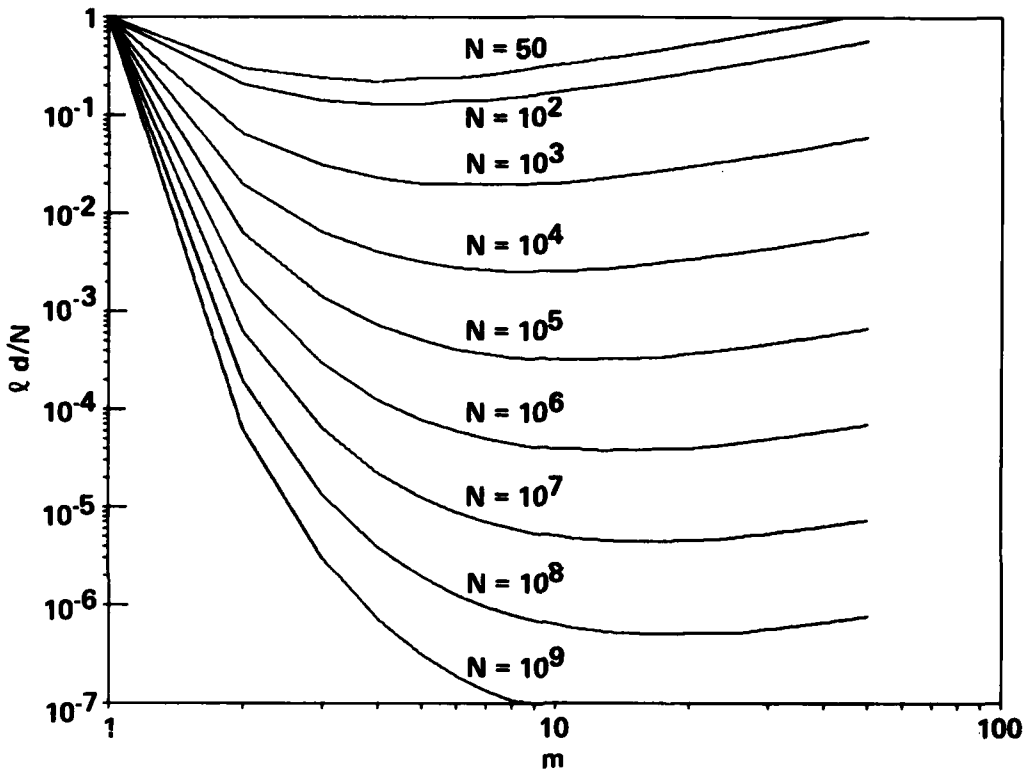


Figure 2.8. Integer Minimum Relative Table Length, Given m.

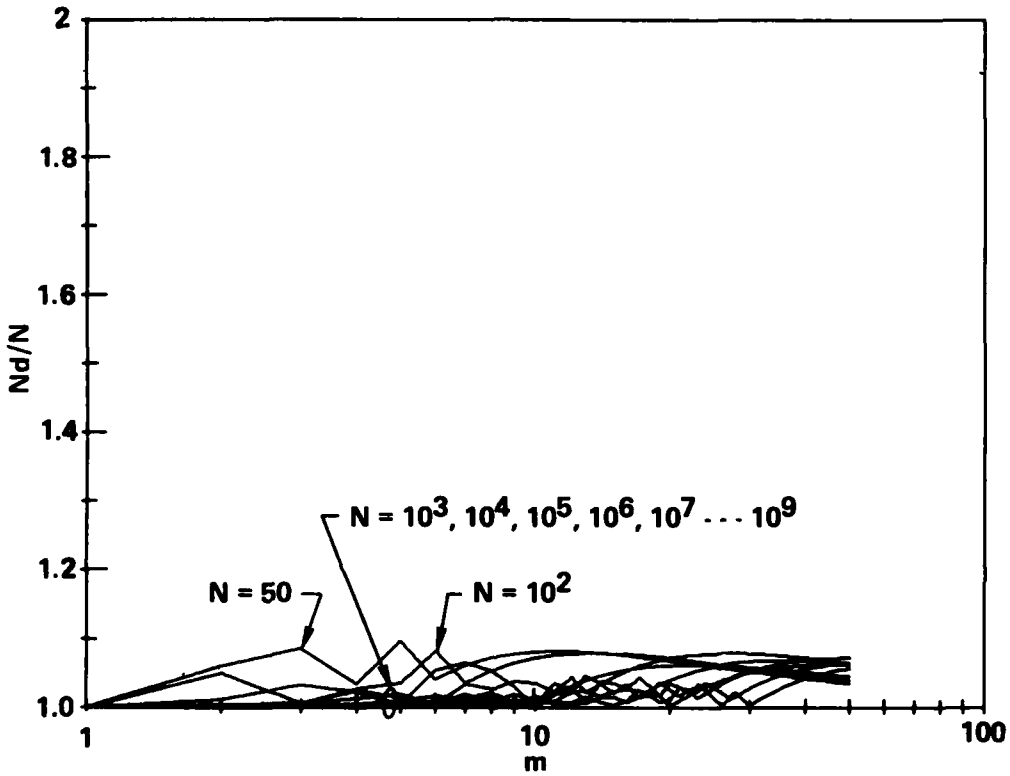


Figure 2.9. Slack in Size Constraint With the Integer Optimization.

length is

$$\overline{\ell}' = mN^{1/m} - m .$$

In order to find the global optimum, let us differentiate $\overline{\ell}'$ with respect to m,

$$\frac{d\ell}{dm} = \left(1 - \frac{\ln N}{n}\right) N^{1/m} - 1.$$

Taking the 2nd derivative, we find

$$\frac{d^2\overline{\ell}'}{dm^2} = \frac{(\ln N)^2}{m^3} \; N^{1/m} .$$

For $m > 0$, $\frac{d^2\overline{\ell}'}{dm^2}$ is positive; hence $\frac{d\ell'}{dm}$ is a non-decreasing function of n. Consequently it is not greater than its value at $m = \infty$, which is equal to zero. We conclude that ℓ' is a decreasing function of m and that it reaches its minimum ℓ_*', for m equal to infinity.

$$\ell_* = \lim_{m \to \infty} \; n \, (N^{1/m} - 1) = \ln N$$

In summary, the global optimum is such that

$$\begin{cases} m_*' = + \infty \\ \ell_*' = \ln N \\ n_k = 1 \qquad k \geq 1 \end{cases} .$$

The above result is to be compared with Eq. (2.19),

$$\ell_* = e \, \ell_*'$$

which indicates that theoretically, an improvement of a fraction $\frac{1}{e}$, of the global minimum length can be obtained. These limiting results are, however, meaningless in the integer case.

Integer Valued Solution

For a fixed m, Proposition 2.6 still holds true. As for the global optimum the problem to solve is,

$$\begin{cases} \text{min: } \sum_{k=1}^{m} n_k - m \\ \text{over: } n,\ m \qquad \text{integer valued} \\ \text{s.t.: } \prod_{k=1}^{m} n_k \geq N \end{cases} \tag{2.25}$$

Recall that the real valued solution is such that

$$n_k = \lim_{m \to \infty} N^{1/m} = 1^+$$

Therefore we are not surprised that the following proposition holds true.

Proposition 2.7

There exists a non-degenerate (i.e., no one component is equal to one) global optimum vector $\underset{\sim}{n}^*$ which is such that

$$n_k^* = 2 \qquad k = 1, 2, \ldots, m_* \tag{2.26}$$

and

$$m_* = \left\lceil \frac{\ln N}{\ln 2} \right\rceil \tag{2.27}$$

Proof

Let us first show that if $\underset{\sim}{n}^*$ is an optimal vector which contains at least one component equal to 1, then the vector obtained after deletion of all the 1's is also optimal.

Assume that $n_{m_*} = 1$, then consider $\underset{\sim}{n}'$ such that

$$n_k' = n_k \qquad k = 1, \ldots, m_* - 1 .$$

Notice that the table length and the product of the components are unchanged. Thus $\underset{\sim}{n}'$ is also feasible, hence it is optimal. The above operation can be repeated several times until we eliminate all the components which are equal to one. In the sequel we restrict our considerations to non-degenerate solutions (i.e., no one component is equal to one).

Let ust first prove this intermediary result.

Fact:

If $\underset{\sim}{n}^*$ is a non-degenerate optimal vector then it must be composed of zero, one or two components equal to 3 with all the others equal to 2.

Proof

The proof proceeds by contradiction. Assume that $\underset{\sim}{n}_*$ is a non-degenerate optimal vector which does not satisfy the above fact. It must be such that at least one of the following conditions is true; (1) at least one component of $\underset{\sim}{n}^*$ is greater than or equal to 4; (2) at least three components of $\underset{\sim}{n}^*$ are equal to 3.

With respect to (1), assume that $n_{m_*} \geq 4$, then consider the vector $\underset{\sim}{n}'$.

$$\begin{cases} n'_k = n_k & k = 1, 2, \ldots, m_* - 1 \\ n'_i = 2 & i = m_*, m_* + 1, \ldots, m_* + m_0 \end{cases} \tag{2.28}$$

where m_0 is such that

$$2^{m_0} < n_{m_*} \leq 2^{m_0+1} \tag{2.29}$$

Thus
$$\prod_{k=1}^{m_*+m_0} n'_k = 2^{m_0+1} \prod_{k=1}^{m_*-1} n_k \geq \prod_{k=1}^{m^*} n_k$$

Consequently $\underset{\sim}{n}'$ is feasible. Also the objective function is

$$\sum_{k=1}^{m_*+m_0} n_k' - (m_* + m_0) = \sum_{k=1}^{m_*-1} n_k + 2(m_0 + 1) - (m_0 + m_*).$$

If we subtract the above table length from the one obtained with $\underset{\sim}{n}^*$, then the difference is

$$\Delta = n_{m_*} - m_0 - 2 \tag{2.30}$$

If $\Delta > 0$, then we reach a contradiction. Let us prove that in fact $\Delta > 0$.

i. $n_{m_*} = 4 \Rightarrow m_0 = 1 \Rightarrow \Delta = 1$

ii. $n_{m_*} \geq 5 \Rightarrow m_0 \geq 2$.

It can easily be shown by induction that

$$2^{m_0} > m_0 + 1 \qquad \forall\, m_0 \geq 2$$

From Eqs. (2.29) and (2.30),

$$\Delta > 2^{m_0} - m_o - 2 \geq 2^{m_0} - m_o - 1 > 0$$

Thus Δ is always greater than zero, which is a contradiction.

<u>With respect to (2)</u>, assume that

$$n_{m_*-2} = n_{m_*-1} = n_{m_*} = 3$$

consider the vector $\underset{\sim}{n}'$ composed of $m_* + 2$ elements,

$$n_k' = \begin{cases} n_k & k = 1, \ldots, m_* - 3 \\ 2 & k = m_* - 2, \ldots, m_* + 2 \end{cases}$$

It is obvious that $\underset{\sim}{n}'$ is feasible and that it reduces the table length by one, which contradicts the fact that n^* is optimal. Q.E.D.

End of Proof of Proposition 2.7

The rest of the proof is a consequence of the above fact. Let $\underset{\sim}{n}_*$ be a non-degenerate optimal vector. If $\underset{\sim}{n}_*$ does not satisfy Proposition 2.7, then, because of the above fact, it must contain one or two 3's. Any 3 can be replaced by two 2's, without changing the objective function and still keeping feasibility. As a result we arrive at a vector composed only of 2's and for that vector to be optimal the number of components must be a solution of

$$\begin{cases} \text{min: } m \\ \text{over: } m \quad \text{integer} \\ \text{s.t.: } 2^m \geq N \end{cases}$$

The solution of the above problem is given by Eq. (2.27), which terminates the proof.

2.4 Conclusions

This chapter was primarily concerned with the introduction of the MHR schemes and their underlying hierarchical clustering structure as solutions to the reduction of the routing information and its associated overhead. We found, indeed, that enormous gains can be obtained whereby the length of the routing tables may be reduced from N entries to the order of $e \cdot \ln(N)$ entries. However, a shortcoming of these gains is the increase in the path length of a message in the network. This comes about from the fact that a given node must send all its traffic to a given cluster, on the same path to that cluster. This path will, in general, be optimal only for a subset of the nodes in the destination cluster. Consequently, some messages will follow longer

paths than they should. This issue will be addressed in the next chapter.

It is also possible that less routing adaptability could result from the MHR schemes because of the aggregation of the routing information. This fact may, however, be beneficial in our context of large networks where the routing policy need not adjust to very remote and probably short lived fluctuations.

CHAPTER 3

STATIC EVALUATION OF HIERARCHICAL ROUTING: BOUNDS ON THE INCREASE IN NETWORK PATH LENGTH

Built into the hierarchical routing, proposed in Chapter 2, is the reduction of routing information. As a result there will be, in general, an increase in the network path length. The magnitude of the increase is closely related to the length of the routing table and to the choice of a specific aggregate routing variable.

In this chapter, two hierarchical routing schemes and a non-clustered (non-hierarchical) scheme are presented, and their routes are characterized under some "equilibrium" conditions. Then, with some assumptions on the network topology and on the partitioning structure, bounds on the increase in the network path length are derived. These bounds are expressed and studied in terms of the relative table length (ℓ/N). They demonstrate an asymptotic result for a class of large networks. The result shows that when the number of nodes grows to infinity, enormous table reduction can be obtained with a relatively insignificant increase in the network path length.

3.1 Path Characteristics for Hierarchical and Non-Hierarchical Adaptive Routing Policies

The purpose of this section is to characterize the actual or virtual routes obtained from the routing tables under certain equilibrium conditions as defined below. The routing schemes are assumed to belong to the class of hierarchical or non-hierarchical adaptive

policies, considered in Chapter 2. Such policies basically propagate routing information describing the length of the paths to reach any destination node or a set of nodes. The path length is defined as the sum of the lengths of all the channels which constitute that path. Moreover, the length of a given channel is usally a random variable which may reflect the utilization and/or the excess capacity and/or any other information which partly or entirely describes the stochastic state of that channel. The transient nature of adaptive routing renders the analysis of the above problem extremely complicated. In order to make any progress we will assume that all channels are of constant length. This is a simplifying assumption which will, however, allow us to capture the effect of clustering on the network path length; this is the main objective of this chapter. Morever the above assumption is an accurate description of routing policies which are only sensitive to changes in the network topology, and of more general policies operating under light traffic conditions [KLEI 64]. Furthermore if all the channels are considered to be of equal length (say 1), then the routing information is simply what we defined earlier as the hop distance. Such routing information is in general utilized by routing policies to, at least, detect changes in the network topology.

In summary we will restrict our considerations to hierarchical or non-hierarchical routing schemes (also referred to as clustered and non-clustered routing schemes) which use as routing information the path length only. Also we consider that all channels are of constant length. In what follows we first assume that all channels are of equal

length (one hop) and then we generalize to arbitrary (constant) length channels.

3.1.1 Non-Clustered Routing (NCR)

Recall that the NCR schemes operated with complete routing information, i.e., the routing tables contain one entry per destination node in the network. Because of the frequent exchange of routing information between neighboring nodes, the routing tables will in general indicate the next node in the shortest path to any destination. This is due to the fact that at each update, a node compares its own routes to the neighbor's routes and updates its table by keeping the shorter routes. The above approach is in essence a dynamic search for the minimal paths in the network [HU 69], [FRAN 71]. As a consequence, after a certain number of updates (less than or equal to the diameter of the network) if no changes occur in the network topology, all the tables will indicate exactly the shortest paths. If a change in topology occurs, then some of the paths will not be optimal during a transitory period (time for the routing information to percolate to the concerned areas in the network). However for the purpose of comparison with the hierarchical routing, we will assume that the tables always carry the shortest path information. A more formal treatment of the above is presented within the context of the clustered schemes since, as we will see in the sequel, a 1-level hierarchical routing is equivalent to an NCR scheme.

3.1.2 Specification of the MHR Schemes

Built into the MHR schemes, considered in Chapter 2, is the reduction of the routing information whereby one entry in a routing table may be reserved for more than one destination node. Routing information is aggregated whenever it is exchanged between special nodes in different clusters at any level. Such special nodes will be referred to as exchange nodes. Two MHR schemes will be presented below. They differ only in the definition and subsequently the computation of the aggregate routing information. The two schemes will be referred to as the Closest Entry Routing (CER) and the Overall Best Routing (OBR) schemes. In order to proceed with their description, we first need to specify the underlying m-level hierarchical partitioning of the set of nodes of the network.

Assumption 3.1

The underlying MHC structure of the set of network nodes is such that all clusters at the same level k, are of equal degree, n_k, $k = 1, \ldots, m$. Also the subset of nodes composing a cluster at any level and their incident channels constitute a 1-connected (at least one path exists between any pair of nodes) cluster subnetwork.

The former property of the above assumption partly satisfies Proposition 2.1 which defines the optimal clustering structure that we will eventually use. The latter property is necessary, since the traffic exchanged between nodes in the same cluster must follow paths included in that cluster's subnet.

Because of the above assumption the notation of Chapter 2

can be greatly simplified. Particularly, the degree vector is reduced to $\underset{\sim}{n} = (n_1, n_2, \ldots, n_m)$. Moreover, if there is no need to identify a cluster with its entire address vector, then a shorter notation, such as below, can be used.

$$C_k(j) \triangleq j^{th} \text{ entry in } k^{th} \text{ level cluster entries}$$

or

$$C_k(s) \triangleq k^{th} \text{ level cluster containing an arbitrary node } s.$$

As a consequence of Assumption 3.1, the routing tables at any node will have the format shown in Fig. 3.1, and will contain $\ell = n_1 + n_2 + \ldots + n_m$ entries. Notice that self entries are included in the routing table (RT). The self entries of the RT at an exchange node may be assigned to carry the aggregate routing information from one cluster to another. The content of the self entries in tables at other nodes (non-exchange nodes) need not be specified in this study. Two aggregation procedures, each pertaining to a particular MHR scheme (OBR or CER), are presented below. Also, possible ways of implementing those procedures are suggested with the assumption that neighboring nodes exchange their RT's as routing updates (as mentioned in Chapter 2).

Aggregation of Routing Information

i. With respect to the CER (Closest Entry Routing) scheme, no routing information describing the internal behavior of a cluster is propagated outside the cluster. With this rule, a cluster is regarded from the outside as a single (super-) node whose distance to itself is equal to zero. A way of implementing this scheme is to assign

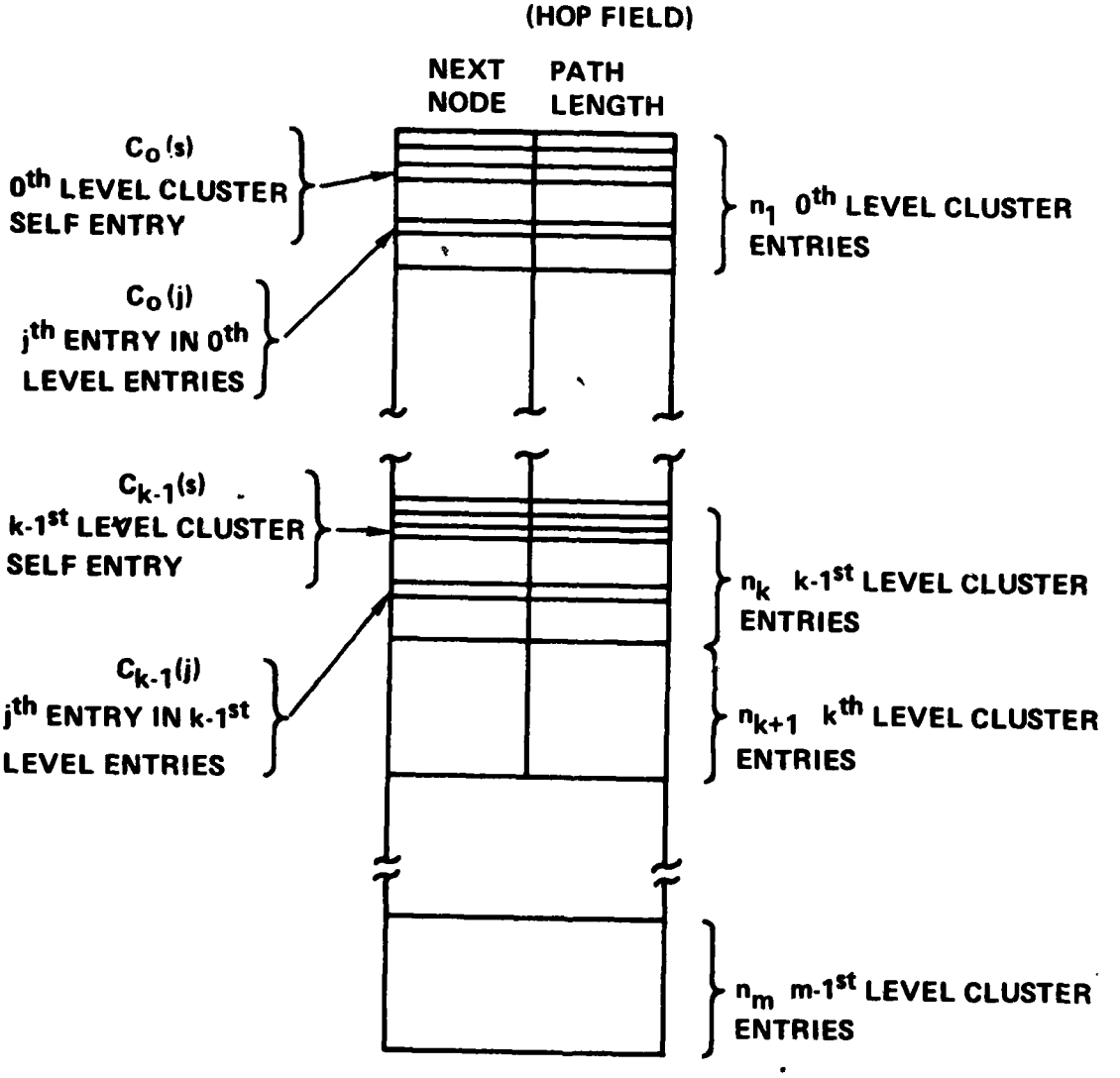


Figure 3.1. Routing Table at Node s, with an m-Level Hierarchical Clustering.

zero as the content of all the self entries at any RT located at an exchange node. In other words, the distance from an exchange node to the clusters at all levels, to which it belongs, is considered to be equal to zero.

ii. As for the OBR (Overall Best Routing) scheme, the average estimated distance from an exchange node to all the nodes in its cluster (including itself) will be propagated when necessary as the aggregate routing information relative to that cluster. A possible implementation is, at any exchange node's RT, to assign to the hop field of the kth level cluster self entry the average of the contents of the hop fields of all the $k-1^{st}$ level cluster entries in that RT. When required, the computation of the averages must proceed sequentially, starting from $k = 1$ to $k = m - 1$ (see Fig. 3.1). The hop field of the 0th level cluster self entry is always set to zero. The interpretation of the above rule is that, at any exchange node, say e, the 1st level cluster self entry of the RT indicates the estimated average distance from node e to any node in the same 1st level cluster as e, including e. In general, the kth level cluster self entry indicates the estimated average distance (in hops) from node e to any node in the same k^{th} level cluster as e, including e.

Notice that the content of the self entry fields can be well computed by the exchange node receiving the routing update or by the one generating the update. This fact will become clear in the example treated right after the specification of the update rule.

<u>Update rule</u>

Let s and t be two neighbor nodes (i.e., they are connected

by a channel (s, t)) which belong to the same k^{th} level cluster C_k and not to any lower level cluster, (k = 1, 2, ..., m). Let $C_{k-1}(s)$ and $C_{k-1}(t)$ respectively denote the $k\text{-}1^{st}$ level clusters to which s and t belong. From the condition above, we know that

$$C_{k-1}(s)\ ,\ C_{k-1}(t) \subset C_k \quad \text{and} \quad C_{k-1}(s) \cap C_{k-1}(t) = \phi$$

As a consequence, and due to the RT organization as specified in Chapter 2, the routing tables at s and t are such that all the p-level cluster entries for p = 0, 1, ..., k - 2 refer to different cluster destinations, whereas all the other entries refer to the same cluster destinations.

The object of the updating procedure is to compare the estimated lengths of the paths from s or t to any common destination. Then, the routing tables are updated to show the better paths. More formally, let

$$C_j(i) \qquad i = 1, 2, \ldots, n_{j+1}; \qquad j = k - 1, \ldots, m - 1$$

denote a common (to s and t) j^{th} level cluster destination. To that cluster is associated entry i (at both tables) among the jth level cluster entries; that entry will also be denoted by $C_j(i)$ (Fig. 3.1). Also let $HF(u, C_j(i))$ represent the content of the hop field of entry $C_j(i)$ at node u (u = s or t). Finally, whenever node t receives an update message from node s, then for each common destination entry $C_j(i)$ the following updating algorithm is performed.

$$\begin{aligned} &\text{IF} \quad HF(t, C_j(i)) > 1 + HF(s, C_j(i)) \\ &\text{THEN} \quad HF(t, C_j(i)) \leftarrow 1 + HF(s, C_j(i)) \\ &\qquad \text{NEXT NODE FIELD OF } C_j(i) \leftarrow s \qquad \text{END} \end{aligned} \tag{3.1}$$

Initially all the entries are set to a large value (∞), except for the self entries. If a CER is used then all the self entries are set to zero, and if an OBR is used then only the 0th level cluster self entries are set to zero, e.g., at node s

$$\begin{cases} HF(s, C_0(s)) \triangleq HF(s, s) = 0 \\ HF(s, C_k(s)) = \begin{cases} 0 & \text{CER} \\ \infty & \text{OBR} \end{cases} \qquad k = 1, \ldots, m-1 \\ \text{all other entries} = \infty \end{cases}$$

Notice that in the algorithm above, it is assumed that all the routing information contained in the non-common destination entries in node s routing table is aggregated as specified before, to represent $HF(s, C_{k-1}(s))$. Moreover the content of the common self entries is not relevant.

The fact that $HF(s, C_{k-1}(s))$ need only be computed for updating purposes, makes it clear that either the sending (s) or the receiving node (t) can perform the aggregation of routing information.

A few more remarks can be stated about the above updating rule.

i. If s and k belong to the same 1st level cluster, then their RT's contain only common destination entries. As a result, Algorithm 3.1 will be performed for all the entries in the table.

ii. A unique "degenerate" MHR routing scheme corresponds to either the OBR or the CER schemes with only one hierarchical level, and it is exactly the non-clustered scheme specified above. Moreover, for such a degenerate case all the network nodes belong to the same unique 1st level cluster; hence, as expected, the updating algorithm will be

performed for all the entries in the RT's.

iii. For any pair of nodes s, t, the common region in the routing tables can be determined by inspecting the address vectors of s and t. A similar operation is required by the routing function described later in this section.

We are now ready to treat an example which will expose the mechanisms of updates and aggregation.

Example of update

Let us consider the 14 node network shown in Fig. 3.2.

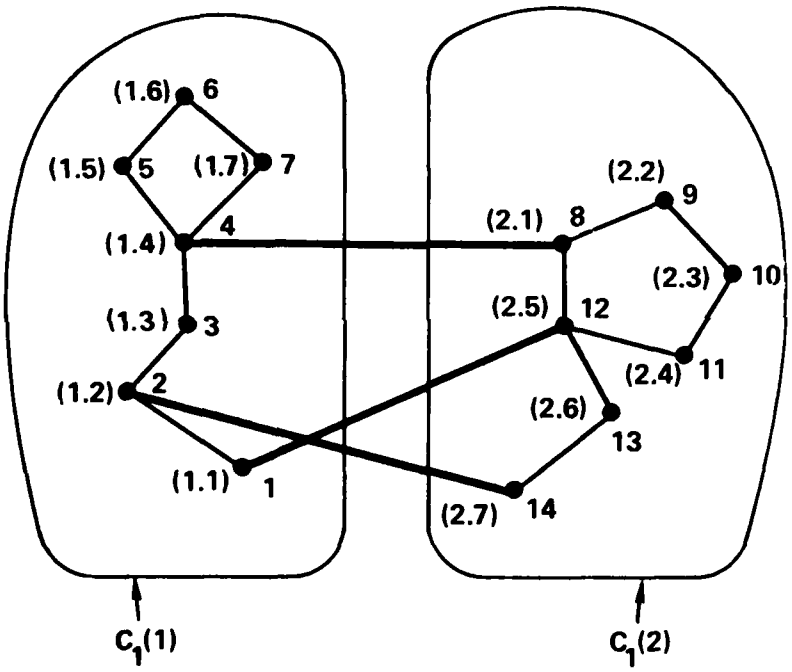


Figure 3.2. A 2-Level Clustered Network.

If no clustering is considered then each node contains a 14-entry routing table. Numbers from 1 to 14 are assigned to the nodes to serve as addresses (shown on the right of each node in Fig. 3.2). If a hierarchical routing is used then the underlying clustering structure is

composed of 2 clusters $C_1(1)$, $C_1(2)$ containing 7 nodes each, i.e., $n_1 = 7$, $n_2 = 2$, $m = 2$ (see Fig. 3.2). The node addresses are as defined in Section 2.3.1.1, and they are shown on the left of the nodes in Fig. 3.2. With this clustering structure the length of an RT is reduced to 9.

In what follows we will present the outcome of the updates and a few key steps. The initialization is performed as mentioned above. After a certain number of updates, if an NCR is used, all the RT's will show (in the hop field) the shortest distances to all the nodes in the network, and the first nodes of the corresponding shortest paths in the "next-node" field. A typical RT is shown in Table 3.1.

Destination	Next Node	Hop Field
1	1	1
2	1	2
3	1	3
4	8	2
5	8	3
6	8	4
7	8	3
8	8	1
9	8	2
10	11	2
11	11	1
12	x	0
13	13	1
14	13	2

Node 12, NCR Used

Table 3.1 RT at Node 12

If an OBR or a CER is utilized, then updates between nodes in the same cluster are performed similar to the non-clustered case. As a consequence, after a certain number of updates the RT's will show the shortest internal (included in the cluster) paths for nodes in the same cluster. Tables 3.2.a and 3.2.b show the outcome of the updates at the exchange nodes routing tables. For the sake of clarity, the contents of the 1st level cluster self entries are not shown, and also, the other cluster entries are left equal to the initial value (∞).

Destination						
(1,1)		0	(1,1)	1	(1,3)	3
(1,2)	(1,2)	1		0	(1,3)	2
(1,3)	(1,2)	2	(1,3)	1	(1,3)	1
(1,4)	(1,2)	3	(1,3)	2		0
(1,5)	(1,2)	4	(1,3)	3	(1,5)	1
(1,6)	(1,2)	5	(1,3)	4	(1,5)	2
(1,7)	(1,2)	4	(1,3)	3	(1,7)	1
(1,)						
(2,)		∞		∞		∞
at node	(1,1)		(1,2)		(1,4)	

Table 3.2.a RT's at the Exchange Nodes

Destination

(2,1)		0	(2,1)	1	(2,1)	3
(2,2)	(2,2)	1	(2,1)	2	(2,6)	4
(2,3)	(2,2)	2	(2,4)	2	(2,6)	4
(2,4)	(2,5)	2	(2,4)	1	(2,6)	3
(2,5)	(2,5)	1		0	(2,6)	2
(2,6)	(2,5)	2	(2,6)	1	(2,6)	1
(2,7)	(2,5)	3	(2,6)	2		0
(1,)		∞		∞		∞
(2,)						
at node	(2,1)		(2,5)		(2,7)	

Table 3.2.b RT's at the Exchange Nodes

Let us now look at the exchange of updates between nodes in different clusters. As an example, assume that node (2,5) receives update from node (1,1). The first 7 entries (in their RT's) do not refer to common destinations and as such they are only utilized to compute the aggregate routing information. That information (0 for CER, average for OBR) is computed and stored in the 1st level cluster self entry at node (1,1)'s RT, by either the sending or receiving node. (Results of such computations are shown in Table 3.3.)

Content of entry (1,)						
		2 + 5/7		2		1 + 3/7	OBR
		0		0		0	CER
at node	(1,1)		(1,2)		(1,4)		

Table 3.3 Aggregation of Routing Information

After the computation of the aggregate information, Algorithm 3.1 is performed. The outcome of this single step is shown in Table 3.4 for all the exchange nodes in $C_1(2)$.

Entry (1,)							
	(1,4)	2 + 3/7	(1,1)	3 + 5/7	(1,2)	3	OBR
	(1,4)	1	(1,1)	1	(1,2)	1	CER
at node	(2,1)		(2,5)		(2,7)		

Table 3.4 Intermediate Update

This information recorded in entry $C_1(1)$ in the above RT's will percolate inside cluster $C_1(2)$. At one point (2,5) will receive an update from (2,1) which will (if OBR is used) trigger a change in $C_1(1)$'s entry. It can be easily checked that entries reserved to $C_1(1)$ in (2,1) and (2,7) will not be affected by updates coming from nodes in $C_1(2)$. As a result, the final outcome for such entries is as shown in Table 3.5.

Entry (1,)	(1,4)	2 + 3/7	(2,1)	3 + 3/7	(1,2)	3	OBR
	(1,4)	1	(1,1)	1	(1,2)	1	CER
at node	(2,1)		(2,5)		(2,7)		

Table 3.5 Final Outcome

Notice that the resulting average path length from (2,5) to the nodes in $C_1(1)$, is

3 + 5/7	CER	(computed from Table 3.5)
3 + 3/7	OBR	(see Table 3.5)
2 + 4/7	NCR	(see Table 3.1)

As for (2,1) and (2.7), that average path length remains the same with either one of the hierarchical schemes. The comparison of the above numbers shows, as expected, that globally CER will induce a longer network path length than OBR which, in turn, induces a longer path than NCR.

Routing Function

The problem here is to decide upon the RT entry to utilize in order to forward (or to send) a message from a node of address $\underset{\sim}{i} = (i_m, i_{m-1}, \ldots, i_1)$ to a node of address $\underset{\sim}{j} = (j_m, j_{m-1}, \ldots, j_1)$ (see Section 2.3.1.1). Let us first notice that if k is the level of the lowest level cluster to which both nodes belong, then their address vectors must match starting from the left, up to and including, the index k + 1, i.e.,

$$i_p = j_p \qquad p = m, m - 1, \ldots, k + 1$$

$$i_k \neq j_k$$

As an example we compare the addresses of a few pairs of nodes chosen from the 3 level clustered network shown in Fig. 2.1. The outcome of the comparison is given in the table below.

(i_3,i_2,i_1)	(j_3,j_2,j_1)	k
(1,3,3)	(3,2,1)	3
(1,3,3)	(1,2,2)	2
(1,3,3)	(1,3,2)	1

Table 3.6 Matching the Address Vectors

Because of the above property, the routing algorithm can then easily perform a sequential, from left to right, matching of the address vectors to find the value of k. Then the next node field of the j_k^{th} entry among the k-1st level cluster entries (see Fig. 3.1) will indicate the next node on which to forward the message.

With the above specifications of the MHR and NCR schemes, we are now ready to address the question as to what is the content of the hop fields at any RT, under some defined equilibrium conditions.

3.1.3 Path Characteristics

If no changes occur in the topology of the network, after a certain number of updates the contents of the hop fields in the routing table will reach "minimal" constant values. In what follows, this

situtation will be referred to as <u>equilibrium</u> condition. Similar to the dynamic programming approach, the above property is due to the fact that improvements are made sequentially at each update over the distance from one node to any cluster (see Algorithm 3.1). The question arises as to what is the meaning of the routing information at equilibrium, or in other words, what are the characteristics of the paths indicated by the routing tables. We have already noticed that for the degenerate one level hierarchical clustering, i.e., when no clustering is used, those paths correspond to the shortest paths in the current topology. Before we proceed, a few more definitions and notation are necessary.

h^c_{st} = Length of the estimated minimum path from node p to node t as derived from the routing information at node s. (The superscript c stands for clustered routing.)

Internal path = a path is defined to be internal (included) in a cluster C_k if all the nodes in that path belong to that cluster.

h^i_{st} = Length of the shortest path from node s to node t <u>included in the lowest level cluster</u> to which both s and t belong.

Exchange node = (defined previously) An exchange node (to be denoted by e or e_i) of a given cluster is a node of that cluster which is connected to one or more nodes external to that cluster.

$A_k(\underset{\sim}{i}_{k+1}) \triangleq$ Subset of all the exchange nodes which connect cluster $C_k(\underset{\sim}{i}_{k+1})$ with any k^{th} level cluster which belongs to the same $k+1^{st}$ level cluster as $C_k(\underset{\sim}{i}_{k+1})$.

$w_{eC_k} \triangleq$ Aggregate variable representing cluster C_k and as computed from the routing information contained at the exchange node e of C_k.

From the above definitions and previous specifications we notice first that a network node (0th level cluster) is its own exchange node, second that

$$\begin{cases} w_{eC_k} = \begin{cases} \dfrac{1}{|C_k|} \sum\limits_{f \in C_k} h^c_{ef} & \text{for the OBR scheme} \\ 0 & \text{for the CER scheme} \end{cases} \\ w_{eC_0} = 0 \end{cases} \tag{3.2}$$

where $|C_k|$ represents the number of nodes in cluster C_k and f is an arbitrary node of C_k. The above considerations allow us to characterize the path lengths under the MHR schemes.

<u>Proposition 3.1</u>

Let s and t be two arbitrary nodes which belong to the same k^{th} level cluster C_k but not to any lower level cluster; then the length of the path from node s to node t as derived at equilibrium from the routing information contained at node s, satisfies the recursive equation below,

$$h^c_{st} = h^i_{se_0} + h^c_{e_0t} \tag{3.3}$$

where e_0 is an exchange node of $C_{k-1}(t)$ which is such that

$$h^i_{se_0} + w_{e_0 C_{k-1}(t)} = \min_{e_j \varepsilon A_{k-1}(t)} \left\{ h^i_{se_j} + w_{e_j C_{k-1}(t)} \right\} \qquad (3.4)$$

where $C_{k-1}(t)$ is the $k\text{-}1^{st}$ level cluster which contains node t, and $A_{k-1}(t)$ is its corresponding subset of exchange nodes as defined above.

<u>Proof</u>

The proof proceeds by induction on the level k of the lowest level common cluster. In what follows $C_j(s)$ and $C_j(t)$, $j = 1, \ldots, m$, will always respectively denote the jth level clusters to which s and t belong.

<u>k = 1</u>

s, t belong to the same 1st level cluster C_1, then

$$C_0(s) = A_0(s) = \{s\}$$

$$C_0(t) = A_0(t) = \{t\}$$

Also, since the distance of a node to itself is zero, then

$$h^c_{e_0 t} = h^c_{tt} = 0$$

In order to prove Eq. (3.3) there remains to show that

$$h^c_{st} = h^i_{st}$$

i.e., that h^c_{st} is the length of the shortest path from s to t included in C_1. This is true since the RT of any node in C_1 contains an entry for node t; hence at equilibrium we obtain the minimal internal path from s to t. Notice that if m = 1, i.e., the degenerate case, all nodes belong to the same cluster C_1 which corresponds to the entire set of

nodes, hence $h^i_{st} = h_{st}$. In other words, when no clustering is used the routing information indicates, at equilibrium, the shortest (hop) path; which checks with the conclusion reached in Section 3.1.1.

Assuming that Proposition 3.1 is true up to k - 1, let us show that it is true for k.

Proof for k

Let C_k be the kth level cluster common to s and t. All the nodes in C_k contain in their RT's one entry for cluster $C_{k-1}(t)$. The propagation and the subsequent updating of the RT's among the nodes of C_k, is equivalent to finding the minimum path, internal to C_k, from any node in $\{C_k - C_{k-1}(t)\}$ to the fictitious supernode $SC_{k-1}(t)$ shown in Fig. 3.3. In other words, seen from any node in $\{C_k - C_{k-1}(t)\}$,

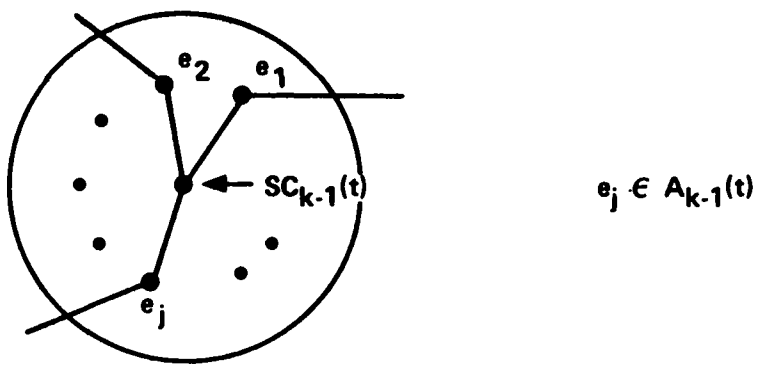


Figure 3.3. Equivalent Representation of Cluster $C_{k-1}(t)$.

cluster $C_{k-1}(t)$ is equivalent , in terms of distance, to a center node $SC_{k-1}(t)$ connected to all the exchange nodes in $A_{k-1}(t)$. If $e_j \in A_{k-1}(t)$ then the length of the equivalent edge, from e_j to the

center node, is equal to the aggregate information representing cluster $C_{k-1}(t)$ as seen from e_j, i.e.,

$$\ell(e_j, SC_{k-1}(t)) = w_{e_j C_{k-1}(t)} \tag{3.5}$$

Notice if the CER scheme is used then the above equivalent representation reduces to the single node $SC_{k-1}(t)$. In general, from Eq. (3.2)

$$w_{e_j C_{k-1}(t)} = \begin{cases} \dfrac{1}{|C_{k-1}(t)|} \displaystyle\sum_{f \in C_{k-1}(t)} h^c_{e_j f} & \text{(OBR)} \\ 0 & \text{(CER)} \end{cases} \tag{3.6}$$

Since e_j and f belong to the same $k\text{-}1^{st}$ level cluster then $h^c_{e_j f}$ is defined because of the induction hypothesis. Hence, the above equation is known.

If e_0 is the exchange node in $A_{k-1}(t)$ which belongs to the minimal path from s to $SC_{k-1}(t)$ obtained at equilibrium, then e_0 satisfies Eq. (3.4) which represents the length of that minimal path. Due to the routing function, previously specified, all messages to be forwarded or sent from node s to node t will follow the same minimal path up to the exchange node e_0. At that point e_0 and t belong to the same $k\text{-}1^{st}$ level cluster, hence, $h^c_{e_0 t}$ is known from the induction hypothesis. Consequently

$$h^c_{st} = h^i_{se_0} + h^c_{e_0 t}$$

where e_0 satisfies Eq. (3.4). Q.E.D.

Remarks

i. If the CER scheme is used, then Eq. (3.4) becomes

$$h^i_{se_0} = \min_{e_j \varepsilon A_{k-1}(t)} \{h^i_{se_j}\} \tag{3.7}$$

The above equation indicates that e_0 is the closest exchange node of $C_{k-1}(t)$ (for paths included in C_k) to node s. This explains the nomenclature: Closest Entry Routing.

ii. Arbitrary Channel Lengths. Instead of assigning (as above) length 1 (1 hop) to all the channels, we intend to let the channels be of variable lengths. Let a_{ij} represent the length of channel (i,j), then the length of a path, π_{st}, from node s to node t is defined as

$$\ell(\pi_{st}) = \sum_{(i,j)\varepsilon\pi_{st}} a_{ij} \tag{3.8}$$

The specifications of the MHR and the NCR schemes are still valid except that any hop distance is replaced by the new assigned length. Hence Algorithm 3.1 becomes

$$\begin{aligned} &\text{IF} \quad HF(t, C_j(i)) > a_{ts} + HF(s, C_j(i)) \\ &\text{THEN} \quad HF(t, C_j(i)) \leftarrow a_{ts} + HF(s, C_j(i)) \\ &\qquad \text{NEXT NODE FIELD OF } C_j(i) \leftarrow s \qquad \text{END.} \end{aligned} \tag{3.9}$$

Also if we let h^c_{st}, h^i_{st}, and w_{eC_k} be as defined previously, then Proposition 3.1 still holds true.

3.2 Bounds on the Increase in Path Length

The effect of the clustering (reduction of routing information) is the increase of the path length between any pair of nodes, s, t, of an amount $h^c_{st} - h_{st}$. A measure of performance of the MHR schemes could be the relative increase of the average path length, i.e.,

$$D = \frac{h_c}{h} - 1 \tag{3.10}$$

where h_c and h denote the average path length in the network respectively with and without clustering,

$$\begin{aligned} h &= \frac{1}{N(N-1)} \sum_{s,t \in S}\sum h_{st} \\ h_c &= \frac{1}{N(N-1)} \sum_{s,t \in S}\sum h^c_{st} \end{aligned} \tag{3.11}$$

Proposition 3.1 provides a means for computing the values of h^c_{st} for any pair of nodes s, t, for a given outcome of the m-level hierarchical clustering of the set of nodes, S. Consequently, for that particular situation, it is numerically possible to evaluate the relative increase D, Eq. (3.10) and then compare the clustered with the non-clustered schemes. Moreover, with further assumptions on the structure of the hierarchical partitioning of the nodes, we can obtain analytic bounds on the increase in the path length.

Assumption 3.2

The diameter[1] of any k^{th} level cluster subnet (see assumption 3.1) is less than or equal to a quantity d_k, k = 1, ..., m.

[1] Recall that the diameter of a network is the maximum shortest path between pairs of nodes [HARA 72].

Notice that d_m represents the diameter of the entire network and that $d_k > d_{k-1} \geq 0$ for all k's.

Assumption 3.3

Any cluster at any level $k = 1, 2, \ldots, m$ contains the shortest path (if it is not unique, then at least one is contained) between any given pair of nodes which belong to that cluster.

Assumption 3.2 is simply the specification of the outcome of the clustering of the nodes, since the d_k's can be of any value. Whereas Assumption 3.3 is a natural property that any clustering scheme should seek. The reason for this is that traffic between nodes in the same cluster must (because of the routing function above) follow paths internal to that cluster.

The above assumptions lead to the derivation of some simple bounds. These bounds, on the increase in path length, pertain to the routing schemes (OBR, CER) described in Section 3.1. All the properties listed below rely on Assumptions 3.1 and 3.2. If Assumption 3.3 is used, it will be so specified.

Lemma 3.1

Under the above conditions, the value of h^c_{st} for any pair of nodes s, t which belong to the same k^{th} level cluster is such that

$$h^c_{st} \leq \sum_{j=1}^{k} d_j \qquad \begin{array}{l} \forall\ s, t \in \text{same } k^{th} \text{ level cluster} \\ \forall\ k = 1, 2, \ldots, m \end{array} \tag{3.12}$$

Proof

The proof proceeds by induction on k. First if k = 1 then similar to the proof of Proposition 3.1 (for k = 1) $h^c_{st} = h^i_{st}$. Then

from Assumption 3.2, $h^i_{st} \leq d_1$, hence $h^c_{st} \leq d_1$ which checks Eq. (3.12). Now assuming that Lemma 3.1 is true for all levels up to k - 1, let us show that it is true for k.

Let s, t be a pair of nodes which belong to the same k^{th} level cluster C_k. Let C_p be the lowest level cluster to which both s and t belong, hence $p \leq k$. If $p < k$ then from the induction hypothesis.

$$h^c_{st} \leq \sum_{j=1}^{p} d_j \leq \sum_{j=1}^{k} d_j$$

which checks Eq. (3.12). Else if p = k, then Eq. (3.3) holds true. In Eq. (3.3), because of Assumption 3.2

$$s,\ e_0 \in C_k \Rightarrow h^i_{se_0} \leq d_k$$

Also, because of the induction hypothesis

$$e_0,\ t \in C_{k-1}(t) \Rightarrow h^c_{e_0 t} \leq \sum_{j=1}^{k-1} d_j$$

Substituting the above relations into Eq. (3.3), we arrive at Eq.(3.12).

Lemma 3.1 leads to a bound on the increase in the average path length and on D (Eq. (3.10)).

<u>Proposition 3.2</u>

Under the conditions above and Assumption 3.3, the increase in the average path length in the network due to the reduction of routing information, is such that:

$$h_c - h \leq \sum_{k=1}^{m-1} \left[1 - \frac{n_1 n_2 \ldots n_k - 1}{N - 1}\right] d_k \qquad (3.13)$$

Proof

Let $C_k(s)$ denote the kth level cluster ($k = 0, 1, \ldots, m$) to which s belongs. Then from Eq. (3.11),

$$h_c - h = \frac{1}{N(N-1)} \sum_{s \varepsilon S} \sum_{k=1}^{m} \sum_{\substack{t \varepsilon C_k(s) \\ t \notin C_{k-1}(s)}} (h^c_{st} - h_{st}) \qquad (3.14)$$

The above expression was obtained through the decomposition of the computation of the average over all the nodes in the network. Then for a given node, s, the decomposition i done over nodes in the same 1st level cluster as s, excluding s ($C_0(s) = s$), then over nodes in the same 2nd level cluster as s, excluding those in the same 1st level cluster as s, etc...

Let $C_{k-1}(j)$ be a $k\text{-}1^{st}$ level cluster included in $C_k(s)$; there are n_k such clusters, then

$$\sum_{\substack{t \varepsilon C_k(s) \\ t \notin C_{k-1}(s)}} (h^c_{st} - h_{st}) = \sum_{\substack{j=1 \\ C_{k-1}(j) \cap C_{k-1}(s) = \phi}}^{n_k} \sum_{t \varepsilon C_{k-1}(j)} (h^c_{st} - h_{st}) \qquad (3.15)$$

Since $C_{k-1}(j) \cap C_{k-1}(s) = \phi$ and both are included in C_k's, Eq. (3.3) holds true for s and any node t in $C_{k-1}(j)$, hence

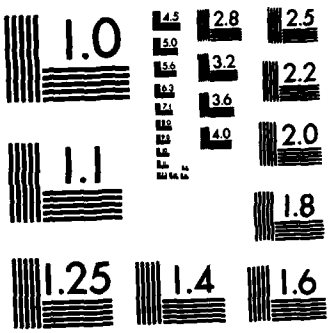
$$\sum_{e \varepsilon C_{k-1}(j)} h^c_{st} = |C_{k-1}(j)| \; h^i_{se_0} + \sum_{t \varepsilon C_{k-1}(j)} h^c_{e_0 t} \qquad (3.16)$$

where e_0 satisfies Eq. (3.4). Two cases to consider are

i. OBR scheme. From Eq. (3.2)

AD-A122 995 DESIGN CONSIDERATIONS FOR LARGE COMPUTER COMMUNICATION NETWORKS(U) CALIFORNIA UNIV LOS ANGELES SCHOOL OF ENGINEERING AND APPLIED SCIENCE F KAMOUN APR 76 2/5

UNCLASSIFIED UCLA-ENG-7642 DAHC15-73-C-0368 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

$$w_{e_0 C_{k-1}(j)} = \frac{1}{|C_{k-1}(j)|} \sum_{f \varepsilon C_{k-1}(j)} h^c_{e_0 f} \tag{3.17}$$

Substituting Eqs. (3.17) and (3.4) (in Eq. (3.4) t is replaced by j) into Eq. (3.16), we arrive at

$$\sum_{t \varepsilon C_{k-1}(j)} h^c_{st} = |C_{k-1}(j)| \min_{e_j \varepsilon A_{k-1}(j)} \{h^i_{se_j} + w_{e_j C_{k-1}(j)}\} \tag{3.18}$$

Let us define e_s to be the closest (inside $C_k(s)$) exchange node of $A_{k-1}(j)$ to node s, i.e.,

$$h^i_{se_s} = \min_{e_j \varepsilon A_{k-1}(j)} \{h^i_{se_j}\} \tag{3.19}$$

From Eq. (3.18) and for any exchange node e_j, particularly e_s, the relation below is true.

$$\sum_{t \varepsilon C_{k-1}(j)} h^c_{st} \leq |C_{k-1}(j)| \, h^i_{se_s} + \sum_{t \varepsilon C_{k-1}(j)} h^c_{e_s t} \tag{3.20}$$

Note that in the equation above w was replaced by its value as defined by Eq. (3.2).

Moreover, from Assumption 3.3

$$h^i_{st} = h_{st} \qquad \forall\, s,\ t \tag{3.21}$$

Then from the definition of e_s,

$$h_{st} = h^i_{st} \geq h^i_{se_s} \qquad \forall\, t \,\varepsilon\, C_{k-1}(j) \tag{3.22}$$

Consequently,

$$\sum_{t \varepsilon C_{k-1}(j)} h_{st} \geq |C_{k-1}(j)| \, h^{i}_{se_s} \tag{3.23}$$

Substituting Eq. (3.23) into Eq. (3.20), we arrive at

$$\sum_{t \varepsilon C_{k-1}(j)} (h^{c}_{st} - h_{st}) \leq \sum_{t \varepsilon C_{k-1}(j)} h^{c}_{e_s t} \tag{3.24}$$

Note that e_s, $t \varepsilon C_{k-1}(j)$, then from Lemma 3.1,

$$h^{c}_{e_{st}} \leq \sum_{j=1}^{k-1} d_j \qquad \forall \, t \varepsilon C_{k-1}(j) \tag{3.25}$$

From Assumption 3.1

$$|C_{k-1}(j)| = n_1 n_2 \ldots n_{k-1} \qquad \forall \, k, \, j \tag{3.26}$$

Substituting Eq. (3.24 - 3.26) into Eq. (3.15), we find

$$\sum_{\substack{t \varepsilon C_k(s) \\ t \notin C_{k-1}(s)}} (h^{c}_{st} - h_{st}) \leq (n_k - 1) n_1 n_2 \ldots n_{k-1} \sum_{j=1}^{k-1} d_j \tag{3.27}$$

Note that this last equation is true for any level k, and for any node s, hence by substituting it into Eq. (3.14), we obtain

$$h_c - h \leq \frac{1}{N - 1} \sum_{k=1}^{m} n_1 n_2 \ldots n_{k-1} (n_k - 1) \sum_{j=1}^{k-1} d_j$$

Note that for $k = 1, \sum_{j=1}^{0} \triangleq 0$.

Interchanging the summations in the equation above, we get

$$h_c - h \leq \frac{1}{N - 1} \sum_{j=1}^{m-1} d_j \sum_{k=j+1}^{m} n_1 n_2 \ldots n_{k-1} (n_k - 1)$$

which after some algebra, gives Eq. (3.13).

To end the proof it is left to show that the CER scheme also satisfies Proposition 3.2.

ii. CER Scheme. From Eqs. (3.2) - (3.4) and (3.19), we find

$$\begin{cases} h^c_{st} = h^i_{se_s} + h^c_{e_s t} \\ \sum_{t \varepsilon C_{k-1}(j)} h^c_{st} = |C_{k-1}(j)| h^i_{se_s} + \sum_{t \varepsilon C_{k-1}(j)} h^c_{e_s t} \end{cases} \tag{3.28}$$

This equation is equivalent to Eq. (3.20) except that the inequality is tight here. Two conclusions: (1) the rest of the proof can proceed exactly as in (i), (2), Eq. (3.28) compared to Eq. (3.20), indicates that the summation of path lengths obtained with the OBR scheme is smaller than or equal to the one obtained with the CER scheme. Hence the average path with an OBR is smaller than or equal to the average path length with a CER.

The above proposition deals with averages; we now intend to put a bound on the increase of the path length between an arbitrary pair of nodes s, t.

Lemma 3.2

Under the previous conditions and Assumption 3.3, and for the CER scheme

$$h^c_{st} - h_{st} \leq \sum_{j=1}^{k-1} d_j \qquad \begin{array}{l} \forall\, s,\, t \,\varepsilon \text{ same } k^{th} \text{ level cluster } C_k \\ \forall\, k = 1, 2, \ldots, m \end{array} \tag{3.29}$$

Proof

Under the above condition Eq. (3.28) holds true

$$h^c_{st} = h^i_{se_s} + h^c_{e_s t}$$

Since e_s , $t \in C_{k-1}(j)$, then from Lemma 3.1

$$h^c_{e_s t} \leq \sum_{i=1}^{k-1} d_i$$

Also, because of Assumption 3.3, and Eq. (3.19)

$$h_{st} = h^i_{st} \geq h^i_{se_s}$$

Combining the last three relations together, we find Eq. (3.29).

Note that lemma 3.2 is not true with the OBR scheme, because such a scheme tries to find the best overall exchange node (e_o) to route the messages to, instead of using the closest one.

We observed previously that Assumption 3.3 is a realistic one, but if it is not specifically built into the clustering algorithm, there is no guarantee that the outcome of the clustering always satisfies that assumption. This remark leads us to the following proposition.

Proposition 3.3

Under the conditions of Proposition 3.2 and with Assumption 3.3 removed,

$$h_c - h \leq \sum_{k=1}^{m-1} d_k \qquad (3.30)$$

Proof

Let s, t be an arbitrary pair of nodes and let C_k be the lowest

level cluster to which s, t belong. Two cases to consider are

i. $k < m$. From Lemma 3.1

$$h^c_{st} \leq \sum_{j=1}^{k} d_j$$

Hence

$$h^c_{st} - h_{st} \leq h^c_{st} \leq \sum_{j=1}^{k} d_j \leq \sum_{j=1}^{m-1} d_j \tag{3.31}$$

ii. $k = m$. C_m represents the entire set of nodes, hence

$$h^i_{st} = h_{st}$$

Then similar to the proof of Proposition 3.2, we can show that Eq. (3.27) holds true for $k = m$, i.e.,

$$\sum_{\substack{t \in C_m(s) \\ t \notin C_{m-1}(s)}} (h^c_{st} - h_{st}) \leq (n_m - 1) n_1 n_2 \ldots n_{m-1} \sum_{j=1}^{m-1} d_j \tag{3.32}$$

Because of Eq. (3.31) and Assumption 3.1

$$\sum_{t \in C_{m-1}(s)} (h^c_{st} - h_{st}) \leq n_1 n_2 \ldots n_{m-1} \sum_{j=1}^{m-1} d_j \tag{3.33}$$

Substituting Eqs. (3.32) and (3.33) into (3.14) we arrive at Eq. (3.30).

Conclusion

In summary, several fairly general bounds have been derived, depending on the assumptions and/or the routing schemes selected. In the next paragraph we will study the behavior of some of those bounds in the context of a defined class of networks. Let us also show that for the degenerate case of 1-level hierarchical routing (NCR scheme)

the bounds derived over the increase in path length are tight; i.e., for m = 1, Eqs. (3.13) and (3.30) become

$$h_c - h \leq 0$$

but since $h_c - h \geq 0 \Rightarrow h_c = h$. Also Eq. (3.29) becomes $h_{st}^c = h_{st}$.

3.3 Static Performance Evaluation of the MHR Schemes for a Family of Networks

If temporarily we do not take into account the significant gains obtained in reducing the CPU, storage and line utilizations required by the routing procedures, then the application of the MHR schemes will result in a degradation of the performance of the network, as compared to the utilization of a non-clustered scheme. The loss in performance (delay, throughput) is closely related to the average path length a message follows in the network. The evaluation of the increase in path length provides us with a first cut modeling of the loss in network performance. Moreover, the study of the bounds, derived previously, represents a worst case evaluation of the MHR schemes. Since the evaluation is in terms of path length, we will refer to it as static performance evaluation. The gains obtained are modeled by the single variable ℓ/N which represents the reduction of routing information (we will refer to ℓ/N as the relative table length). The static performance evaluation is performed over a class of computer networks.

3.3.1 A Family of Large Distributed Networks

The networks to be considered are all the connected graphs

upon which it is possible to fit an m-level hierarchical clustering whose outcome satisfies Assumptions 3.1 - 3.3. Also the resulting cluster subnets at any level are of diameters bounded by a power law function of the number of nodes in that cluster; i.e., if n is the size of a cluster and d the diameter of that cluster's subnet then

$$d \leq bn^{v} + c \tag{3.34}$$

where b, c, v are positive parameters and $0 \leq v \leq 1$ (see below).

If N is the size of such a network, then the average path length (hop distance) of that network, h, must be a power law function of N,

$$h = aN^{v} \tag{3.35}$$

where a is a positive parameter.

Grid type networks [BARA 64], hexagonal networks, etc., fall into that category when the MHC results in subnetworks of similar structure as the original and when the path lengths are expressed in hops. Expressions for the average path length (with a uniform traffic matrix $\gamma_{ij} = \gamma$) and for the diameter of the grid and the torus networks have been derived in Appendix A. Some of the results obtained are:

$$\text{Square grid of size N} \quad \begin{cases} h = \frac{2}{3}\sqrt{N} \\ d = 2\sqrt{N} - 2 \end{cases} \tag{3.36}$$

$$\begin{array}{c}\text{Square torus of size N,} \\ \text{(with } \sqrt{N} \text{ odd)}\end{array} \quad \begin{cases} h = \frac{\sqrt{N}}{2} \\ d = \sqrt{N} - 1 \end{cases} \tag{3.37}$$

Furthermore, if the partitioning of either the square grid or torus networks results in grid cluster subnets at all levels (see Fig. 5.6), then for any cluster subnet of size n its diameter d is such that

$$d \leq 2\sqrt{n} - 2 \tag{3.38}$$

As a consequence the grid and torus networks fit the above descriptions. Note also that for those networks the exponent v (Eqs. (3.34) and (3.35)) is equal to 1/2.

In general, the exponent v reflects the connectivity of the network considered. For very highly connected networks v is in the neighborhood of zero; e.g., for a fully connected network $v = 0$ ($h = 1$, $d = 1$). Whereas for very low connected networks v is in the neighborhood of one; e.g., for loop or chain type networks, $v = 1$.

Computer communication networks fall into the class of distributed networks. This class includes networks such as the ARPANET, the NPL network [DAVI 73], the Cyclades network [ZIMM 75], TELENET, etc. The main characteristic of those distributed networks is their low connectivity. In general a connectivity 2 (or 3) is imposed on their design. For large distributed networks a connectivity of 3 to 4 seems more appropriate [NAC 73]. The torus networks considered above are of connectivity 4 and with an exponent $v = 1/2$, hence they appear to be good representatives of large distributed networks. Moreover, their topological structures lead to simple partitions such as square subgrid clusters. In the sequel, we will first derive a limiting result valid for the entire class of networks, then we will restrict our considerations to values of a, b, c, v as obtained for a torus.

3.3.2 Limiting Performance of the MHR Schemes

As mentioned earlier, the static performance evaluation of the MHR schemes involves the determination of a bound on the relative increase in the average path length. For the above family of networks, the size of a kth level cluster is $n_1 n_2 \ldots n_k$ and because of Eq. (3.34),

$$d_k = b(n_1 n_2 \ldots n_k)^v + c \qquad k = 1, 2, \ldots, m \tag{3.39}$$

This family of networks also satisfies Assumptions 3.1 - 3.3, hence Proposition 3.2 holds true. From Eqs. (3.10), (3.13) and (3.39)

$$D = \frac{h_c}{h} - 1 \leq E \triangleq \frac{1}{a(N-1)N^v} \sum_{k=1}^{m-1} \left(N - \prod_{i=1}^{k} n_i\right)\left(b\left(\prod_{i=1}^{k} n_i\right)^v + c\right) \tag{3.40}$$

Notice that E is defined as the bound on D. It is the behavior of E versus ℓ/N that we are interested in. Two cases will be distinguished in the study of E; first when the MHC results in a minimum table length, Eq. (2.6), second when the MHC results in a vector minimizing the bound E for a given table length ℓ.

For an optimal clustering structure we know from Proposition 2.1 that the degree vector must satisfy Eq. (2.5). Substituting Eq. (2.5) into Eq. (3.40), we arrive at

$$E = \frac{1}{a(N-1)N^v} \sum_{k=1}^{m-1} \left[N - N^{k/m}\right]\left[bN^{kv/m} + c\right]$$

and if we restrict v to be different from zero, then after some algebra

$$E = \frac{1}{a(N-1)N^v}\left[N\left[b\,\frac{N^v - N^{v/m}}{N^{v/m}-1} + c(m-1)\right] - b\,\frac{N^{v+1} - N^{\frac{v+1}{m}}}{N^{\frac{v+1}{m}}-1} - c\,\frac{N - N^{1/m}}{N^{1/m}-1}\right] \tag{3.41}$$

Notice again that for $m = 1$, $E = 0$.

Also from Eq. (2.6), the relative table length is

$$\frac{\ell}{N} = \frac{mN^{1/m}}{N} \tag{3.42}$$

The above considerations lead to the general limiting result below.

<u>Proposition 3.4: Limiting Performance</u>

Consider the above family of networks and the above MHR schemes (OBR, CER) with a <u>fixed</u> number of levels m and an optimal clustering structure. Then, as N, the number of nodes, goes to infinity, the "static" performance of the MHR schemes approaches that of a non-clustered routing scheme, while the relative table length approaches zero; i.e.,

$$N \to \infty \Rightarrow \begin{cases} \dfrac{h_c}{h} \to 1 \\[2ex] \dfrac{\ell}{N} \to 0 \end{cases} \tag{3.43}$$

Thus we claim we can get the best of both possible worlds.

<u>Proof</u>:

From Eq. (3.42), the above limit of ℓ/N is obvious (recall that m is fixed). Also from Eq. (3.3), it is obvious that

$$h^c_{st} \geq h_{st} \qquad \forall\, s, t \in S$$

hence

$$h_c \geq h \geq 0 \ .$$

Then because of Eq. (3.40)

$$0 \leq \frac{h_c}{h} - 1 \leq E \tag{3.44}$$

As a consequence, it is enough to prove that the limit of E is zero. Expanding Eq. (3.41) around N^{-1}, we find

$$E = \frac{b}{a} N^{-v/m} + 0\left(N^{-v/m}\right) \tag{3.45}$$

hence

$$\lim_{N \to \infty} E = 0 \qquad \text{Q.E.D.}$$

Notice that the closer v is to one ($v \neq 0$), the faster is the convergence of E to zero. In other words, as could be expected, the more distributed (the less connected) the networks are the better the MHR's perform. Notice also the enormous gains in table length for a relatively insignificant increase in the average path length.

The above results hold true if we <u>relax</u> Assumption 3.3 imposed over the outcome of the clustering of any of the networks which belong to the family considered here. To prove this statement let F be the bound on the relative increase in path, as obtained from Proposition 3.3, i.e., from Eq. (3.30) and (3.39),

$$\frac{h_c}{h} - 1 \leq F \triangleq \frac{1}{aN^v} \sum_{k=1}^{m-1} \left[b(n_1 n_2 \ldots n_k)^v + c\right] \tag{3.46}$$

For an optimal clustering, the above equation becomes,

$$F = \frac{1}{aN^v}\left[b\,\frac{N^v - N^{v/m}}{N^{v/m} - 1} + c(m - 1)\right] \tag{3.47}$$

Similar to E, as N goes to infinity, F goes to zero, which implies that h_c/h goes to 1.

The result of Proposition 3.4 was derived for a fixed m; let us now examine the situation where m is <u>variable</u>. Of interest is the value of m which corresponds to the global optimum clustering structure. That value is, from Eq. (2.17), $m_* = \ln N$.

Substituting Eq. (2.17) into Eq. (3.41), we arrive at

$$E_* = E = \frac{1}{a(N - 1)N^v}\left[bN\,\frac{N^v - \ell^v}{e^v - 1} + c(\ln N - 1)N - b\,\frac{N^{1+v} - e^{1+v}}{e^{1+v} - 1} - c\,\frac{N - e}{e - 1}\right] \tag{3.48}$$

Then

$$\lim_{N\to\infty} E_* = \frac{b}{a}\left[\frac{1}{e^v - 1} - \frac{1}{e^{1+v} - 1}\right] \tag{3.49}$$

As a consequence the result of Proposition 3.4 is not necessarily true anymore. If we consider grid or torus networks, then from Eqs. (3.36)-(3.38),

$$\begin{cases} \text{for a grid:} & a = \frac{2}{3},\ b = 2,\ c = -2,\ v = \frac{1}{2} \\ \text{for a torus:} & a = \frac{1}{2},\ b = 2,\ c = -2,\ v = \frac{1}{2} \end{cases} \tag{3.50}$$

Substituting the above values into Eq. (3.49), we arrive at

$$\lim_{N\to\infty} E_* = \begin{cases} 5.01 & \text{for torus nets} \\ 3.76 & \text{for grid nets} \end{cases} \tag{3.51}$$

Fig. 3.4 illustrates the behavior of E_* (for torus) with respect to N. The curve shows E_* as an increasing function of N. It also shows that the cost of operating at the (global) minimum table length may become quite high (up to 6 times increase in path length). Fortunately, as noticed in Chapter 2, most of the table reduction, for practical purposes, may be obtained with m quite a bit smaller than the global number of levels m_*, and as we will see in later plots (Figs. (3.5) and (3.7)) the cost at a small m is quite minimal. Those figures also show the behavior of E_* versus ℓ/N.

3.3.3 Static Performance Evaluation for the MHR's: Numerical Applications

In the previous section we observed that at the limit ($N \to \infty$) considerable table reduction can be achieved with no loss in performance. In this paragraph, we intend to look at the more general case of a finite N. The purpose is to correlate the degradation in performance with the table reduction. More precisely, we will evaluate a maximum performance degradation in terms of the gains in table length. Also this evaluation will be carried out with an MHC which results first in a minimal table length and second in a minimal bound E.

The numerical study below is restricted to values of a, b, c, v, as obtained for torus networks (Eq. (3.50)), although such a study could easily be repeated for other networks which belong to the family considered here.

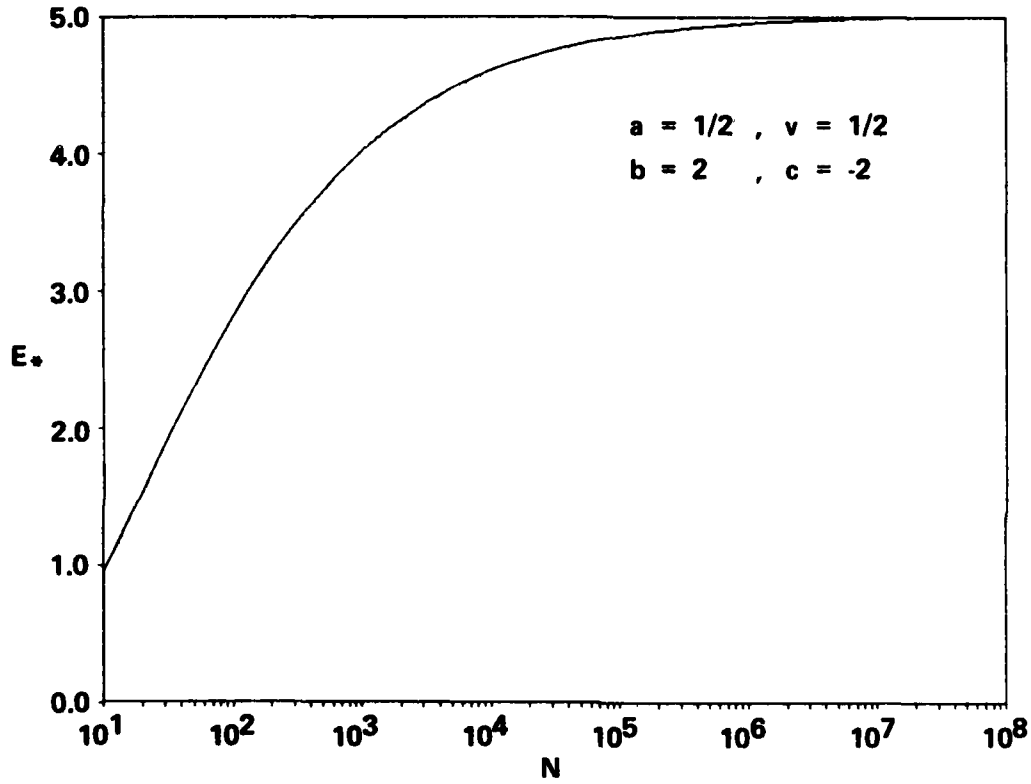


Figure 3.4. Relative Bound on the Increase in Path Length at Global Minimum Table Length.

3.3.3.1 Evaluation with an MHC Resulting in a Minimal Table Length

The expression of E has already been derived, in Eq. (3.41), for an MHC which results in a minimal table length ℓ, Eq. (2.6). If we consider torus networks, then from Eq. (3.50) E becomes

$$\begin{cases} E = \dfrac{2}{(N-1)N^{1/2}} \left[2N \left[\dfrac{N^{1/2} - N^{1/2m}}{N^{1/2m} - 1} - (m-1) \right] \right. \\ \qquad \left. - 2\,\dfrac{N^{3/2} - N^{3/2m}}{N^{3/2m} - 1} + 2\,\dfrac{N - N^{1/m}}{N^{1/m} - 1} \right] \\ \dfrac{\ell}{N} = \dfrac{mN^{1/m}}{N} \end{cases} \tag{3.52}$$

The above equations provide a parametric representation of E as a function of ℓ/N. m acts as the coupling variable in that representation. By letting m vary from 1 to lnN we obtain all the possible values of ℓ/N; and subsequently for each value of ℓ/N we obtain the corresponding value of E. The above range of m is chosen in accordance with the results obtained in Chapter 2 (refer to Proposition 2.2 and Fig. 2.5); and also in accordance with the fact that E is an increasing function of m (this fact is obvious from the proof of Proposition 3.2).

Numerical results are presented in a set of figures as follows: Fig. 3.5 illustrates the behavior of E with respect to ℓ/N and for several values of N. We observe that an original substantial table reduction can be achieved for small values of E, i.e., for a small drop in performance. However if we try to reduce ℓ/N to values close to its global minimum, Eq. (2.19), then E increases sharply. Fig. 3.5 also illustrates the limiting behavior of the MHR schemes (see Proposition

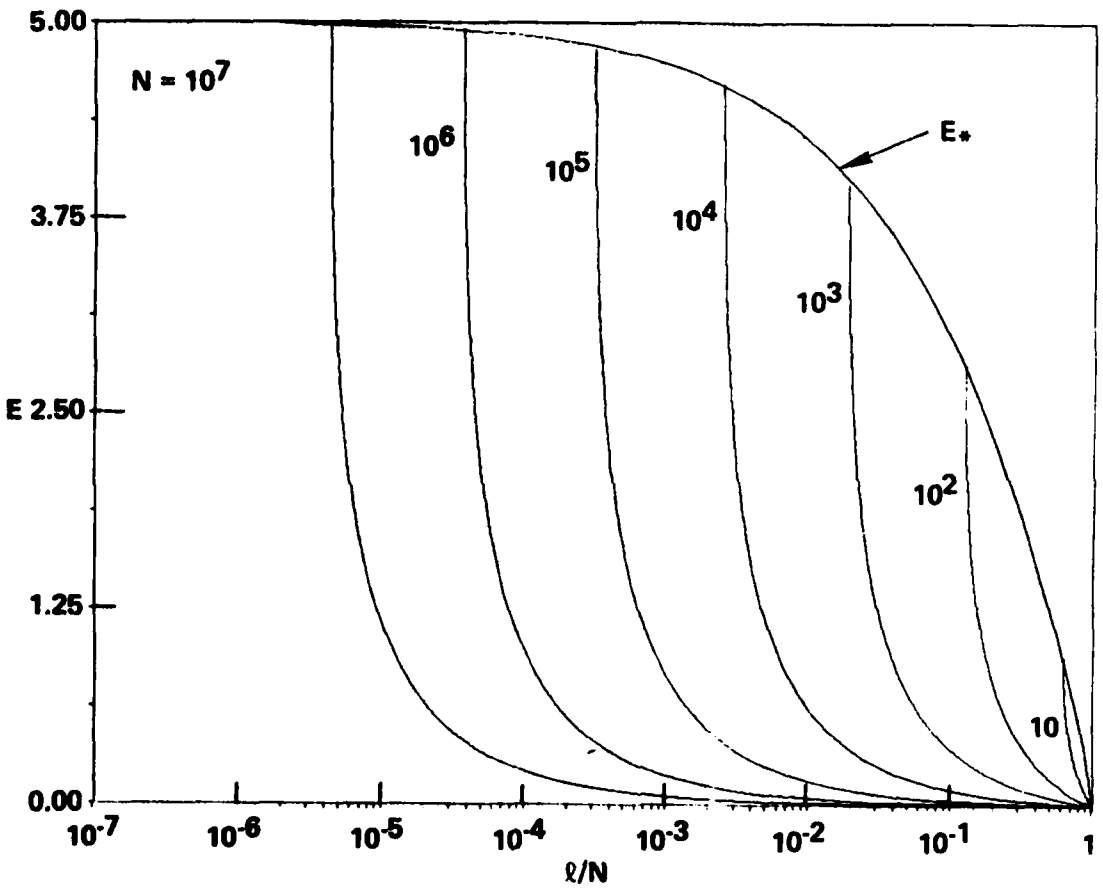


Figure 3.5. Relative Bound on the Increase in Path Length, E, versus the Relative Table Length, ℓ/N.

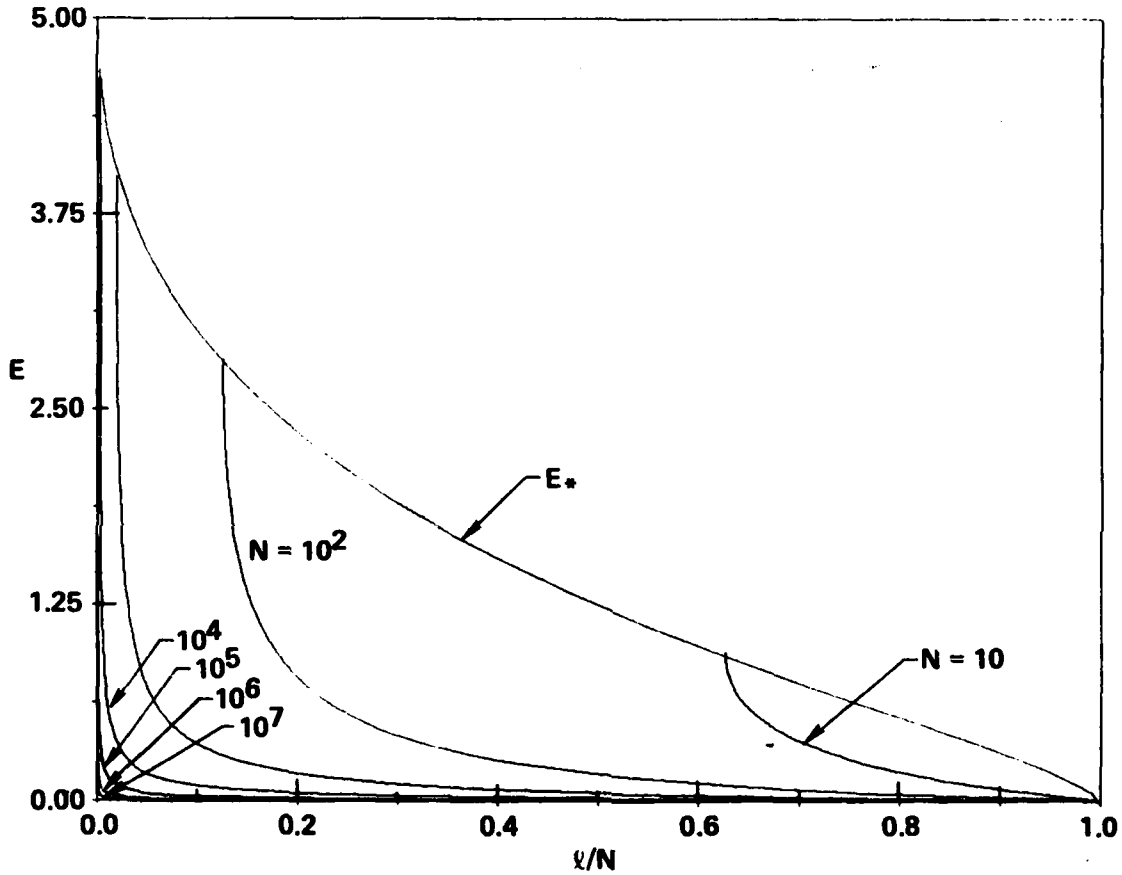


Figure 3.6. E versus ℓ/N, Linear Axis.

3.4) whereby as N becomes larger, more gains can be obtained for a lesser loss in performance. This property is shown by the fact that the curves E versus ℓ/N stay flat on the ℓ/N axis for larger intervals. The linear scale, Fig. 3.6, provides a better illustration of the above phenomenon.

Figs. 3.7 - 3.8 show the behavior of $1/(1 + E)$ with respect to ℓ/N. The function $1/(1 + E)$ represents a lower bound over h/h_c (see Eq. (3.44)). These figures exhibit properties similar to the previous ones.

Finally, Fig. 3.9 shows how much table reduction can be obtained for a given "tolerance" E as a function of the size N. The concentration of the curves for $1 \leq E \leq 5$ (Note from Eq. (3.51) that $E = 5$ is the maximum error for torus networks.) again shows that beyond a certain point the gains in table length can only be achieved at the expense of large losses (large E). However in the range 0 to 1 for E considerable gains can yet be obtained. For that range of E as shown in Fig. 3.10, the corresponding range of the number of levels m is limited to fairly small values. The curves in Fig. 3.10 represent the values of m which are computed from Eq. (3.52) for a given E and N, and which served to find the value of ℓ/N for a given tolerance E (see Fig. 3.9). The fact that m is considered to be a real variable will be interpreted in the next section.

Moreover, in Section 2.3.2.4, we noticed that most of the table reduction is obtained for small values of m. We conclude that the MHR schemes operating with a small number of levels $2 \leq m \leq 4$ yield substantial table reduction for a relatively small increase in

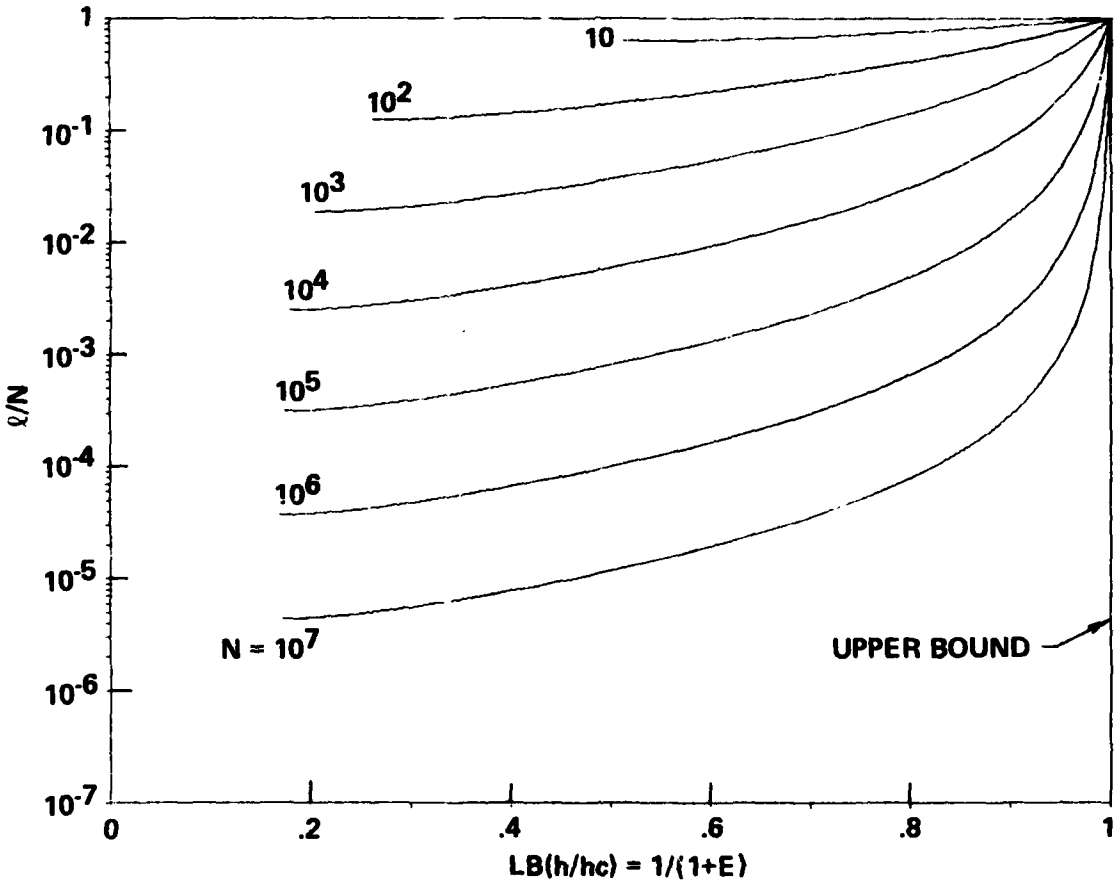


Figure 3.7. Lower Bound on the Ratio of Path Lengths Without and with Clustering, LB(h/hc).

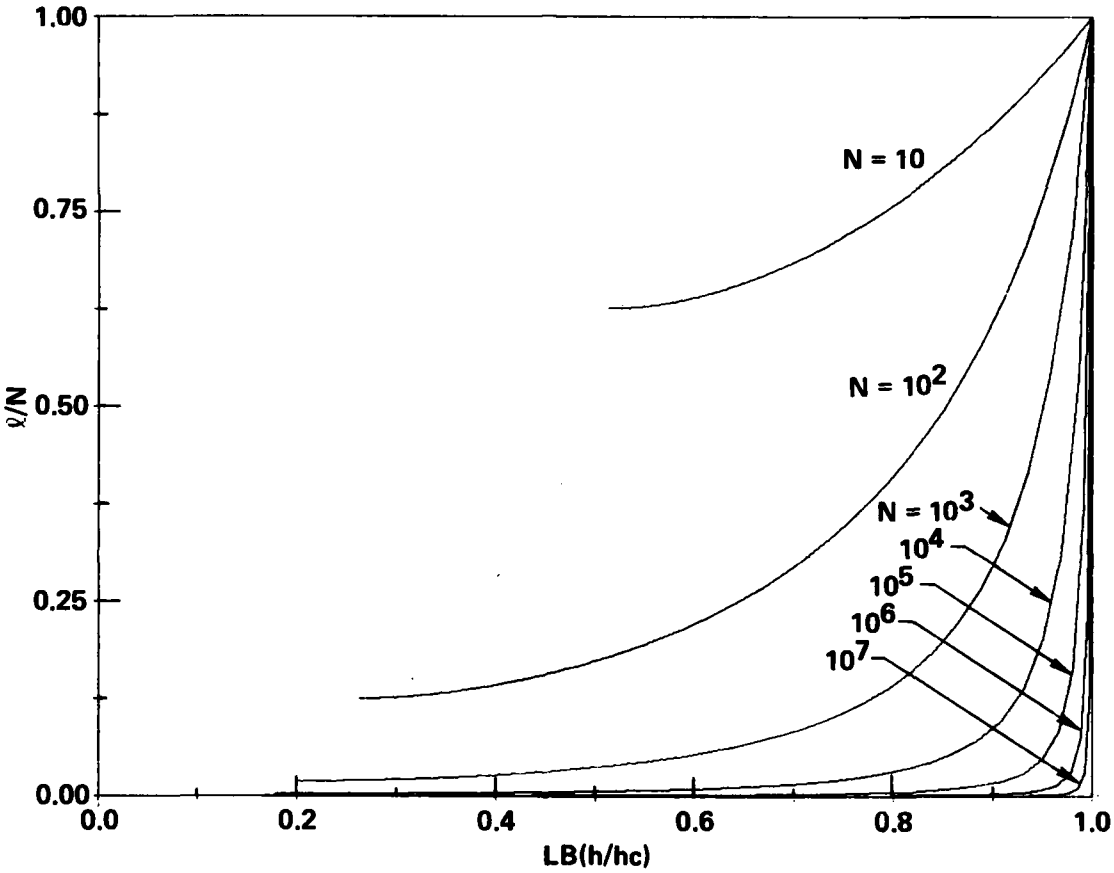


Figure 3.8. LB(h/hc) versus ℓ/N, Linear Axis.

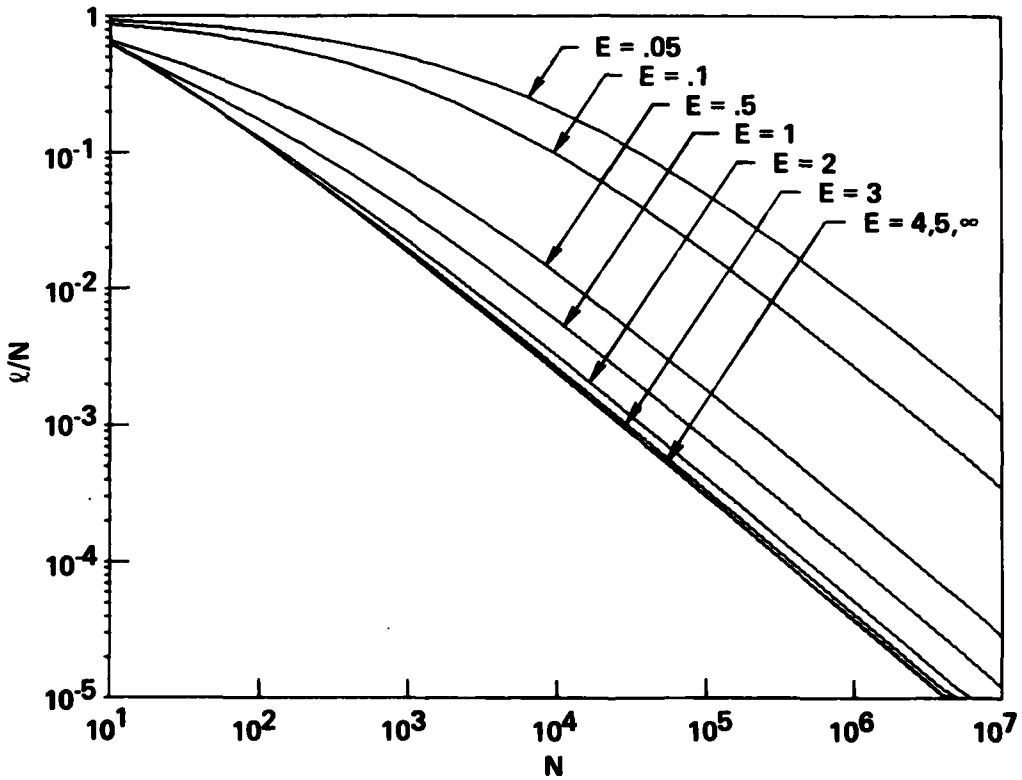


Figure 3.9. Decrease in Table Length for a Given Maximum Increase in Path Length.

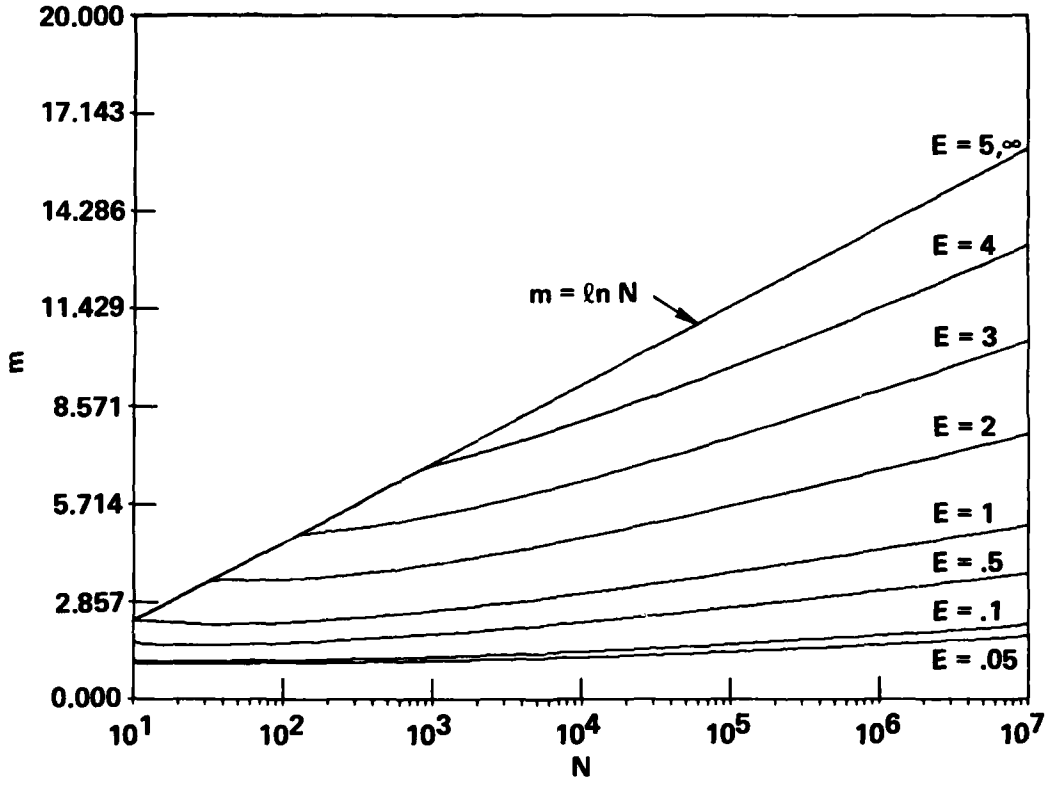


Figure 3.10. Continuous Number of Levels, m.

path length.

The above study considered MHR schemes (OBR, CER) based on clustering structures which yield minimal table length. The question arises as to the existence of a clustering structure which minimizes the increase in path length.

3.3.3.2 Minimal Bounds

In this section we address the problem of choosing a clustering structure (i.e., $\underset{\sim}{n}$, m) which minimizes the increase in path length for a given ℓ and N. Since we have no analytic expression for the increase in path length, we will limit our considerations to minimizing the bound E (or F). In order to simplify the problem we will also assume a partitioning of equal size clusters at any level. Thus for a fixed m, the optimization problem is

$$\begin{cases} \text{Given:} & N,\ \ell,\ m \text{ (integer)} \\ \text{min:} & E \quad \text{(see Eqs. (3.40) and (3.50))} \\ \text{over:} & \underset{\sim}{n} \quad \text{real-valued vector} \\ \text{s.t.:} & \sum_{k=1}^{m} n_k \leq \ell \\ & \prod_{k=1}^{m} n_k = N \end{cases} \tag{3.53}$$

For the abo problem to be feasible, the given table length must satisfy

$$\ell \geq mN^{1/m} \tag{3.54}$$

which is a straightforward consequence of Proposition 2.1. In what follows we assume that ℓ always satisfies Eq. (3.54).

Problem 3.53 can be easily solved numerically for m = 2 and m = 3; beyond that it becomes quite complicated and no effort has been expended in that direction. However, as noticed earlier, m = 2, 3 will capture most of the realistic table reductions for a fairly large range of N (N up to 10^6, 10^7). Two cases to consider are

<u>m = 2</u>

The objective function, Eq. (3.40) and Eq. (3.50), is

$$E_2 = \frac{4}{(N - 1)N^{1/2}} (N - n_1)(n_1^{1/2} - 1) \tag{3.55}$$

Since the bound E is a decreasing function of ℓ (see Fig. 3.5) then at optimality the constraint $n_1 + n_2 \leq \ell$ is tight. Hence n_1 n_2 are such that

$$\begin{cases} n_1 + n_2 = \ell \\ n_1 n_2 = N \end{cases}$$

whose solution is the pair

$$\frac{\ell \pm \sqrt{\ell^2 - 4N}}{2} \tag{3.56}$$

Finally, the optimal solution can be numerically obtained by choosing the pair n_1, n_2 which minimizes E_2 and satisfies Eq. (3.56).

Fig. 3.11 shows the plots of the minimum E_2 (denoted by E_2 on the graph) with respect to ℓ/N and for a set of values of N. Also represented, is the bound E, Eq. (3.52), obtained with a vector $\underset{\sim}{n}$ which satisfies Proposition 2.1 (i.e., $n_k = N^{1/m}$ $k = 1, \ldots, m$). The main

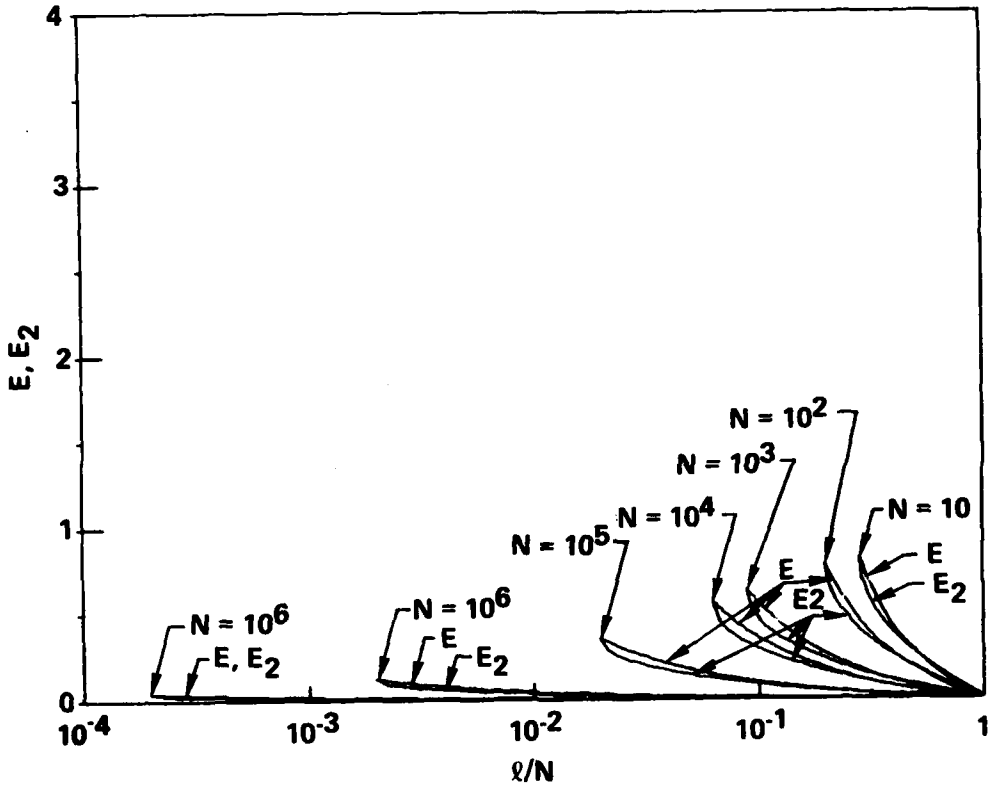


Figure 3.11. Relative Bounds on the Increase in Path Length with a Continuous and a Discrete m, $1 \leq m \leq 2$.

observation is that E is, in general, a slightly larger bound than E_2. For large values of ℓ/N or N, E_2 and E are very small and close to each other. They are equal at $\ell/N = 2N^{-1/2}$ which corresponds to the minimum table length at m = 2 $(n_1 = n_2 = \sqrt{N})$.

<u>m = 3</u>

Similar to the above, at optimality the first constraint is tight, hence

$$\begin{cases} n_1 + n_2 + n_3 = \ell \\ n_1 n_2 n_3 = N \end{cases} \tag{3.57}$$

Then for any feasible vector $\underset{\sim}{n}$ and for a given $n_3 > 0$, the pair n_1, n_2 (or n_2, n_1) must be equal to

$$\frac{1}{2}\left(\ell - n_3 \pm \sqrt{(\ell - n_3)^2 - 4\frac{N}{n_3}}\right) \tag{3.58}$$

Also the expression of the bound for a feasible N is (Eqs. (3.40), (3.50) and (3.57)),

$$E_3 = \frac{4}{(N-1)N^{1/2}}\left[(N - n_1)(n_1^{1/2} - 1) + \left(N - \frac{N}{n_3}\right)\left(\left(\frac{N}{n_3}\right)^{1/2} - 1\right)\right] \tag{3.59}$$

Note that in Eq. (3.59), n_2 was replaced by N/n_1n_3. Then for a given n_3, the minimum of E_3 can be found by evaluating E_3 for the two possible values of n_1, Eq. (3.58). Afterward we minimize the result over n_3. The results of this entirely numerical procedure are shown in Fig. 3.12, (where the minimum is denoted by E_3) Fig. 3.12 also shows the bound E Eq. (3.52) for a continuous m $(1 \leq m \leq 3)$. The curves for the minimum

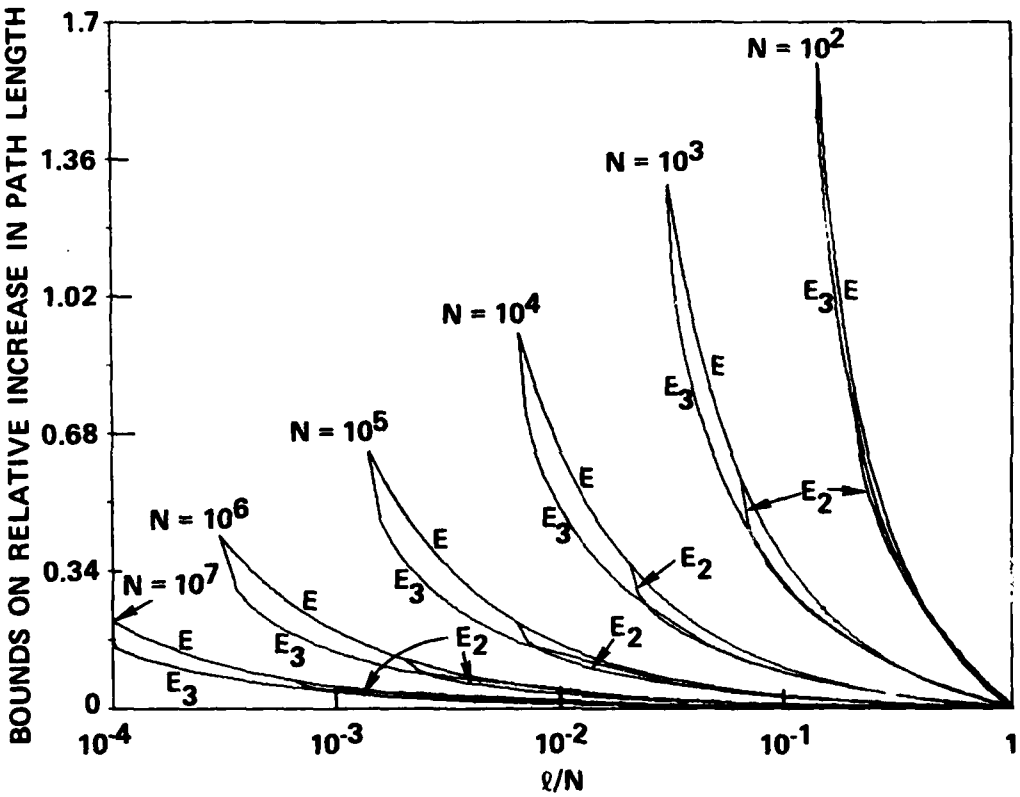


Figure 3.12. Relative Bounds on the Increase in Path Length with a Continuous and a Discrete m, $1 \leq m \leq 3$.

E_2 are also shown here.

Notice that $E = E_3$ for $\ell/N = \frac{3N^{1/3}}{N}$ which corresponds to the minimum table length for a 3-level hierarchy. Similar observations as above apply here. In general for a given ℓ/N, E is slightly worse than E_2 and E_3, except for large values of ℓ/N (i.e., small table reduction) where all the curves are quite close to each other, with sometimes E even a bit better. Also, the lower envelope of E_2 and E_3 corresponds to the overall minimum of Problem 3.53 where m is restricted to either 2 or 3.

The above comparison of E to E_2 and E_3 leads us to the following important remark as to the utilization of a continuous number of levels m.

<u>Remark: Continuous m</u>

The fact that, in general, E is slightly worse than E_2 or E_3 will allow us to always use an MHC which results in a minimal table length ($\ell = mN^{1/m}$) and whose number of levels is continuous, and still evaluate a worst case performance of the MHR schemes. In other words, in the sequel the MHR schemes will always be characterized by E and ℓ as given in Eq. (3.52).

A non-integer value of m may be interpreted as explained in the following example. Assume that we need to operate a network of size N with an MHR but with an allowed maximum increase in path length equal to E_0. For those values of E_0 and N we can find the minimum table length ℓ_0 (as in Fig. 3.9) and the corresponding number of levels m_0, $\ell_0 = m_0 N^{1/m_0}$. Now assume that $m_0 = 1.5$; then we can choose a

number of levels $m = 2$ and a vector $\underset{\sim}{n}$ which minimizes E_2 (Problem 3.5 with $m = 2$ and $\ell = \ell_0$). The minimum value of E_2 is, in general, slightly better than E_0; hence the new MHC with a discrete m satisfies the maximum tolerance allowed.

The above remark is only true for $1 \leq m < 3$ but as noticed previously, that range of m yields most of the table reduction for practical purposes. Also, from the shape of the curves in Fig. 3.12, it appears that these results could be extended to higher values of m.

3.4 Conclusion

In this chapter we examined the effect of hierarchical routing on network path length. Bounds were derived to evaluate a maximum increase in path length for a given table reduction. Furthermore, the bounds demonstrate that no significant increase in path length need be incurred in the limit of a very large network.

The reduction in table length means that more channel capacity and storage are available for the transmission of data traffic in the network. However, those gains were obtained at the expense of longer paths in the network. It is then natural to evaluate the performance of the hierarchical routing in terms of delay and throughput, and to define the region of N where clustering becomes economical. These questions are the object of the next chapter.

CHAPTER 4

STOCHASTIC PERFORMANCE EVALUATION OF THE HIERARCHICAL ROUTING

The previous chapters (2, 3) demonstrated that enormous table reduction can be achieved through hierarchical routing. A shortcoming of those gains was found in the increase of the network path length. This means that the nodal storage and channel capacity recovered as a result of the table reduction may have to be partially or completely (or more) paid back to handle the excess traffic caused by longer network paths. In this chapter we are interested in the trading relations among the table reduction, the nodal storage, the channel capacity, the network size, the throughput and the delay. Several queueing models are developed to capture and exhibit the interrelationships among these variables. The models demonstrate that for some reasonable cost and performance constraints and for a class of symmetrical and distributed networks, the non-hierarchical routing becomes infeasible for N (network size) beyond some "critical" value; whereas, on the other hand, they show that hierarchical routing operating with an appropriate table length is capable of maintaining a fairly good network performance for fairly large values of N.

In what follows, first we present the analysis of network performance under fixed routing, and a fundamental assumption upon which we will build the models mentioned above. This assumption typically results in a mapping of hierarchical and non-hierarchical adaptive routing into deterministic (fixed) routing. Then four models will be

gradually developed to deal with different subsets of the variables above; listed by order of complexity, they are:

Model with no updates and no storage limitation

Model with updates and no storage limitation

Model with no updates and with storage limitation

Model with updates and with storage limitation

4.1 General Considerations

The objective here is to first introduce the original model developed by Kleinrock [KLEI 64] for delay analysis in store-and-forward (S/F) computer networks. Then we introduce a class of symmetrical networks and a main assumption which will serve as a framework for the four models to be developed in this chapter.

4.1.1 Delay Analysis in S/F Computer Networks: Kleinrock's Model

As described in Section 2.1, in a S/F net, a message enters the net at the source node, visits several nodes along the communication path, and leaves the net at the destination node. The time a message spends at each node is the sum of nodal processing time (for routing purposes), of queueing time due to interfering messages, and the transmission time. A very important performance measure of an S/F net is the total average delay, T, a message spends in the network. T is then defined as

$$T = \frac{1}{\Gamma} \sum_{i,j \in S} \gamma_{ij} z_{ij} \tag{4.1}$$

where: γ_{ij} = average message rate from source i to destination j

z_{ij} = average message delay from i to j

Finally Γ is the total input rate (throughput),

$$\Gamma = \sum_{i,j \varepsilon S} \gamma_{ij} \tag{4.2}$$

A straightforward application of Little's result leads to the expression of T in terms of the individual channel delays [KLEI 64],

$$T = \sum_{i=1}^{NA} \frac{\lambda_i}{\Gamma} t_i \tag{4.3}$$

where: λ_i = average traffic rate on channel i

t_i = average nodal processing plus queueing plus transmission on the ith channel.

Unfortunately, we are not able in general to evaluate t_i and λ_i. However, if we make the following assumptions [KLEI 64]:

a. external Poisson arrivals

b. exponential message length distribution (identical for all messages)

c. single packet messages

d. error free channels

e. no nodal delay

f. independence assumption

g. deterministic routing

h. infinite nodal storage

With such assumptions (to be discussed later), the S/F net can be modelled as a network of queues of the Jackson type [JACK 57].

In particular each queue behaves as an independent M|M|1 queue and the average delay on channel i, t_i, is given by

$$t_i = \frac{1}{\mu C_i - \lambda_i} \tag{4.4}$$

where: $\frac{1}{\mu}$ = average message length [kbits/messg]

C_i = capacity* of channel i [kbits/sec]

The average rates λ_i $i = 1, \ldots, NA$, can be easily computed, given the underlying deterministic routing.

The substitution of Eq. (4.4) into Eq. (4.3) gives

$$T = \frac{1}{\Gamma} \sum_{i=1}^{NA} \frac{\lambda_i}{\mu C_i - \lambda_i} \tag{4.5}$$

A simple relationship exists between the total internal traffic Λ, the total external traffic Γ, and the average weighted (with traffic) path length $\bar{n}$.

$$\bar{n} = \Lambda/\Gamma \tag{4.6}$$

where

$$\Lambda = \sum_{i=1}^{NA} \lambda_i \tag{4.7}$$

Moreover in a 0-load condition, i.e., $\gamma_{jk} \to 0 \quad \forall\, j, k$, the total average delay becomes,

$$T_0 = \bar{n} \sum_{i=1}^{NA} \frac{\lambda_i/\Lambda}{\mu C_i} \tag{4.8}$$

where λ_i/Λ is a constant which depends only on the routing scheme.

---

*Not to be confused with the notation used earlier for cluster.

Discussion of the Above Assumptions

Discussions of the above assumptions can be found in [KLEI 64], [FULT 72], [GERL 73A], along with some further extensions of the above model. In particular, we will discuss the last three assumptions in order to motivate some of the models to be considered in this chapter.

Independence Assumption: [KLEI 64], [FULT 72]

This assumption is quite acceptable as long as the network does not contain long chains with no interfering traffic.

Deterministic Routing Assumption:

As mentioned in Section 2.1.1, typically, a network is designed using deterministic routing [GERL 73A] and then operated using adaptive routing. Consequently the validity of this assumption depends critically on the difference of behavior between deterministic and adaptive techniques. A thorough investigation on this subject can be found in [FULT 72]. Fultz finds a close agreement between the two techniques. He showed in a 19-node net application that the difference in delays obtained with a well chosen adaptive technique and with a near-optimal deterministic technique is less than 5 to 10% almost until saturation. The "adaptive" delay tends to be higher, mainly because of the line overhead utilized by the update traffic.

Fultz's study is, of course, dependent on the particular adaptive policy and network considered. However, it demonstrates that adequate adaptive policies can be devised to achieve a performance very close to that of a near-optimal deterministic policy.

Further refinement of the above assumption can be realized by

including the line overhead due to the update traffic. This consideration will become crucial when dealing with large nets, as we will see later.

Infinite Storage Assumption

Nodal storage limitation can cause blocking and, consequently, retransmission (ARPANET) or loss of messages, hence an increase in delay or a decrease in throughput. Moreover, it can lead to serious network degradation and even deadlocks [FULT 72], [KAHN 71], [KLEI 74].

Zeigler [ZEIG 71] investigated some theoretical aspects related to the dynamics of cliques of blocked nodes in a symmetrical network environment.

It is generally accepted [FRAN 70], [COLE 71], that with reasonable storage size, the above assumption is fairly accurate. However this assumption becomes critical if the nodal storage becomes small. This situation is very likely to occur in a large network environment if the routing tables are not reduced to a reasonable length. A model is presented in Section 4.4 to precisely deal with this question.

4.1.2 A Class of Symmetrical Networks

The delay analysis performed in the above model relies on the knowledge of the input rates (λ_i's) to the individual channels in the network. As mentioned earlier, the λ_i's can be determined once we know the deterministic routing policy and the traffic requirement. Moreover, if we know the channel capacities then the λ_i's can be computed to lead to a minimal delay T [FRAT 73].

A shortcoming of the above numerical procedures is that, in general, they hide the interrelationships existing among the different design variables (traffic requirement, channel capacities, network topology and average delay). Fortunately, for some symmetrical networks (see below) a simple analytical relationship exists among the above variables.

The class of nets to be considered in this chapter is composed of all the nets which belong to the family of nets presented in Section 3.3.1, and which also satisfy the following properties.

i. All nodes are equivalent with respect to the topology of the network. Hence they are of equal degree, R.

ii. All channels are of equal capacity C.

iii. All external input traffic rates are equal, i.e.,

$$\gamma_{jk} = \gamma \quad \forall\, j, k \in S \tag{4.9}$$

As an example, torus nets (see Appendix A) fall into this category. The above properties lead to a simple delay expression.

4.1.2.1 Average Delay

For this class of nets, the following relations exist:

Number of (simplex) channels: $NA = R \cdot N$ (4.10)

Total external traffic: $\Gamma = N(N - 1)\gamma$ (4.11)

Furthermore, it is obvious that with this particular topological structure, capacity assignment and traffic requirement, the optimal flow assignment [KLEI 64], [GERL 73A] is a shortest path routing. The selection of the particular shortest paths (in case that more than one

exists) must result in perfectly balanced flows, i.e.,

$$\lambda_i = \lambda \qquad i = 1, \ldots, NA \tag{4.12}$$

Consequently the network path length $\bar{n}$ becomes the average shortest path length h, defined in Appendix A, Eq. (A.2), then from Eq. (4.6)

$$\frac{\Lambda}{\Gamma} = h \tag{4.13}$$

Also the total internal traffic Λ (Eq. (4.7)), becomes

$$\Lambda = NA\ \lambda \tag{4.14}$$

Combining the two equations above, we arrive at

$$\lambda = h\ \frac{\Gamma}{NA} \tag{4.15}$$

If we let t denote the average delay on any channel, then the total average delay becomes, Eq. (4.3),

$$T = ht \tag{4.16}$$

Moreover, as a consequence of Eqs. (4.4), (4.5), (4.12) and (4.15)

$$t = \frac{1}{\mu C - \lambda} \tag{4.17}$$

and

$$T = \frac{1}{\frac{\mu C}{h} - \frac{\Gamma}{NA}} \tag{4.18}$$

This is the result we were aiming at, which simply relates the delay T, the traffic Γ, the channel capacity C and the network path length h.

Eq. (4.18) shows that the net is equivalent (with respect to delay) to a single $M|M|1$ queue with an input rate of Γ/NA and a service

rate $\mu C/h$. This observation leads to the definition of the network utilization

$$\rho = \frac{h}{\mu C}\frac{\Gamma}{NA} \tag{4.19}$$

The above definition will be used in the rest of the chapter, mainly for normalization purposes.

4.1.2.2 A Scaling Scheme

Since the main objective here is to study routing in large nets, it is necessary to specify the structure of those large nets with respect to the size N, in some continuous way. Any such specification will be referred to as a scaling scheme (strategy).

As the network grows, a main objective of a scaling strategy could be to maintain the same average delay T for a reasonable increase of the total traffic Γ and of the network cost (channel capacity cost). The total traffic Γ may be reasonably assumed to increase linearly with the number of nodes. Which, due to the uniform traffic condition ($\gamma_{jk} = \gamma$), is equivalent to assuming that the total input rate per node is maintained constant. Let g denote the total input rate per node divided by the degree of a node,

$$g \triangleq \frac{(N-1)\gamma}{R} \tag{4.20}$$

Note that γ will be adjusted in order to maintain g constant. Consequently and because of Eqs. (4.10), (4.11) and (4.18),

$$\Gamma = NRg = NAg \tag{4.21}$$

$$T = (\mu C/h - g)^{-1} \tag{4.22}$$

In order to maintain a constant delay T_0, with the above traffic requirement, the channel capacity must be such that: (Eq. (4.18)).

$$C = \frac{h}{\mu}\left(\frac{1}{T_0} + g\right) \tag{4.23}$$

which says that C must grow like the network path length h. If we let C_0 be a constant capacity, then the appropriate capacity scaling is

$$C = hC_0 \tag{4.24}$$

Notice that the ratio of total capacity to the total traffic is equal to hC_0/g. This indicates that for a linear growth of N and Γ, the capacity required per message need only grow like h (e.g., $h = \frac{\sqrt{N}}{2}$ for torus nets).

From the above considerations emerge two scaling schemes:

$$\text{Primal scaling:} \quad \begin{cases} C = hC_0 \\ T = T_0 \end{cases} \tag{4.25}$$

$$\text{Dual scaling:} \quad \begin{cases} C = hC_0 \\ \Gamma = NAg_0 \end{cases} \tag{4.26}$$

The outcome of the primal scaling is a traffic $\Gamma = NAg$ where $g = \mu C_0 - 1/T_0$, and conversely the outcome of the dual scaling is a constant delay $T = (\mu C_0 - g_0)^{-1}$. Such outcomes will not be true when dealing with network models which take into account the updates and/or the storage limitation.

4.1.3 Continuous Modelling of the Hierarchical Routing

The objective here is to summarize the key variables governing the behavior of hierarchical routing in a continuous (homogeneous) approach.

Recall that a one level hierarchical routing (CER or OBR) is simply the non-clustered routing (NCR) defined in Chapter 3. As a consequence and throughout this chapter we will refer to the three routing schemes (CER, OBR, NCR) as hierarchical routing schemes with the understanding that $m = 1$ refers to the NCR. Moreover, as observed from previous considerations, the underlying clustering structure is completely defined, knowing the number of levels m. More specifically, the value of m determines exactly the length of the table $\ell = mN^{1/m}$; subsequently, we can use the remark at the end of Section 3.3.3.2 to discretize m and to find the corresponding degree vector $\underset{\sim}{n}$. Notice that if $m = 1$ then $\ell = N$, i.e., a full length table is utilized, which corresponds to an NCR. To summarize, the underlying clustering structure of the hierarchical routing (OBR, CER, NCR) is characterized by the single continuous variable m. For that reason m will be referred to as the degree of clustering

Chapter 3 showed that the reduction in table length from $\ell = N$ to $\ell = mN^{1/m}$, due to the hierarchical routing, is accompanied by an increase in path length from h to h_c. h_c is characterized by Proposition 3.1, which subsequently provides us with the basis for an algorithmic computation of h_c, given a specific network and a specific MHR scheme. Fortunately, the bounds derived in Chapter 3 will allow us to undertake a worst and/or best case analytical performance

evaluation of the hierarchical routing for the class of networks considered here. Recall that those networks constitute a subset of the family studied in Chapter 3, hence we will be able to use the explicit expressions for the relative bounds E or F as derived in Eqs. (3.41) and (3.47).

Notice that, similar to the above, the bounds rely on the continuous behavior of m, and more importantly, they are tight at $m = 1$. These last observations will allow us to use the above continuous approach in dealing with the hierarchical and non-hierarchical routing schemes. The comparison between those two schemes simply reduces to the comparison of a hierarchical routing with $m = 1$ (NCR) to one with $m > 1$ (CER or OBR).

In summary, knowing the degree of clustering m, the increase in path length is characterized by the relative bounds E or F. In general E will be chosen, unless specified otherwise. Hence,

$$h \leq h_c \leq (1 + E)h \tag{4.27}$$

and

$$m = 1 \implies E = 0$$

Even with the above simple specifications of the hierarchical adaptive routing scheme (i.e., m, E) the queueing analysis is still far too complicated. This is true for any adaptive scheme because of its dynamic nature. In the face of these difficulties, we will rely on observations made about the deterministic routing assumption in Section 4.1.1 to justify the modelling of a hierarchical adaptive routing by an "equivalent" deterministic routing; hence the assumption below:

Assumption 4.1

a. The performance of an adaptive hierarchical routing is the same as that of a fixed routing policy whose routes satisfy Proposition 3.1. That is, the length of the "fixed routing" paths are equal to the minimum estimated path lengths as obtained with an MHR.

b. The fixed routing specified above and operating on the class of symmetrical nets considered here, results in equal loads on all channels.

Part (a), again motivated by an earlier remark on deterministic routing, will become more accurate when we include in the model, based on fixed routing, the line and storage utilization due to the adaptive routing. Moreover, if the main objective is to compare hierarchical with non-hierarchical routing, then this assumption appears to be quite acceptable.

Part (b) is motivated by the quite symmetrical structure of the networks considered here, and also by the fact that the main objective of an adaptive policy is to balance the flows over all the channels in the net.

Notice that, because of the above assumption, the NCR (MHR with m = 1) is modelled by the shortest path fixed routing which, as observed in Section 4.1.2.1, leads to the optimal flow assignment for this class of nets.

Conclusion:

As a result of the above considerations, a hierarchical routing is characterized by the degree of clustering, m, and the path length,

h_c, and can be modelled by a balanced fixed routing resulting in paths of the same lengths.

4.2 A Queueing Model with No Updates and No Storage Limitation

In this section, we still temporarily relate the gains obtained by hierarchical routing strictly to the relative table length ℓ/N. In other words, we consider an ideal situation where infinite nodal storage is available and where the line capacity used up by the update information is still negligible. This is similar to the study in Section 3.3 except that we now intend to look at the degradation of the network performance in terms of delay or throughput.

Based on the previous general conditions (Section 4.1) and the above remarks, this situation can be modelled by Kleinrock's model (Section 4.1.1) where the fixed routing is as specified in Assumption 4.1.

The delay analysis, for our class of symmetrical nets, is reduced to the one performed in Section 4.1.2.1 except that h is to be replaced by h_c. As a result the throughput-delay relation becomes (Eq. (4.18)).

$$T_c = \frac{1}{\mu C/h_c - \Gamma_c/NA} \tag{4.28}$$

Notice the change in notation from Γ to Γ_c and T to T_c. The subscript c is added mainly to differentiate between clustered and non-clustered routing for later comparisons. The notation Γ, T will be strictly reserved for the NCR and Γ_c, T_c for all hierarchical routing; of course,

if $m = 1$ then $\Gamma_c = \Gamma$ and $T_c = T$. Similar to Eq. (4.16) and Eq. (4.21) we will also let

$$T_c = h_c t_c \quad , \qquad \Gamma_c = NAg_c \tag{4.29}$$

where t_c and g_c are now variables.

The behavior of the hierarchical routing may now be studied assuming either the primal or dual scaling schemes (see Section 4.1.2.2) in order to specify the growth of our class of nets.

The numerical applications throughout this chapter will assume the coefficients a, b, c, v as obtained for torus nets (Eq. 3.50).

4.2.1 Degradation in Throughput at Constant Delay (Primal Scaling)

Recall that the primal scaling (PS) is characterized by $C = hC_0$ and $T = T_c = T_0$ (Eq. (4.25)) and also for an NCR $h_c = h$, hence from Eq. (4.28) the ratio of throughputs with and without clustering is

$$\frac{\Gamma_c}{\Gamma} = \frac{h\mu C_0/h_c - 1/T_0}{\mu C_0 - 1/T_0}$$

from the above expression the effect of clustering can be seen in the reduction of the line capacity by a fraction h/h_c.

We can now directly apply Proposition 3.4 to state a similar asymptotic result. Namely, as N, the number of nodes, goes to infinity the throughput at constant delay, obtained with an MHR (CER, OBR) with a fixed m, approaches that of an NCR while the relative table length (ℓ/N) approaches zero.

As for the continuous behavior of Γ_c/Γ versus ℓ/N we use Eq. (4.27) to derive the bounds below.

$$LB(\Gamma_c/\Gamma) \triangleq \frac{\frac{\mu C_0}{1+E} - \frac{1}{T_0}}{\mu C_0 - \frac{1}{T_0}} \leq \frac{\Gamma_c}{\Gamma} \leq 1 \tag{4.30}$$

LB( ) is used to denote the lower bound on the variable in parenthesis; similarly, UB( ) will be used to denote an upper bound.

Let us examine the behavior of Γ_c/Γ by plotting its bounds given in Eq. (4.30). Recall that throughout this chapter Eq. (3.52), giving E and ℓ/N, will be used in the numerical examples. This results in the curves shown in Figs. 4.1 - 4.3. The interpretation of those polts is quite similar to that provided for Figs. 3.5 - 3.9 in Section 3.3.3. We reemphasize the fact that initial (substantial) table gains (small[1] m) can be obtained with a relatively small degradation in throughput. Whereas larger table reduction (large[1] m) may drive the lower bound to zero. The asymptotic property is nicely illustrated in the sharp behavior of the lower bound, for a large N, in Fig. 4.2. Also, Fig. 4.3 shows that as N increases the cost incurred with a fixed relative table length ℓ/N goes to zero.

We note also that the curves in Figs 4.1 and 4.2 meet at the point $\ell/N = 1$, $LB(\Gamma_c/\Gamma) = 1$. That point corresponds to m = 1 where the bound is tight.

[1]Recall from Chapter 2 that most of the table reduction is obtained for small values of m, and the remaining reduction up to the global minimum table length is obtained with quite a bit larger m (up to m_*).

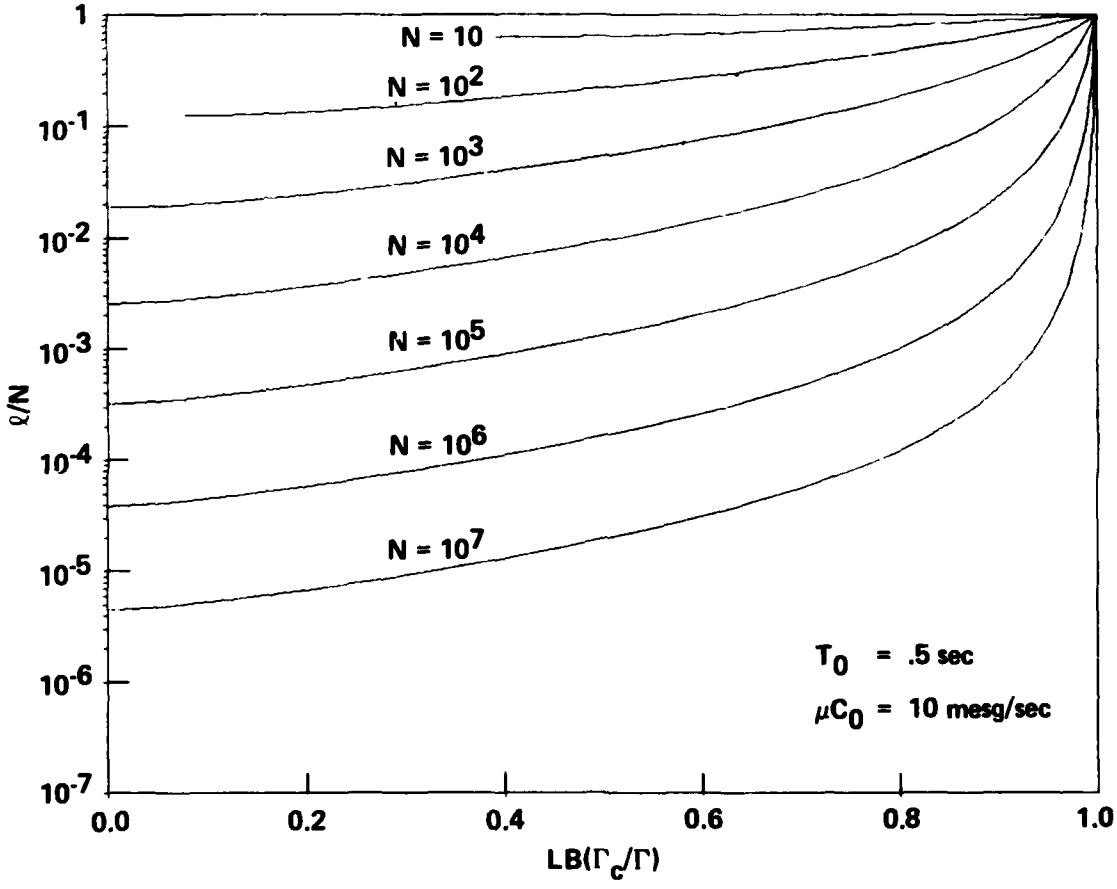


Figure 4.1. Degradation in Throughput at Constant Delay, LB(Γ_c/Γ); Model with no Updates and no Storage Limitation.

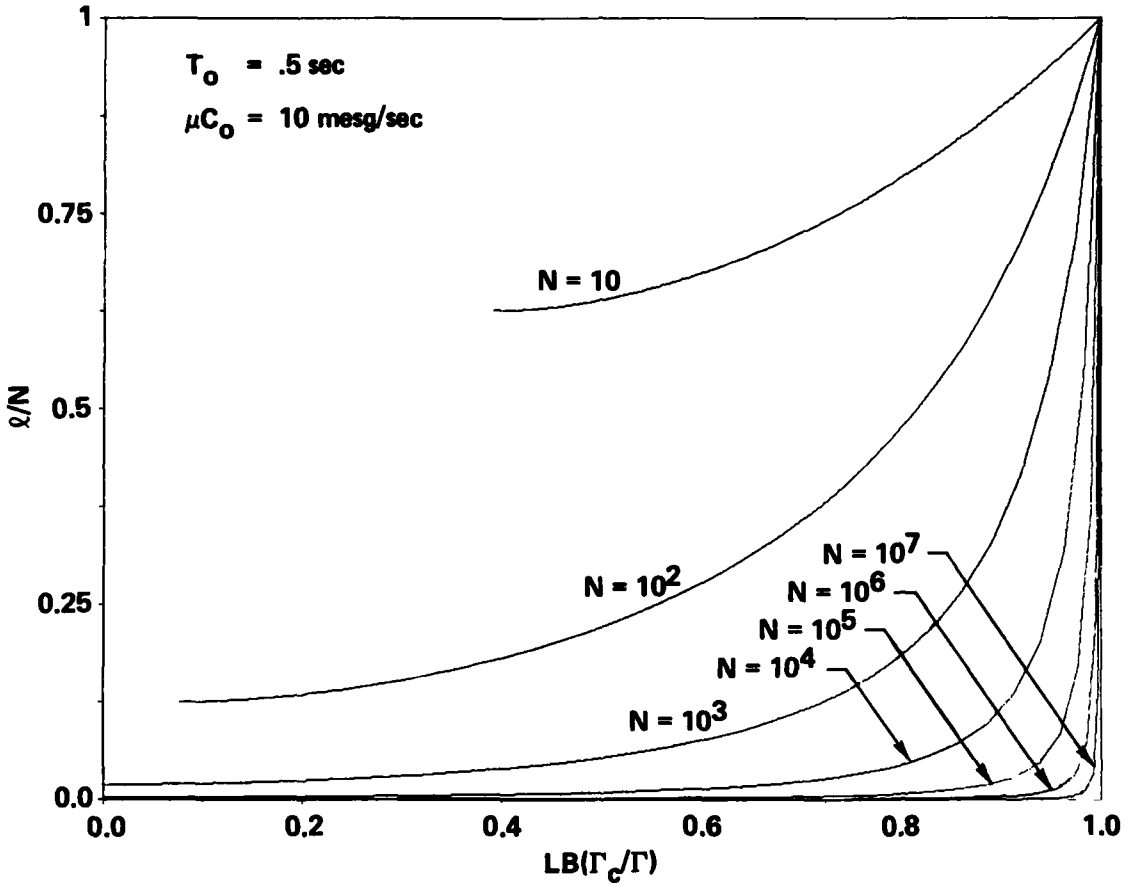


Figure 4.2. LB(Γ_c/Γ) versus ℓ/N, Linear Axis.

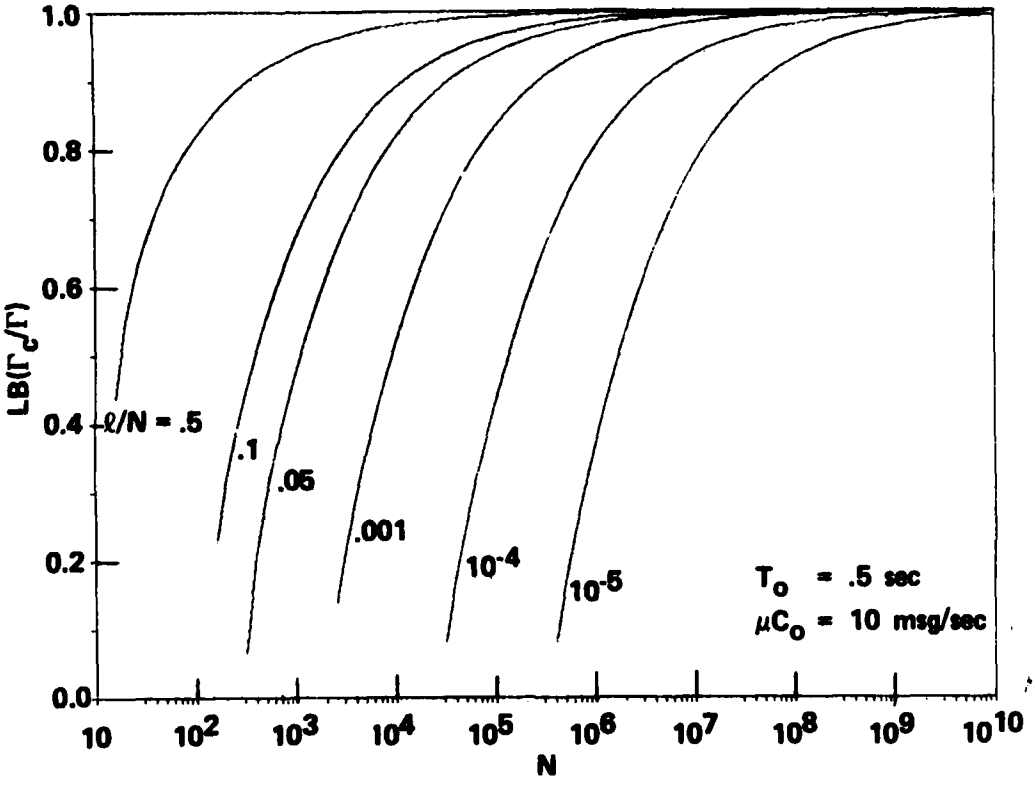


Figure 4.3. Degradation in Throughput at Constant Relative Table Length.

4.2.2 Degradation of Delay at Constant Throughput per Node (Dual Scaling)

We now intend to study the dual behavior by applying the dual scaling, whose purpose is to maintain Γ/NA constant and to let T_c vary. Let $g = g_c = g_0 = \Gamma/NA$ and $C = hC_0$; then the ratio of delays T_c/T is from Eq. (4.28),

$$\frac{T_c}{T} = \frac{\mu C_0 - g_0}{\frac{h}{h_c}\mu C_o - g_0} \tag{4.31}$$

Immediately we obtain the dual asymptotic result whereby as N goes to infinity the network delay obtained with an MHR (m fixed) approaches that of an NCR, while the relative table length goes to zero.

As for the general behavior of T_c/T versus ℓ/N, from Eq. (4.27)

$$1 \leq \frac{T_c}{T} \leq UB\left(\frac{T_c}{T}\right) \triangleq \frac{\mu C_0 - g_0}{\frac{\mu C_0}{1+E} - g_0} \tag{4.32}$$

The inverse of the above bounds are plotted in Fig. 4.4, in order to show the behavior of T/T_c; dual properties are exhibited. We also note that in some regions $\mu C_0/(1+E) < g_0$, hence the corresponding degree of clustering m is infeasible.

4.3 A Queueing Model with Updates and No Storage Limitation

Our previous models expressed the gains obtained from the hierarchical routing strictly in terms of the relative table length ℓ/N. They left out the further (significant) savings incurred in line capacity and nodal storage. As a result of this quite idealistic situation of abundance in capacity and storage, the NCR showed a better

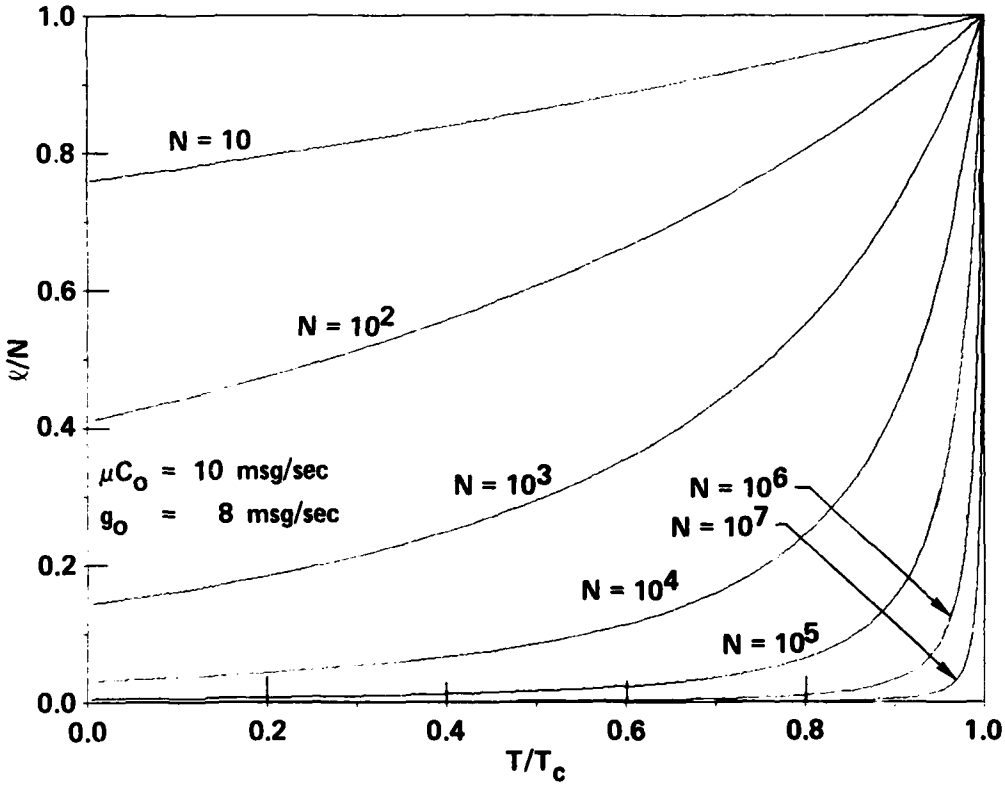


Figure 4.4. Degradation in Delay at Constant Throughput per Node; Model with no Updates and no Storage Limitation.

network performance than the MHR except at the limit of very large nets where they were quite equivalent even for substantial table reductions. This last (asymptotic) property makes us believe intuitively that in a less idealistic situation (storage and capacity limitations) the MHR with the right degree of clustering will eventually (in terms of size) outperform the NCR. Moreover we can already observe that if no clustering is introduced, the linear growth of a full length table (N entries) with N will, at a certain point, use up all the nodal storage and the line capacity (for the exchange of updates), hence driving the network throughput to zero.

In this section we intend to account for the traffic generated by the routing updates (in the evaluation of the λ_i's) while keeping the infinite storage assumption.

As noticed earlier, the average delay in our class of symmetrical nets is very simply related to the average delay at any channel ($T = ht$); therefore, we will first analyze a single channel and then generalize to the whole net.

4.3.1 Priority Model for a Channel

A simple and realistic Head of Line (HOL) model [KLEI 76] is considered here, mainly to capture the effects of updates on the average time spent by a data[1] message waiting to be transmitted on a channel. We assume that updates are originated at regular intervals of time (motivated by ARPANET). Aperiodic updates may be modelled by a "no

[1] A data message is differentiated from an update message.

update" model (Sections 4.2, 4.4) or by a certain distribution governing their generation times. The latter possibility can be easily included in the model below for such distribution as the Poisson distribution. (However, no work has been done in that direction.)

In summary, our model for a channel consists of a single queue operated with a HOL priority discipline and the following traffic characteristics.

i. Update traffic: Deterministic arrival process of rate λ_u. Constant message length $1/\mu_u$ [kbt/messg].

ii. Data traffic: Poisson arrival process of rate λ. Exponential message length of mean $1/\mu$ [kbt/messg].

iii. Queue discipline: HOL preemptive resume between data and update traffic, with a higher priority for updates. FCFS (first-come-first-serve) within each priority.

iv. Channel capacity: C [kbt/sec].

The "preemptive resume" assumption in (iii) is introduced to further simplify the analysis of the model, which otherwise becomes quite complex.

The above model is slightly different from the one analyzed by Kleinrock [KLEI 76] where he considers the arrival processes of all the types of customers (messages) to be governed by a Poisson distribution. However, his methodology can still be used here in order to derive the average time in system for a data message.

With regard to the update traffic, it simply sees a $D|D|1$ system; hence as long as $\lambda_u < \mu_u C$ ($\lambda_u \geq \mu_u C$ means that more than the total capacity is required by the updates, thus the routing becomes

infeasible) there is no queueing of update messages; whereas, an arriving data message will incur a delay from the message (data or update) already in service, from data messages already in the queue and from updates arriving during its system time (see Fig. 4.5)

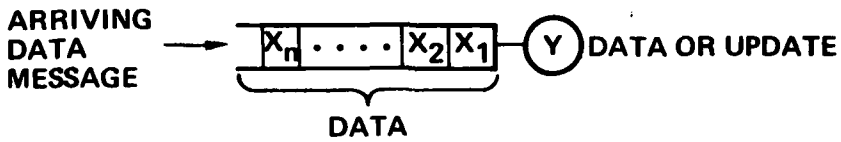


Figure 4.5. State of the Queue as seen by an Arriving Data Message.

Let us now evaluate the different components involved in the delay of a data message.

1. Delay due to messages already in queue. Let $\mathscr{E}$ denote the expectation operator, *n be the number of data messages* (updates do not join the queue) that our arriving data message finds in the queue, and X_i be the service required by the ith message; then the average wait incurred is $\mathscr{E}(\Sigma_i X_i)$.

Clearly, $X_2, X_3, \ldots, X_n$ are identically distributed (exponential distribution of mean $1/\mu$); the question arises as to the distribution of X_1, the service (entire or remaining) time of the data message in the front of the queue (see Fig. 4.5) which, as such, could have been preempted several times in the past. Fortunately, because of our exponential distribution assumption for data messages, the remaining service time is also exponentially distributed no matter how many times that message has been preempted. Hence $\mathscr{E}(\Sigma_i X_i) = \bar{n}/\mu C$,

where $\bar{n}$ is the average number in queue as seen by our arriving message. Since data arrivals are Poisson, then the distribution of n is the same as the distribution of the number of messages in the queue at any arbitrary time. Therefore, if t is the average time in system (queueing + service) for a data message, then because of the previous remark and from Little's results, $\bar{n} = \lambda(t - 1/\mu C)$.

2. Delay due to the message in service. Let Y be the residual life (remaining service time) of the message in service. $\mathscr{E}(Y)$ is the contribution in delay that we need to evaluate. Conditioning on the type of message in service, it is clear that

$$\mathscr{E}(Y) = \mathscr{E}[\text{residual life of data message}] \cdot \frac{\lambda}{\mu C}$$

$$+ \mathscr{E}[\text{residual life of update message}] \cdot \frac{\lambda_u}{\mu_u C}$$

Hence,

$$\mathscr{E}(Y) = \frac{\lambda}{\mu C} \cdot \frac{1}{\mu C} + \frac{\lambda_u}{\mu_u C} \cdot \frac{1}{2\mu_u C}$$

3. Delay due to updates arriving while in system. A data message spends an average time t in the system. During that time, on the average, $\lambda_u t$ update messages arrive and get serviced; hence the average wait time incurred is $\lambda_u t \; 1/\mu_u C$. Notice that even though update arrivals are deterministic, their relative positions with respect to data messages are completely random due to the Poisson distribution of data arrivals.

Finally summing up all the waits incurred by our arriving message and its own service time, we arrive at

$$t = \frac{1}{\mu C} + \left[\lambda t - \frac{\lambda}{\mu C}\right]\frac{1}{\mu C} + \frac{\lambda}{[\mu C]^2} + \frac{\lambda_u}{2(\mu_u C)^2} + \frac{\lambda_u}{\mu_u C}\, t$$

Hence

$$t = \frac{\dfrac{1}{\mu C} + \dfrac{\lambda_u}{2[\mu_u C]^2}}{1 - \dfrac{\lambda}{\mu C} - \dfrac{\lambda_u}{\mu_u C}} \tag{4.33}$$

If we set $\lambda_u = 0$ in the above equation (i.e., if we neglect updates) then we arrive at the original expression for t, Eq. (4.17). The difference between the two equations illustrates the effect of the updates.

4.3.2 Network Model

Similar to the previous model, Assumption 4.1 results in equal channel loads of data messages λ ($\lambda = h_c\Gamma/NA$). Moreover because of the periodic update assumption, all channels will receive an equal rate of updates λ_u.

With regard to the delay analysis for our class of symmetrical networks, Eqs. (4.10) - (4.16) are still valid when we replace h by h_c. Using a similar notation as in Section 4.2 (t, t_c, h, h_c, T, T_c, Γ, Γ_c, g, g_c), we arrive at the throughput-delay relation below which characterizes the performance of the hierarchical routing.

$$T_c = h_c \frac{1 + \dfrac{\mu}{\mu_u}\dfrac{\lambda_u}{2\mu_u C}}{\mu C - h_c\dfrac{\Gamma_c}{NA} - \dfrac{\mu}{\mu_u}\lambda_u} \tag{4.34}$$

Recall that the size of an update message, $1/\mu_u$, is fixed and proportional to the length ℓ ($\ell = mN^{1/m}$) of the routing table. Hence, it is of the form $1/\mu_u = \varepsilon\ell$ where ε is the inverse of the number of entries which add to 1 kbt. As an example, in the ARPANET, an entry requires 16 bits of storage, hence $\varepsilon = .016 = 1/62.5$. For further normalization with respect to the average data message, $1/\mu$, we choose ε such that

$$\frac{1}{\mu_u} = \frac{\varepsilon\ell}{\mu} \tag{4.35}$$

(In the numerical examples we will, in general, choose $1/\mu = 1$ kbt and $\varepsilon = 1/64$.)

The behavior of the hierarchical routing may now be studied for networks whose growth is governed by our scaling schemes.

4.3.3 Performance Evaluation of the Hierarchical Routing

The primal and dual scaling schemes, Eqs. (4.25) - (4.26) need further specifications as to the choice of the update rate λ_u in terms of the network parameters.

4.3.3.1 Scaling of λ_u

Let us recall that the main purpose of the routing updates is to provide the routing decision algorithm with a good estimate of the state of the network. Since at each update, exchange of information (not necessarily synchronized among all nodes) occurs only between neighboring nodes, then the propogation of a change occuring in a certain region of the net to another region might require a number of updates equal to the distance separating the two regions. Consequently,

as the network grows and if we wish a change, conveyed through the exchange of updates, to reach remote areas within reasonable time, then it is necessary to increase the update rate as N increases. We may also argue the "very" remote areas would not be as concerned with that change as the closer ones, thus the update rate probably need not increase as fast as N. A realistic compromise would consist in the use of higher update rates (as N increases) but only to propagate less and less information about a region as we move away from that region. This remark is again a key motivation behind the hierarchical routing.

From the above considerations emerge three possible specifications for λ_u.

i. $\lambda_u = \lambda_u^0 = \text{constant}$

ii. $\lambda_u = h\lambda_u^0 = aN^v\lambda_u^0$ $\qquad \frac{\sqrt{N}}{2}\lambda_u^0$ for a torus

iii. $\lambda_u = aN^{v/2}\lambda_u^0$ $\qquad \frac{N^{1/4}}{2}\lambda_u^0$ for a torus

Choice (i) represents a worst-case condition whereby the update rate is insensitive to the increase in network size.

Choice (ii) appears to be more intuitive since the update information needs on the order of h (average path) periods to percolate throughout the net.

Choice (iii) is a compromise between the two above; it indicates that routing information need not percolate as fast in the entire net, but only within a certain area comprising roughly $N^{1/2}$ nodes.

We consider these three choices below.

4.3.3.2 Throughput at Constant Delay (Primal Scaling)

With the primal scaling, $C = hC_0$ and $T = T_0$, the network throughput is (Eqs. (4.34) and (4.35))

$$\frac{\Gamma_c}{NA} = \frac{h}{h_c}\mu C_0 - \frac{1}{T_0} - \frac{\varepsilon\ell\lambda_u}{h_c} - \frac{\lambda_u}{2\mu C_0 T_0}\frac{\varepsilon^2\ell^2}{h} \tag{4.36}$$

For any routing to be feasible, the right hand side of the above equation must be positive.

Asymptotic Behavior

As the number of nodes goes to infinity and under the constraint on λ_u below (Eq. (4.37)), for a hierarchical routing to be feasible, its number of levels m must be greater than or equal to a certain value, m_0. Moreover, the asymptotic throughput (Eq. (4.38)) shows that at the limit, with any feasible hierarchical routing (m fixed), the effect of the updates on the channel utilization becomes negligible.

The condition on λ_u is such that there exists m_0 which satisfies

$$\lim_{N\to\infty} \frac{\varepsilon^2 m_0^2 N^{2/m_0}\lambda_u}{2\mu C_0 T_0 a N^v} = 0 \tag{4.37}$$

This condition is motivated by the fact that the predominant term (as $N \to \infty$) in Eq. (4.36) must go to zero as $N \to \infty$ for any hierarchical routing to be feasible. (Notice that we replaced, in that term, ℓ by $mN^{1/m}$ and h by aN^v.)

The proof of the above fact follows very simply:

Let m be fixed and $m \geq m_0$; then from Proposition 3.4, $\lim_{N \to \infty} h/h_c = 1$, and from Eq. (4.37), $\lim_{N \to \infty} \frac{\varepsilon \ell \lambda_u}{h_c} = 0$; hence

$$\lim_{N \to \infty} \frac{\Gamma_c}{NA} = \mu C_0 - \frac{1}{T_0} \tag{4.38}$$

which terminates the proof of the above fact.

The above limiting throughput is equal to that obtained in our previous model where no updates are considered. It is a more realistic result because now only a feasible routing ($m \geq m_0$) can achieve that performance. This brings us to a consideration of the range of m_0 in terms of the specific choice of λ_u. Let λ_u be of the form $\lambda_u = N^x$ ($0 \leq x \leq 1$); then for the limit in Eq. (4.37) to be true, m_0 must be such that

$$\frac{2}{m_0} + x - v < 0$$

Thus

$$0 < \frac{2}{m_0} < v - x$$

For the above relation to be possible, x must be less than v with this condition

$$m_0 > \frac{2}{v - x}$$

Applying this result to our selected scaling schemes for λ_u, and using the fact that $0 < v \leq 1$, we arrive at

$$\lambda_u = \lambda_u^0 \Rightarrow x = 0 \Rightarrow m_0 > 2/v \geq 2$$

$$\lambda_u = aN^v \lambda_u^0 \Rightarrow x = v \qquad \text{infeasible}$$

$$\lambda_u = aN^{v/2} \lambda_u^0 \Rightarrow x = v/2 \Rightarrow m_0 > 4/v \geq 4$$

As a consequence only the first and third schemes may yield a feasible hierarchical routing. Moreover a non-hierarchical routing is always infeasible at the limit of very large networks.

The above properties are illustrated in the numerical study below.

General Behavior

Similar to Section 4.2.1, a lower and upper bound on the throughput can be derived using Eqs. (4.27) and (4.36).

$$LB\left(\frac{\Gamma_c}{NA}\right) = \frac{1}{1+E}\left[\mu C_0 - \frac{\varepsilon \ell \lambda_u}{h}\right] - \frac{1}{T_0} - \frac{\lambda_u \varepsilon^2 \ell^2}{2\mu C_0 T_0 h} \qquad (4.39)$$

$$UB\left(\frac{\Gamma_c}{NA}\right) = \mu C_0 - \frac{\varepsilon \ell \lambda_u}{h} - \frac{1}{T_0} - \frac{\lambda_u \varepsilon^2 \ell^2}{2\mu C_0 T_0 h} \qquad (4.40)$$

Let us examine the behavior of Γ_c/NA with respect to N and m by plotting its bounds normalized by the maximum throughput (per channel Γ/NA), Eq. (4.38); i.e., we will plot

$$\frac{LB(\Gamma_c/NA)}{\mu C_0 - 1/T_0} \quad \text{and} \quad \frac{UB(\Gamma_c/NA)}{\mu C_0 - 1/T_0}$$

The values selected for the different variables are:

μC_0 = 6 messg/sec

T_0 = .5 sec

λ_u^0 = .07 μC_0

ε = 1/64

Recall that E and ℓ are given in Eq. (3.52).

Three sets of curves are shown in Figs. 4.6 - 4.8 and correspond respectively to the three specifications of λ_u, i.e., $\lambda_u = \lambda_u^0$, $\lambda_u = N^{1/4}\lambda_u^0/2$, $\lambda_u = N^{1/2}\lambda_u^0/2$. A few remarks emerge from the observations of those figures.

Remark 1: Optimal degree of clustering

Lower and upper bound envelopes are also plotted (by hand) on each of those figures. As a result, hierarchical routings with the appropriate degree of clustering m, will achieve a throughput (normal ized) in the region comprised between the lower and upper envelopes.

The optimal m corresponding to a particular envelope can be determined numerically as well as the envelope itself. Such an operation can as easily be done by hand for the graphs presented here. Given N, choosing m on the lower bound envelope (Fig. 4.6) guarantees a minimum throughput equal to the corresponding point on the envelope. Fig. 4.9 shows the relative table length (obtained with those values of m) respective to Fig. 4.6, and also the discretized number of levels (as suggested in Section 3.3.3.2).

Remark 2: Feasibility and viability

The fact that the lower bound and upper bound, for a specific set of parameters, become closer for small values of Γ_c and meet at $\Gamma_c = 0$ indicates, first, up to what size N a certain degree of clustering is feasible, and second, that the points $\Gamma_c = 0$ occur when the line capacities are totally utilized by the updates.

Table 4.1 below, shows approximate values of the points where

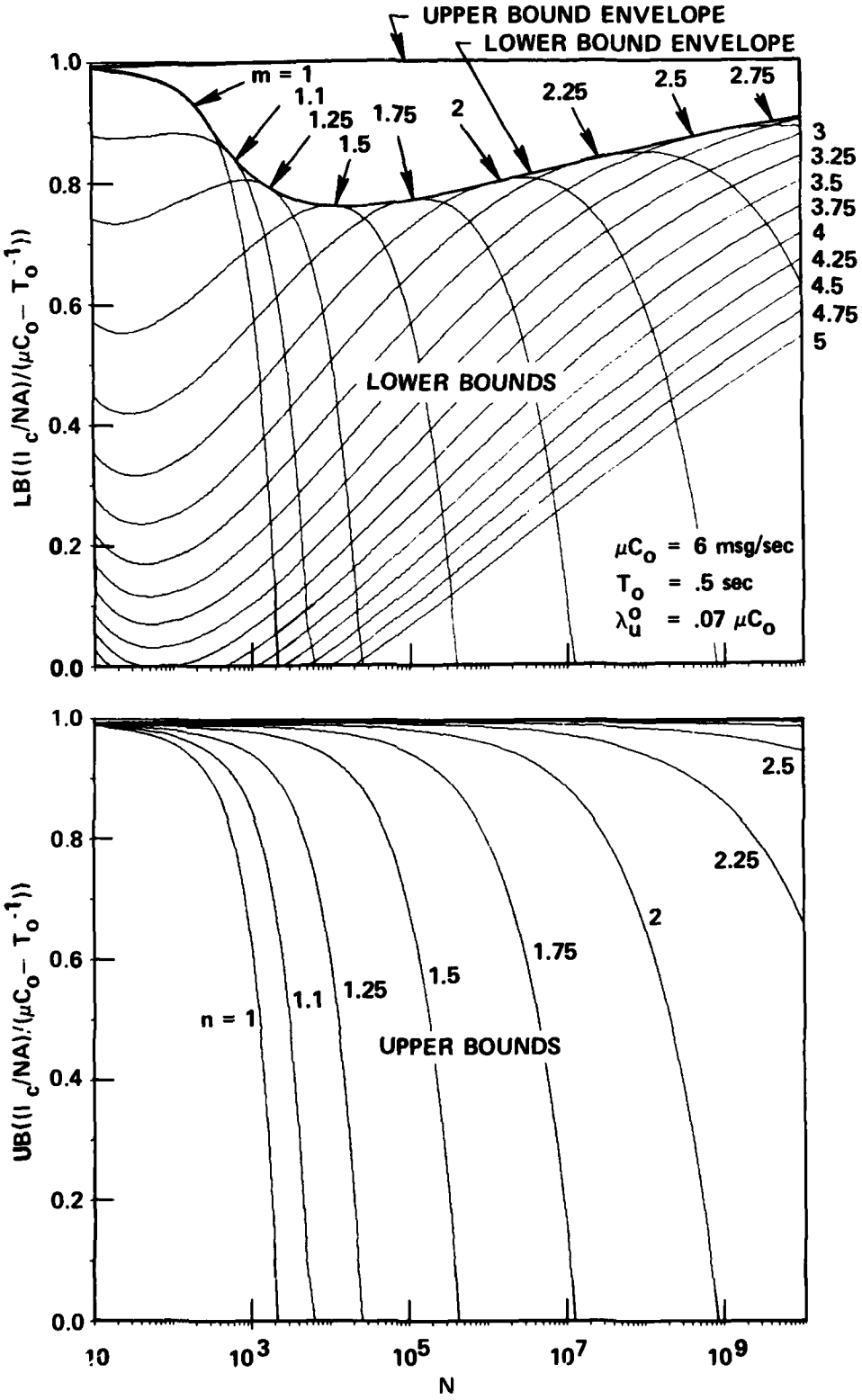


Figure 4.6. Throughput at Constant Delay; Model with Updates, $\lambda_u = \lambda_u^o$.

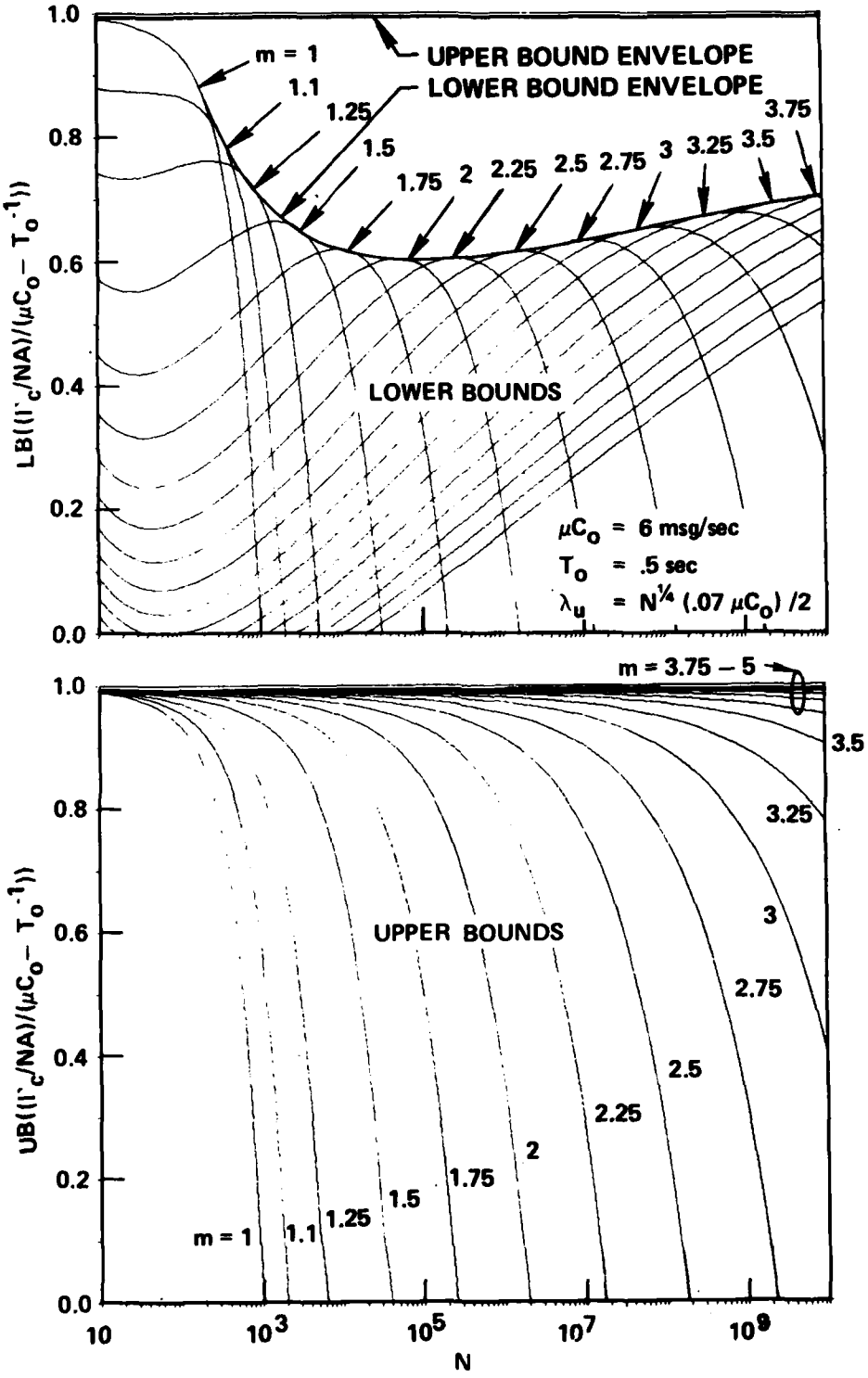


Figure 4.7. Throughput at Constant Delay; Model with Updates, $\lambda_u = N^{1/4}\ \lambda_u^o/2$.

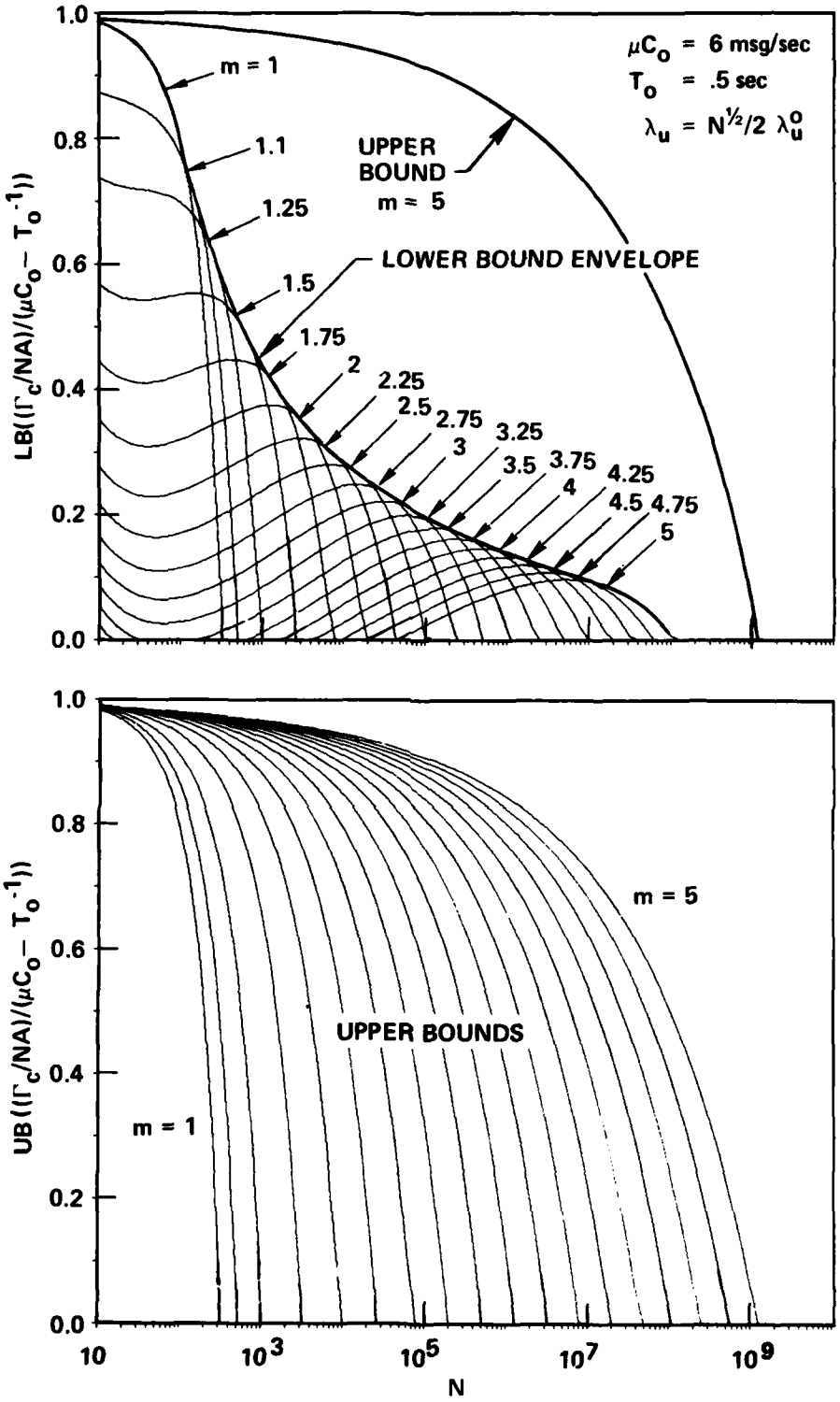


Figure 4.8. Throughput at Constant Delay; Model with Updates, $\lambda_u = N^{1/2}\lambda_u^o/2$.

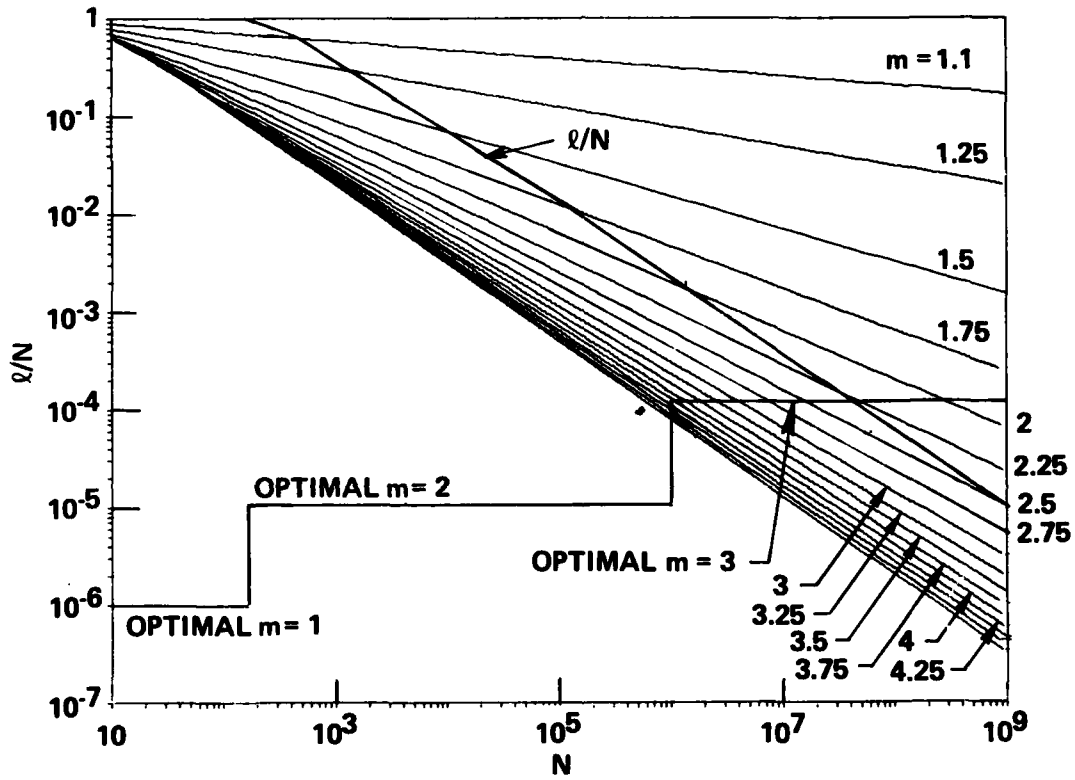


Figure 4.9. Table Length Associated with the Lower Bound Envelope on the Throughput; $\lambda u = \lambda \overset{o}{u}$.

a non-hierarchical routing (m = 1) becomes infeasible and also shows the points beyond which a 2-level hierarchical routing becomes certainly (due to a lower bound) more viable. Notice that the faster the rate of update exchange λ_u is, the smaller those critical values of N are.

	Points at which NCR becomes infeasible ($\Gamma = 0$, $m = 1$)	Points beyond which clustering is better
$\lambda_u = \lambda_u^0$	$N \simeq 2000$	$N \simeq 200$
$= \lambda_u^0 \frac{N^{1/4}}{2}$	$\simeq 1000$	$\simeq 150$
$= \lambda_u^0 \frac{N^{1/2}}{2}$	$\simeq 300$	$\simeq 100$

Table 4.1 Critical Values of N

Remark 3: Asymptotic behavior

The shape of the lower bound envelopes for $\lambda_u = \lambda_u^0$, $N^{1/4}\lambda_u^0/2$, show an initially decreasing and then slowly increasing behavior with respect to N; the increase will eventually bring the curves closer to their asymptote 1. However, for $\lambda_u = N^{1/2}\lambda_u^0/2$, the lower bound envelope is a decreasing function of N which, as predicted earlier, will eventually reach zero. This means that in the neighborhood of a certain size N, the hierarchical routing altogether becomes infeasible. Fortunately, that size is well beyond 10^8!

4.3.3.3 Delay at Constant Throughput per Node (Dual Scaling)

Γ_c/NA is maintained constant, i.e., $\Gamma_c/NA = g_0$, hence

$$T_c = \frac{1 + \frac{\lambda_u}{2\mu C_0} \frac{\varepsilon^2 \ell^2}{h}}{\frac{h}{h_c} \mu C_0 - g_0 - \frac{\varepsilon \ell \lambda_u}{h_c}} \tag{4.41}$$

Again, applying Eq. (4.27), we derive an upper and a lower bound on T_c

$$UB(T_c) = \frac{1 + \frac{\lambda_u}{2\mu C_0} \frac{\varepsilon^2 \ell^2}{h}}{\frac{1}{1 + E} \left[\mu C_0 - \frac{\varepsilon \ell \lambda_u}{h} \right] - g_0} \tag{4.42}$$

$$LB(T_c) = \frac{1 + \frac{\lambda_u}{2\mu C_0} \frac{\varepsilon^2 \ell^2}{h}}{\mu C_0 - \frac{\varepsilon \ell \lambda_u}{h} - g_0} \tag{4.43}$$

Dual properties to the primal scaling hold true here and are illustrated in the curves shown in Figs. 4.10 and 4.11.

Notice also that under the condition of Eq. (4.37) and for a given m, $m \geq m_0$; then

$$\lim_{N \to \infty} T_c = \frac{1}{\mu C_0 - g_0}$$

which is the dual of Eq. 4.38, and is used to normalize the values of T_c in the plots.

4.4 A Queueing Model with No Updates and with Storage Limitation

For large networks, the earlier infinite nodal storage assumption becomes less reasonable. The purpose of this section is to develop and analyze a Kleinrock-like model which takes into consideration the

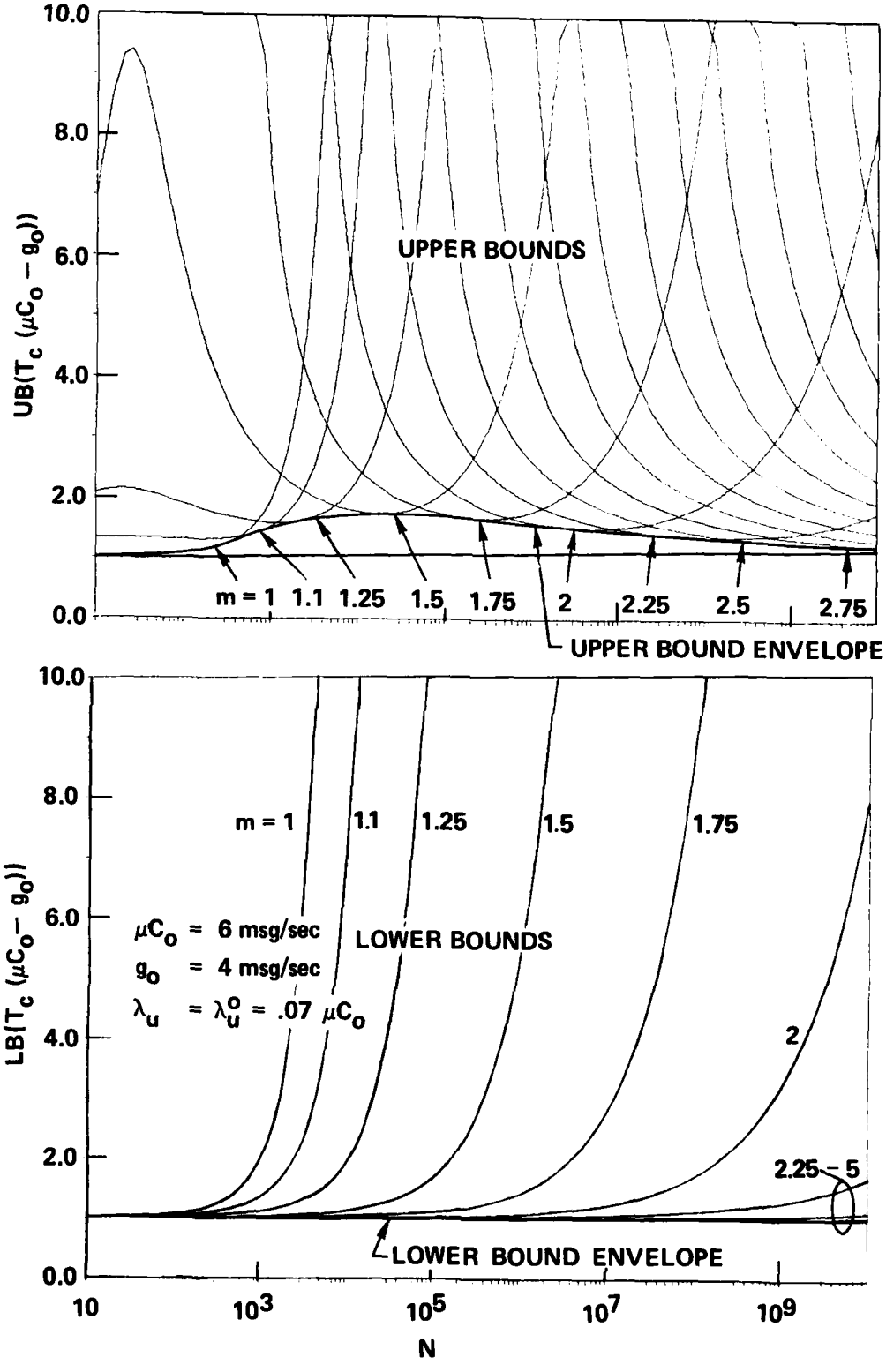


Figure 4.10. Delay at Constant Throughput per Node; Model with Updates, $\lambda_u = \lambda_u^o$.

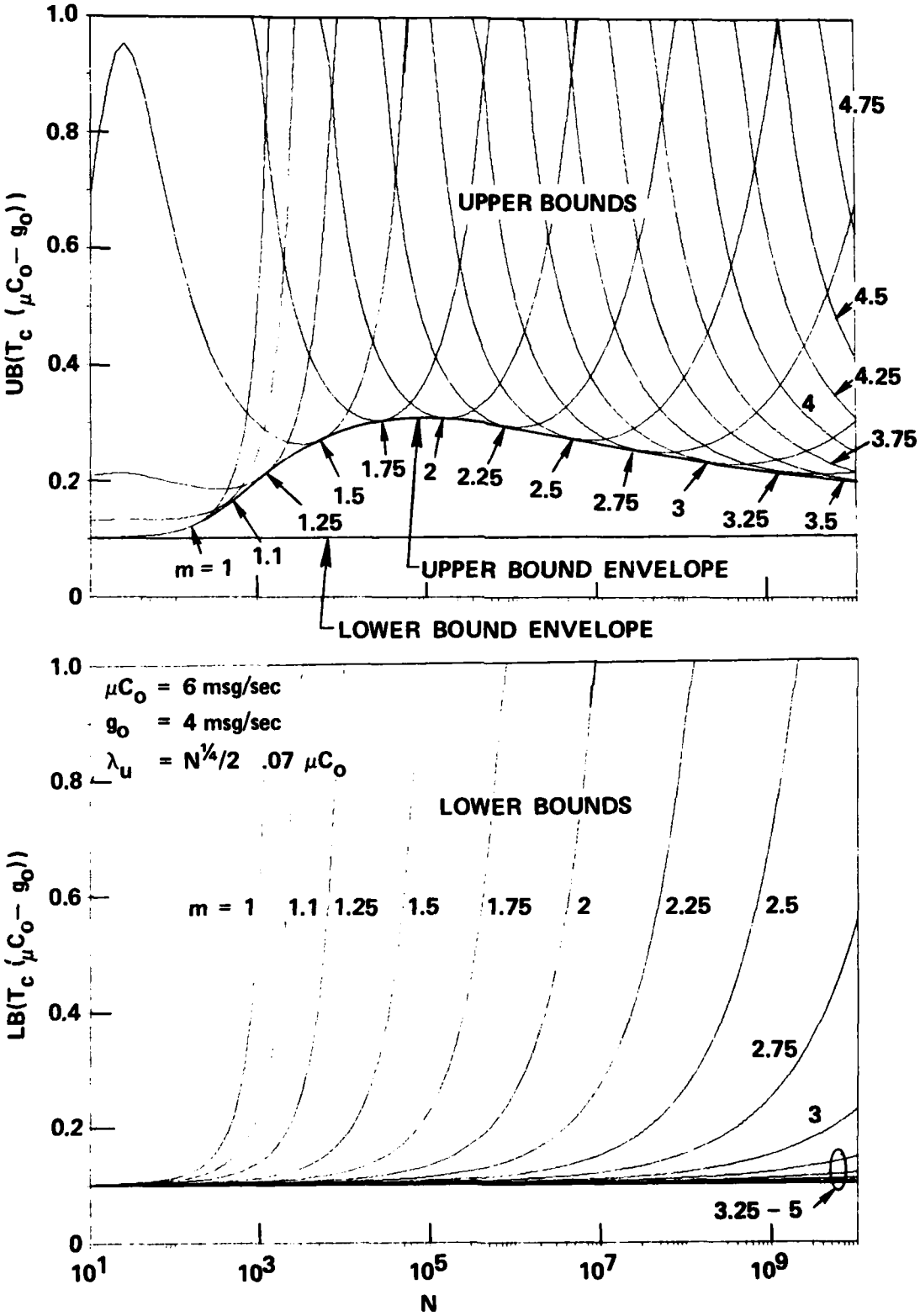


Figure 4.11. Delay at Constant Throughput per Node; Model with Updates, $\lambda u = N^{1/4}/2\ \lambda_u^o$.

limitation of nodal storage. Then based on that model, we study the behavior of hierarchical routing as applied to the class of symmetrical networks. We choose not to account for the effect of updates on the line capacity utilization, in order both to model situations where updates can be neglected and, mainly, to isolate the consequences of finite nodal storage on the network performance.

Similar to the previous section, this study will also demonstrate a remarkable efficiency of hierarchical routing in large networks.

4.4.1 A Loss-Queueing Model for Symmetrical Networks

4.4.1.1 The Model

Again we consider the class of symmetrical networks and, in addition, a constraint is imposed on the number of buffers reserved for the store-and-forward function. As a result of the limited storage, three issues arise: the validity of the exponential message length distribution; the fate of the rejected messages; and the sharing of the pool of S/F buffers among the outgoing channels.

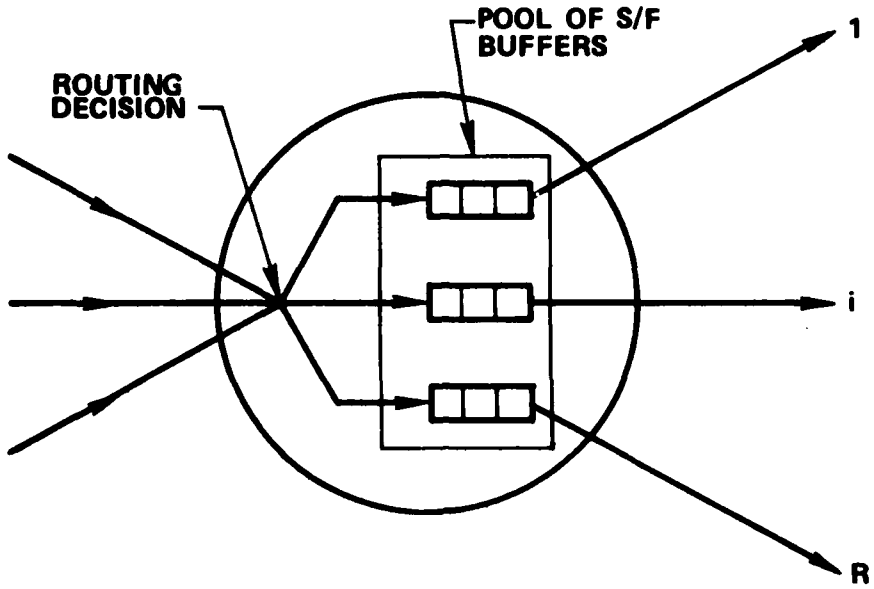
With respect to the message length, it is clear that a maximum size must be imposed, as is always the case in practical situations. As an example, in the ARPANET the maximum packet size is equal to 1008 data bits. The IMP S/F storage is divided into buffers (pages), each of which can accomodate a maximum size packet and cannot be utilized by more than one packet at a time. As a result, one might feel that the assumption of exponentially distributed packets should be replaced with a constant length packet assumption. However, measurements on the

ARPANET [KLEI 74] have shown that the average size of a data message is roughly 250 bits. The fact that the average message length is much smaller than the buffer size, and recognizing that messages[1] which do not fit in a single buffer occur with a very small probability (and hence can be neglected), motivates us to keep the exponential message length assumption. A better approximation would be to assume a truncated exponential message length distribution, but this makes the analysis much more complicated and no closed form solution has been obtained [CHU 69].

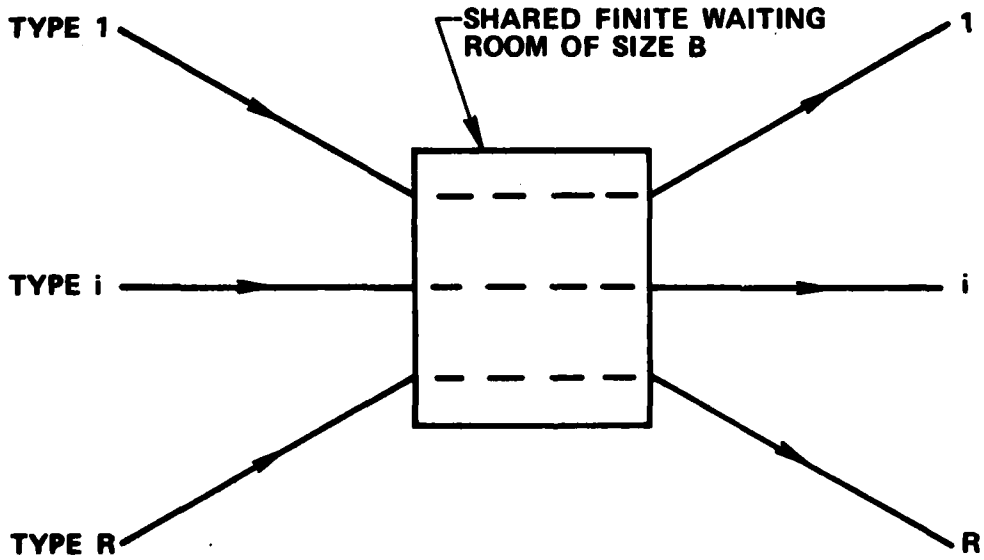
As for the rejected messages, they can be either transmitted by the sending node (after a time-out, as in the ARPANET) or considered as lost (as with blocked telephone calls). The retransmission mode is what actually prevails in general S/F networks like the ARPANET, NPL, etc. However, this mode introduces strong dependencies among the stochastic behavior of neighboring (and even more distant) nodes [ZEIG 71] to the point that an analysis seems out of reach. As a result, we will restrict our considerations to a loss model, in which case the dependencies between nodes due to storage limitation are eliminated.

Finally with respect to the sharing of the pool of S/F buffers between the outgoing channels, the S/F function of a node can be represented as in Fig. 4.12a. There, we see that accepted messages are first submitted to the routing policy and then conceptually join a pool of buffers (B buffers). The routing decision is assumed to be <u>fixed</u> (hence independent of buffer utilization). As a consequence, a node

[1] Recall the single packet messages assumption.



a. S/F FUNCTION



b. R QUEUEING SYSTEMS

Figure 4.12. Model of the S/F Function of a Node.

is equivalent to R single queueing systems which share a pool of B buffers under some scheme. A given channel can only serve messages routed to that channel; hence messages can be considered as forming R classes of customers, and those pertaining to class i are to be uniquely served by the ith channel of that node $i = 1, 2, \ldots, R$ (see Fig. 4.12b). Also similar to the independence assumption [KLEI 64], we will assume that nodes (channels in different nodes) are stochastically independent and that the input streams at each node are governed by independent Poisson distributions.

As a result of the above considerations and assumptions introduced along with them, the network can be considered as a collection of independent nodes, each of which can be modelled as R $M|M|1$ queues sharing a waiting room of size B. The traffic offered at each node is governed by a fixed routing decision, and the probability of blocking at a node is a function of the sharing scheme utilized. Several such schemes are proposed and studied in Appendix B. We list among them: complete sharing (CS); sharing with a constraint on a maximum queue length (SMXQ); sharing with a minimum allocation of buffers per queue (SMA), etc.

4.4.1.2 Analysis

We first consider a few remarks before we proceed with the analysis. Because of the symmetry of our class of networks, we assume that the fixed routing results in equal offered loads on the channels. The offered load λ is defined as the input rate of traffic before acceptance or rejection by a node. Moreover, all nodes are assumed to

contain the same number of buffers B and to use the same buffer sharing strategy. As a result, the probability of blocking (to be denoted by P_B) is the same at all nodes.

Because of the possiblity of loss of messages, the offered external traffic Γ is no longer equal to the throughput of the network, which we denote by Γ_s (s for successful traffic). In what follows, we intend to find Γ_s and the average delay T of the successful traffic.

Throughput versus load

Γ is now referred to as the traffic load. Let us define P_s as the probability that in a steady state, a message transmitted over the network reaches its destination successfully. Clearly

$$P_s = \Gamma_s/\Gamma \tag{4.44}$$

Let $\underset{\sim}{h}$ be the discrete random variable representing the distance in hops between any pair of nodes, as derived from the fixed routing policy. Also, let $P_r[\underset{\sim}{h} = k]$ be the fraction of node-pairs at distance k and H(z) the corresponding z-transform, i.e., $H(z) = \Sigma_k z^k P_r[\underset{\sim}{h} = k]$. (See Appendix A, Section A.4 for more details and for the computation of H(z) for a torus net.)

Because of the uniform traffic assumption,

$$P_s = \sum_{k\geq 1} P_r \text{ [message successful/message travels over k hops] } P_r[\underset{\sim}{h} = k]$$

Since nodes are assumed independent, the probability that a message is not rejected over k hops is $[1-P_B]^k$. Notice that a generated message is subject to rejection, whereas a message reaching its destination is always accepted. Thus

$$P_s = \sum_{k \geq 1} [1 - P_B]^k P_r[\underset{\sim}{h} = k]$$

And from Eq. (A.25),

$$P_s = H(1 - P_B) \tag{4.45}$$

Hence

$$\Gamma_s = H(1 - P_B)\Gamma \tag{4.46}$$

<u>Relation between the load Γ and the total offered internal traffic Λ</u>

Λ is now the sum of the <u>offered</u> input rates to all the network channels. With the above definition of λ, the relation $\Lambda = NA\lambda$ still holds true. However Eq. (4.6) ($\Lambda = \bar{n}\Gamma$) is no longer true due to the possible loss of messages. A similar approach, as used for the derivation of Eq. (4.6) in [KLEI 64], is considered here to derive a relation between Λ and Γ

The contribution of γ_{st}, the rate of traffic from s to t ($\gamma_{st} = \gamma$), to Λ is

$$\sum_{k=0}^{h_{st}-1} \gamma_{st}(1 - P_B)^k = \frac{1 - (1 - P_B)^{h_{st}}}{P_B} \gamma_{st}$$

The proof goes as follows: if h_{st} is the length (in hops) of the unique path from s to t, then the contribution of γ_{st} to the 1st hop (i.e., node s) is γ_{st}; to the 2nd hop, $\gamma_{st}(1 - P_B)$; and in general, to the kth hop it is $\gamma_{st}(1 - P_B)^{k-1}$. The rest of the proof is obvious.

Recall that $\Gamma = N(N - 1)\gamma$ and $\gamma_{st} = \gamma$, then

$$\Lambda = \frac{\Gamma}{P_B}\left[1 - \sum_{s,t} \frac{(1 - P_B)^{h_{st}}}{N(N - 1)}\right]$$

Grouping together all paths of length k and using Eqs. (A.24) and (A.25) we arrive at

$$\Lambda = \frac{\Gamma}{P_B}\left[1 - \sum_{k\geq 1} (1 - P_B)^k P_r[\underset{\sim}{h} = k]\right]$$

hence

$$\Lambda = \frac{1 - H(1 - P_B)}{P_B}\Gamma = \frac{1 - P_s}{P_B}\Gamma \qquad (4.47)$$

The above relation (and Eq. (4.46)) is quite general; it only assumes that all nodes are independent and have an equal probability of blocking P_B. H(z) can be determined analytically or numerically given a particular network and the associated fixed routing policy.

Notice that if $P_B = 0$, i.e., infinite nodal storage assumption, Eq. (4.47) becomes undefined; however, the application of L'Hopital's rule results in $\Lambda = H'(1)\Gamma$. $H'(1)$ is, in fact, equal to the average network path length; hence we are back to the expression derived in [KLEI 64] (i.e., $\Lambda = \bar{n}\Gamma$).

Average delay of successful traffic

Due to the symmetry of our class of nets, a non-rejected message will incur the same delay t at each hop; therefore the average network delay is,

$$T = n_s t$$

where n_s is the average path length of the successful traffic. Note that h is no longer the average path length; this is intuitively due to the fact that messages which travel on longer paths are more likely to be rejected. Therefore, we expect n_s to be, in general, smaller than h. The determination of n_s follows similar to that of P_s.

Let γ_{jk}^{s} be the rate of successful traffic from node j to node k, and h_{jk} the path length between the two nodes, then obviously

$$\gamma_{jk}^{s} = (1 - P_B)^{h_{jk}} \gamma_{jk}$$

Also, the sum of γ_{jk}^{s} is what we have defined as Γ_s. Then, from the definition of the average path length (Eq. (A.3)),

$$n_s = \frac{\sum_j \sum_k \gamma_{jk}^{s} h_{jk}}{\sum_j \sum_k \gamma_{jk}^{s}} = \frac{1}{\Gamma_s} \sum_j \sum_k (1 - P_B)^{h_{jk}} \gamma_{jk} h_{jk}$$

Recall that $\gamma_{jk} = \gamma$, and $\Gamma = N(N - 1)\gamma$; thus

$$n_s = \frac{\Gamma}{\Gamma_s} \sum_j \sum_k (1 - P_B)^{h_{jk}} \frac{h_{jk}}{N(N - 1)}$$

Grouping together all paths of length k, we arrive at

$$n_s = \frac{\Gamma}{\Gamma_s} \sum_{k \geq 1} (1 - P_B)^k k P_r[\underset{\sim}{h} = k]$$

From the definition of H(z) (Eq. (A.25) and Eqs. (4.44),(4.46)), we find

$$n_s = \frac{1 - P_B}{P_s} H'(1 - P_B) = (1 - P_B) \frac{H'(1 - P_B)}{H(1 - P_B)} \qquad (4.48)$$

where H'(z) is the derivative of H(z) with respect to z.

Note that if $P_B = 0$, then $n_s = h$ (recall that $H'(1) = h$ and $H(1) = 1$).

If we let $P_B \to 1$ and apply l'Hopital's rule to the above equation (recall that $H(0) = 0$), we arrive at $n_s = 1$. This result

indicates that at the limit ($P_B \to 1$) only 1-hop traffic may still be successful.

Probability of Blocking

As noticed earlier, a node may be modelled by R M|M|1 queueing systems with a shared finite waiting room of size B. Each server is offered an input rate λ ($\lambda = \Lambda/NA$, Eq. (4.47)), and has a service rate equal to μC. Among the sharing schemes studied in Appendix B, complete sharing is optimal when ρ is small. Below, we show that the maximum throughput Γ_s is obtained for a load, say Γ_m, which is such that $\rho < 1$; moreover in order to maintain a good network performance some control should be introduced in order to keep the offered load at a level less than or equal to Γ_m. As a result it seems reasonable to choose a complete sharing (CS) scheme. However if a retransmission mode is used or if the channels receive very unbalanced traffic loads then a different scheme may be necessary.

The analysis of the CS scheme in Appendix B leads to an expression of P_B in terms of $\lambda/\mu C$, Eq. (B.33), which, combined with Eq. (4.47) results in the system of equations below.

$$\begin{cases} \lambda = \dfrac{1 - H[1 - P_B]}{P_B} \dfrac{\Gamma}{NA} & \text{(a)} \\[2ex] P_B = \dfrac{\binom{B + R - 1}{R - 1}\left(\dfrac{\lambda}{\mu C}\right)^B}{\sum_{K=0}^{B} \binom{K + R - 1}{R - 1}\left(\dfrac{\lambda}{\mu C}\right)^K} & \text{(b)} \end{cases} \qquad (4.49)$$

Also the average delay per channel, t, for successful traffic is given by Eq. (B.37) hence

$$T = n_s \frac{1/\mu C}{1 - \lambda/\mu C} \frac{\sum_{K=0}^{B-1} \binom{K + R - 1}{R - 1} \left(\frac{\lambda}{\mu C}\right)^K - \binom{B + R - 1}{R} \left(\frac{\lambda}{\mu C}\right)^B}{\sum_{K=0}^{B-1} \binom{K + R - 1}{R - 1} \left(\frac{\lambda}{\mu C}\right)^K} \tag{4.50}$$

The above system of equations (4.49a-b), may be solved numerically. A numerical algorithm will be presented below. Once we know λ and P_B, we can determine the performance of the net in terms of throughput Γ_s and delay T. Before we proceed, let us derive the limiting throughput of the network obtained with an infinite traffic load.

Limiting throughput

We claim that

$$\lim_{\Gamma \to \infty} \Gamma_s = \frac{B}{B + R - 1} H'(0) NA\mu C \tag{4.51}$$

Proof:

From the determination of Λ we see that $\lambda > \Gamma/RN$ (i.e., λ is greater than 1/R the traffic generated at the node itself), hence $\Gamma \to \infty \Rightarrow \lambda \to \infty$. Also from Eq. (4.49b) it is clear that

$$P_B = 1 - \frac{B}{B + R - 1} \frac{\mu C}{\lambda} + O(1/\lambda) \tag{4.52}$$

where $O(1/\lambda)$ is such that $\lim_{\lambda \to \infty} \lambda O(1/\lambda) = 0$. Thus, $\lim_{\lambda \to \infty} P_B = 1$.

Now from Eq. (4.49a) we see that $P_B = 1 \Rightarrow \lambda = \Gamma/NA$. As a result,

$$\lim_{\Gamma \to \infty} \lambda \frac{NA}{\Gamma} = 1$$

Also from Eq. (4.52)

$$\lim_{\Gamma \to \infty} \frac{\Gamma}{NA} (1 - P_B) = \frac{B}{B + R - 1} \mu C$$

Now from Eq. (4.46)

$$\Gamma_s = \Gamma H(1 - P_B) = (1 - P_B)\Gamma H'(0) + \Gamma O(1 - P_B)$$

Then, combining the above facts, we arrive at Eq. (4.51).

This limiting result has the following simple interpretation. First, note that $H'(0) = P_r[\underset{\sim}{h} = 1]$ is the fraction of pairs of nodes at distance one (i.e., neighboring nodes), and $(B\mu C/(B + R - 1)$ is the limiting throughput of any channel of the R M|M|1 system of queues (see Eq. (B.39)). As a result, the limiting throughput represents the fraction of successful traffic which has to travel over a single hop. The other fraction (finite) of initially successful traffic has to travel over at least another hop; in trying to do so, it will compete with an infinite amount of traffic generated at the next node, and thus it will be rejected. This checks with the previous result: $P_B \to 1 \Rightarrow n_s \to 1$.

<u>Algorithm for the solution of Eq. (4.49)</u>

Let us study the behavior of the two equations (4.49 a - b)

i. <u>Variation of λ versus P_B</u>, Eq. (4.49a). Differentiating Eq. (4.49a) with respect the P_B, we find

$$\frac{d\lambda}{dP_B} = \frac{\Gamma}{NA} \frac{P_B H'[1 - P_B] - 1 + H[1 - P_B]}{P_B^2}$$

From the definition of H(z), Eq. (A.25)

$$H'(z) = \sum_{k \geq 1} k z^{k-1} P_r[\underset{\sim}{h} = k]$$

Let $z = 1 - P_B$, the numerator of the 2nd fraction becomes

$$Y = (1 - z)H'(z) - 1 + H(z)$$

$$Y = \sum_{k \geq 1} [z^k - 1 + (1 - z)kz^{k-1}] \, P_r[\underset{\sim}{h} = k]$$

Let X_k be the quantity in brackets

$$\frac{dX_k}{dz} = k(k - 1)z^{k-2}(1 - z)$$

Hence for $0 \leq z \leq 1$, $dX_k/dz \geq 0$, thus X_k is an increasing function of z and finally Y is an increasing function of z. When z varies from 0 to 1, i.e., when P_B varies from 1 to 0, Y varies from $H'(0) - 1$ to 0. Since $H'(0) = P_r[\underset{\sim}{h} = 1]$, then $H'(0) - 1 \leq 0$ and $Y \leq 0 \;\; \forall \;\; z \in [0.1]$. Therefore, λ is a decreasing function of P_B (see Fig. 4.13) and from previous remarks: $P_B = 1 \Rightarrow \lambda = \Gamma/NA$; $P_B = 0 \Rightarrow \lambda = h\Gamma/NA$.

ii. <u>Variation of P_B with respect to λ</u>, Eq. (4.49b)

It can be easily shown that the derivative of P_B with respect to λ is always positive. Thus P_B is an increasing function of λ. Also

$$\lambda = 0 \Rightarrow P_B = 0 \quad , \quad dP_B/d\lambda = 0$$

$$\lambda \to \infty \Rightarrow P_B \to 1 \quad , \quad dP_B/d\lambda \to 0$$

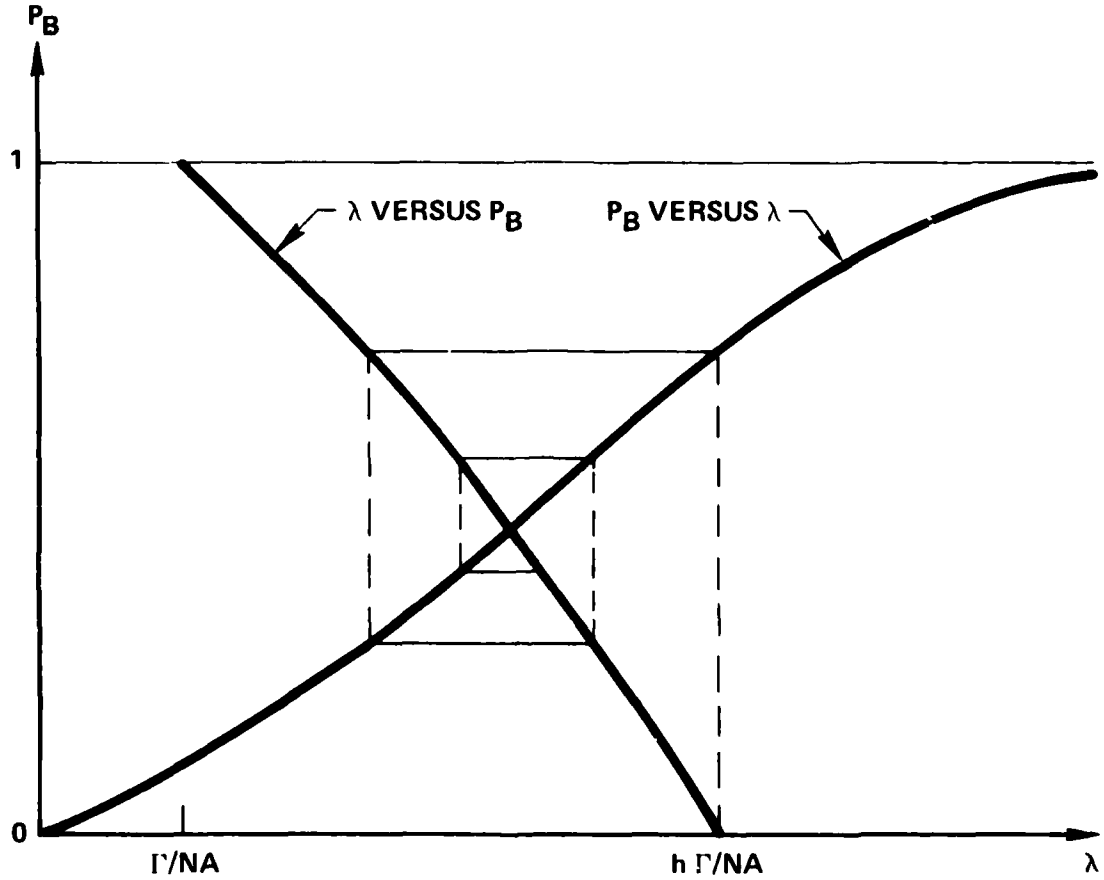


Figure 4.13. Solution of Eq. (4.49).

iii. Algorithm.

Fig. (4.13) shows that there is a unique solution to Eq. (4.49). The numerical evaluation is straightforward, and can use any converging iteration procedure. Namely, let us consider the sequences

$\{\lambda_k\}_k$ and $\{P_B\}_k$ such that,

$$\begin{array}{lcl} \lambda_1 = h\frac{\Gamma}{NA} & \Rightarrow & P_B^1 \\ \lambda_2 = f(P_B^1) & \Rightarrow & P_B^2 \\ \vdots & & \vdots \\ \lambda_k = f(P_B^{k-1}) & \Rightarrow & P_B^k \\ \vdots & & \vdots \end{array}$$

where the λ_k's are computed from Eq. (4.49a) and the P_B^k's from Eq. (4.49b). Obviously, $\lambda_\infty = \lim_{k \to \infty} \lambda_k$ is the solution of the system of equations. In practice, we stop the search when $|\lambda_k - \lambda_{k-1}| \leq \delta$; where δ is a certain tolerance.

We are now ready to consider a numerical example.

4.4.1.3 Application of the Loss Model to Torus Networks

Let us consider a torus (i.e., R = 4) operated with a fixed shortest path routing whose z-transform is given by Eq. (A.41) (where $P = \sqrt{N}$). Of interest is the study of the behavior of the successful traffic Γ_s with respect to the load Γ, as well as the behavior of the delay T with respect to Γ or Γ_s.

Numerical results are shown in Figs. 4.14 - 4.17. Those results were obtained for N = 121, $\delta = 10^{-4}$, and μC = 20 msg/sec.

More precisely, the graphs show the normalized traffic and delay. The normalization is based on Eq. (4.19) which defines the utilization of the net, and on Eq. (4.18). Thus the curves show the variables

$$\rho = \frac{h}{\mu C}\frac{\Gamma}{NA}, \quad \rho_s \triangleq \frac{h}{\mu C}\frac{\Gamma_s}{NA}, \quad \frac{\mu CT}{h}$$

Figs. 4.14 and 4.15 show (linear and semi-log abscissa) that as Γ increases, Γ_s increases to a maximum value and then decreases to its limiting value in Eq. (4.51). These results are similar to that of a contention system [TOBA 74], [LAM 74], except that the non-retransmission (loss) of rejected messages eliminates the possibility of unstable states.

Note that if $B = \infty$ then Γ_s is equal to Γ for ρ varying from 0 to 1. For $\rho \geq 1$ a steady state solution does not exist. This is no longer true for a finite buffer size. However, with limited storage, as ρ increases beyond 1, the throughput decreases quite a bit and thus more messages are being lost. Another effect of finite storage is reflected in the behavior of the average delay T which asymptotically reaches a constant value as ρ goes to infinity (see Fig. 4.16). Since for $\rho \to \infty$ (i.e., $\Gamma \to \infty$) only 1-hop traffic may be successful (see Eq. (4.51)), then the asymptotic value of T corresponds to the delay on one hop (i.e., at one node) under the condition of an infinite input rate. From Eq. (B.39), that value for one node is: $\lim_{\rho \to \infty} \mu Ct = B/(B + R - 1)$; thus $\lim_{\rho \to \infty} \mu CT/h = B/[(B + R - 1)h]$.

Fig. 4.17 illustrates the behavior of T versus Γ_s. Note that to a given Γ_s correspond 2 points on the curve. This is due to the fact that there are two possible values of Γ which lead to a specific value of Γ_s (see Fig. 4.14).

With regard to the number of buffers B, the plots show initial substantial improvements in maximum throughput as B increases, and

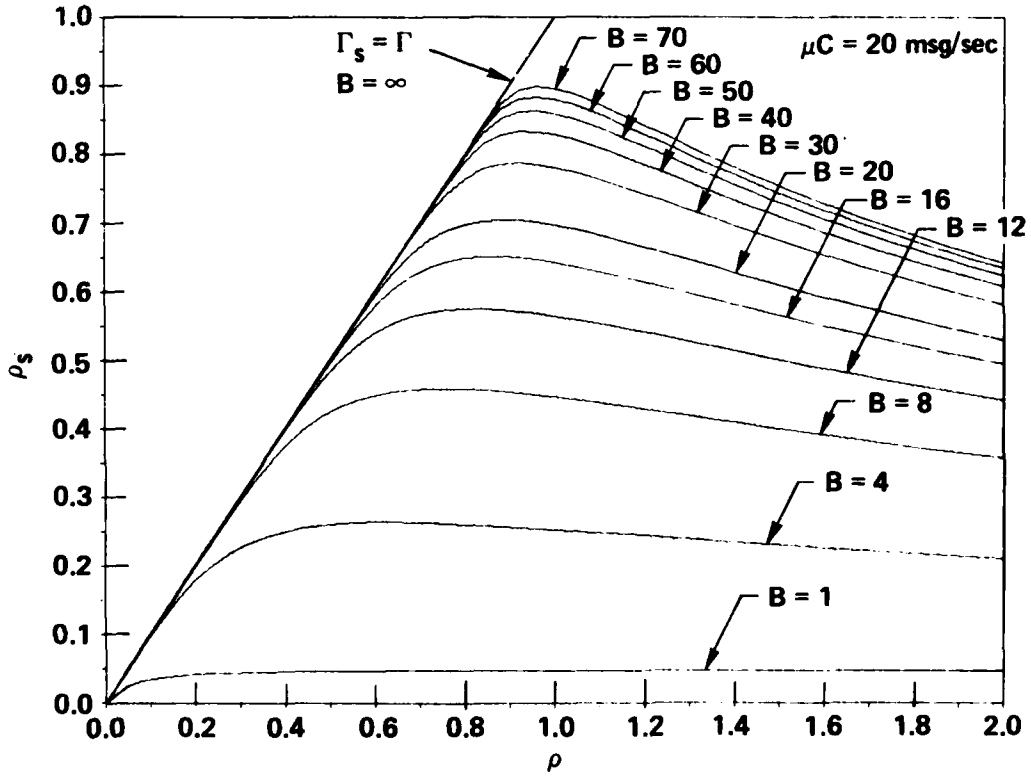


Figure 4.14. Normalized Throughput ρ_s, versus Normalized Load ρ, for a 121-Node Torus with Storage Limitation.

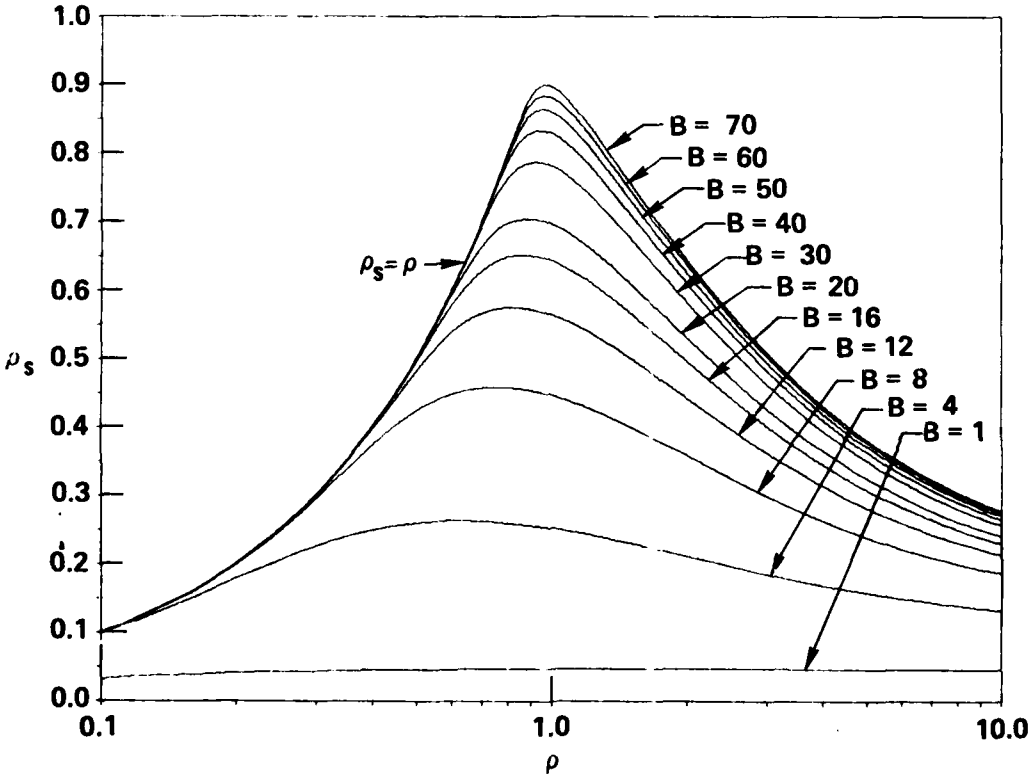


Figure 4.15. ρ_s **versus** ρ**; semi-log Representation.**

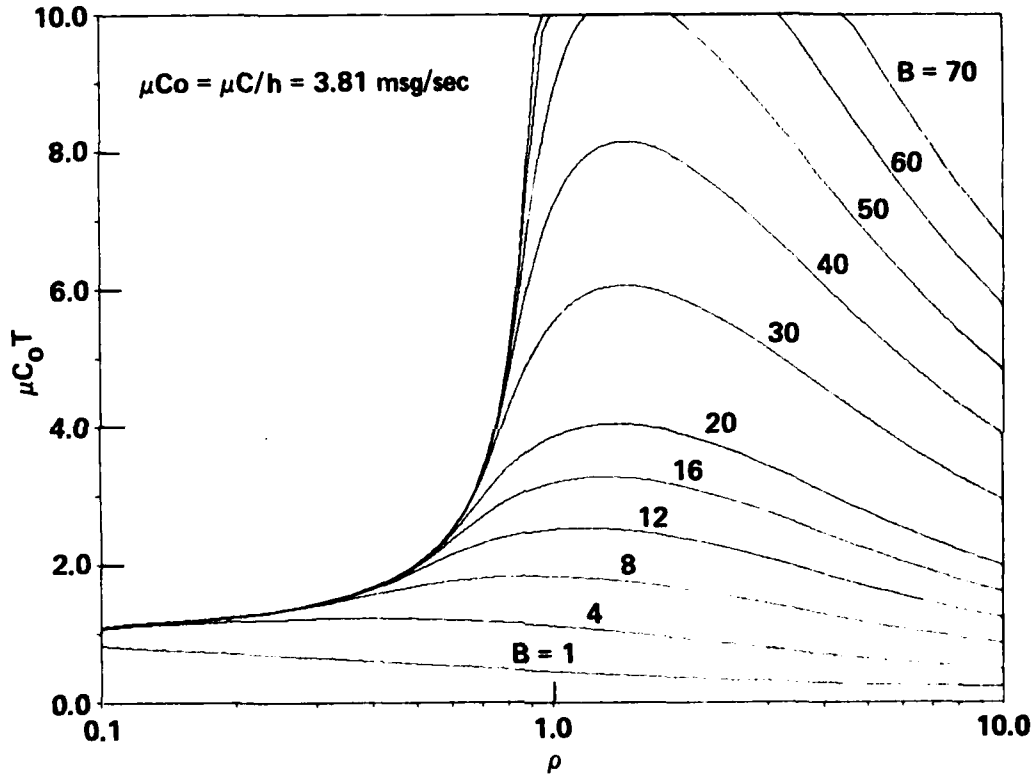


Figure 4.16. Normalized Delay $\mu C_o T$ versus Load ρ, for a 121 Node Torus with Storage Limitation.

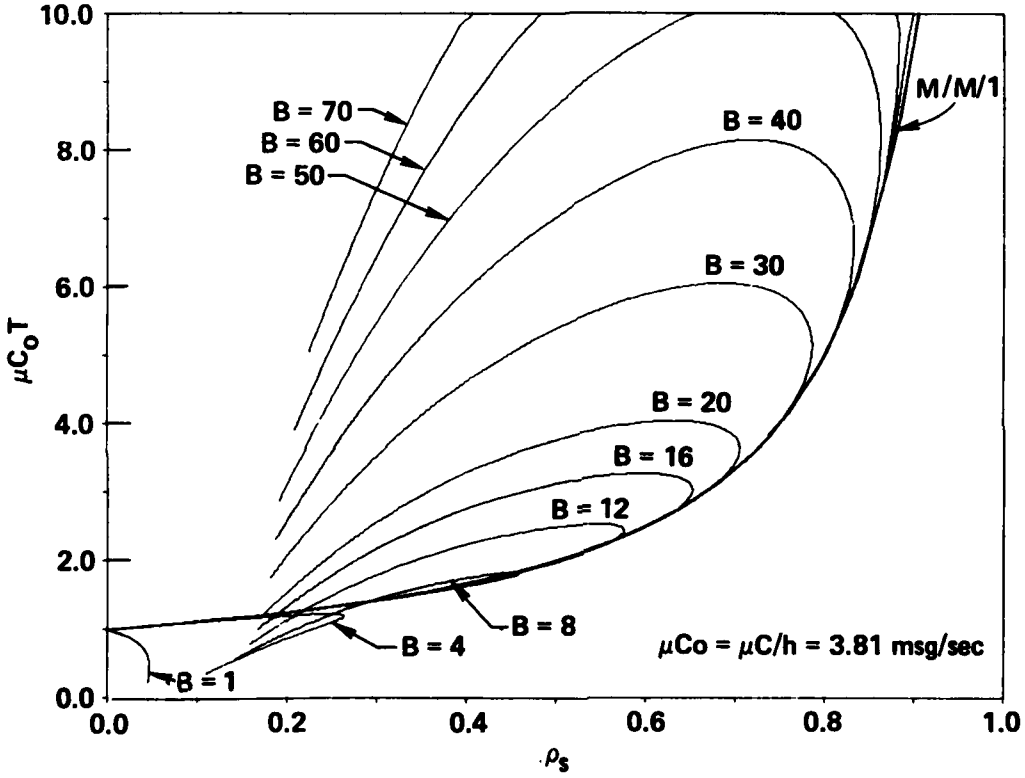


Figure 4.17. Normalized Delay $\mu C_o T$, versus Throughput ρ_s, for a 121-Node Torus with Storage Limitation.

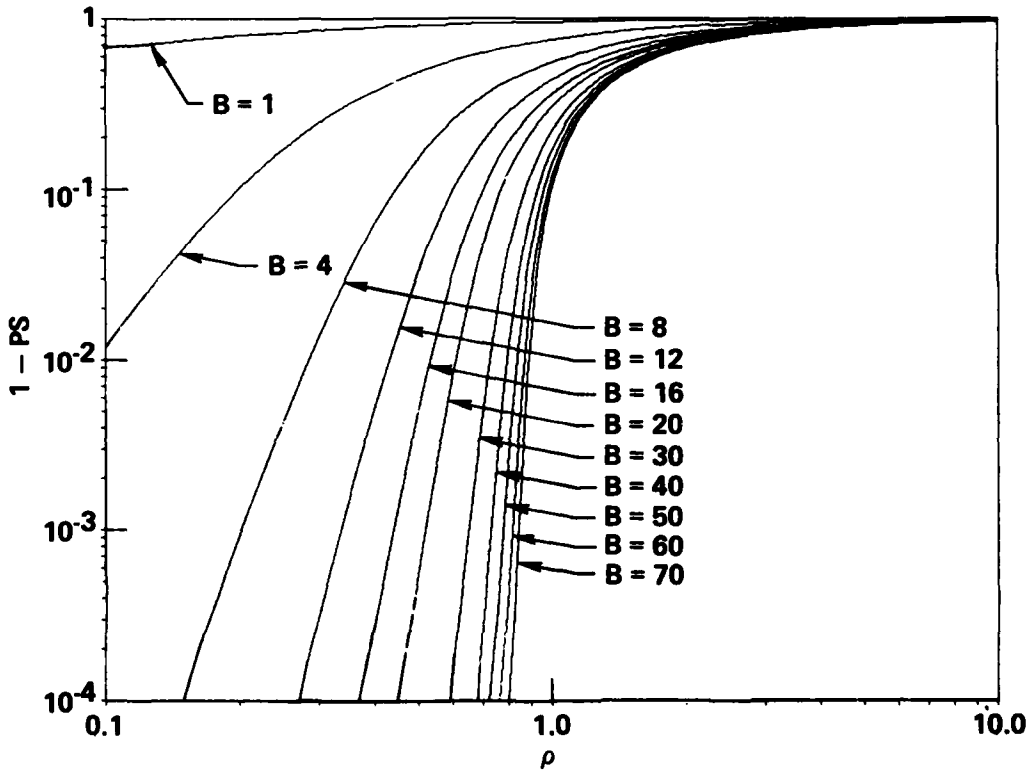


Figure 4.18. Probability of Loss for a 121-Node Torus with Storage Limitation.

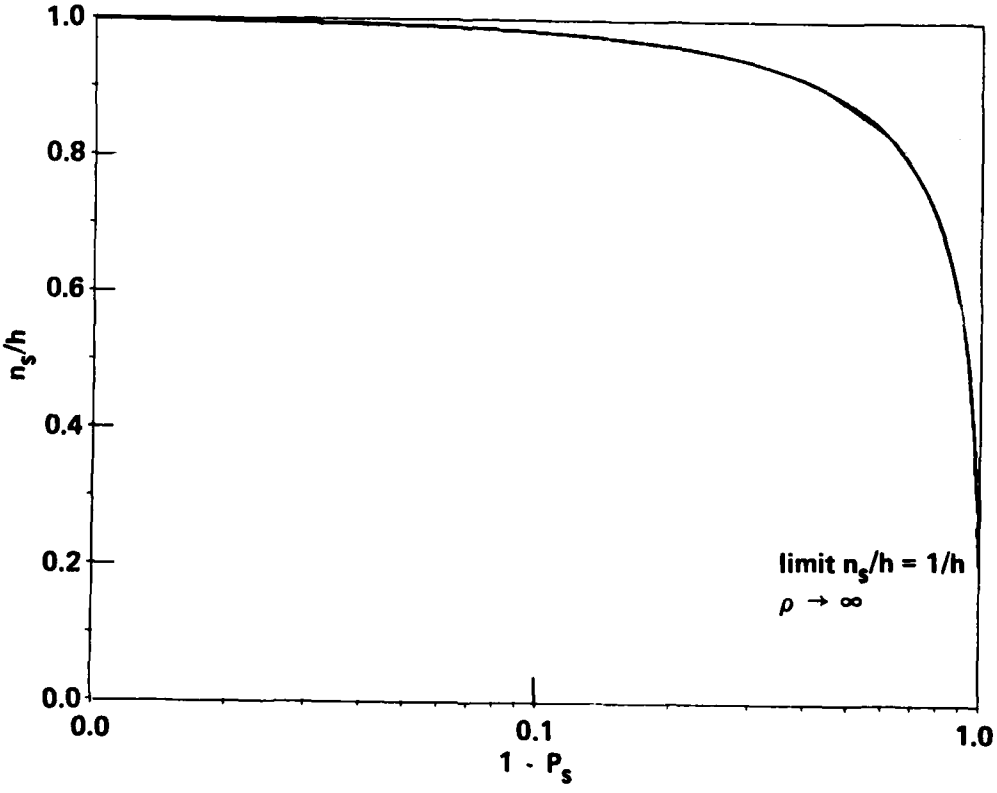


Figure 4.19. Relative Path Length of Successful Traffic, n_s/h; for the 121-Node Torus.

costlier ones when we reach values of ρ_s approximately between 0.8 and 0.9. This phenomenon is better shown in Fig. 4.18 where the probability of loss, $1 - P_s$, is plotted versus the normalized load for different values of B. Finally, Fig. 4.19 shows the behavior of n_s normalized by h, with respect to $1 - P_s$.

In summary, we are now able to evaluate the importance of buffer storage, and as expected, small values of B can degrade the network performance quite a bit. For the example above, B = 20 reduces the performance to roughly 0.68 that which could otherwise be obtained with an infinite number of buffers.

4.4.2 Performance Evaluation of Hierarchical Routing

The above model and Assumption 4.1 will now allow us to study the behavior of hierarchical routing for the class of symmetrical nets. The scaling schemes need further specification as to the scaling of the nodal storage (B).

4.4.2.1 A Buffer Scaling Scheme

As seen in the conclusions above, the effect of B is naturally characterized by the maximum throughput Γ_s, or using our normalized notations, it is characterized by the maximum of $\rho_s = h\Gamma_s/\mu CNA$. Under the conditions of Section 4.1.2, the primal scaling scheme maintained ρ constant as N varied. It is now natural to attempt to keep maximum ρ_s constant. With this objective in mind, let us observe the effect of storage limitation first, in a single node situation, then in a network environment. From Eq. (4.49b) we see that P_B depends only on $\lambda/\mu C$ and B. Thus a scaling which maintains $\lambda/\mu C$ and B constant will result in a constant P_B. Recall that under the conditions of Section 4.1.2,

$\lambda = h\Gamma/NA$; hence $\lambda/\mu C$ is just our previous ρ that the scaling maintained constant.

In a network environment keeping P_B constant will still result in a smaller probability of success P_s as the network grows. This comes about because of the increase in network path length (aN^V) which results from a larger N. It is then necessary to increase B with N, in order to maintain ρ_s constant.

An ad-hoc (heuristic) scaling scheme of B has been devised which, as we will see, satisfies our needs to a large extent. Such a scheme is

$$\lceil B = B_0 \ln h \rceil \tag{4.53}$$

This scheme has been tested on the torus nets as shown in Fig. 4.20 where ρ_s is plotted versus ρ for a set of values of N and two values of B_0. We notice that the maxima of ρ_s occur for roughly the same value of ρ. The curves (except for N = 25) are also fairly close to each other up to and somewhat beyond the maxima. Since, in fact, networks must be operated in that range and since we are dealing with large networks, this scaling scheme appears to be fairly satisfactory.

The aforementioned figures showed the overall behavior of ρ_s versus ρ for a set of values of N. If we restrict our ovservations to the maximum throughput ρ_s, we obtain the curves shown in Fig. 4.21. Those curves were obtained using the combined scaling scheme

$$\begin{cases} C = hC_0 \\ B = \lceil B_0 \ln h \rceil \end{cases} \tag{4.54}$$

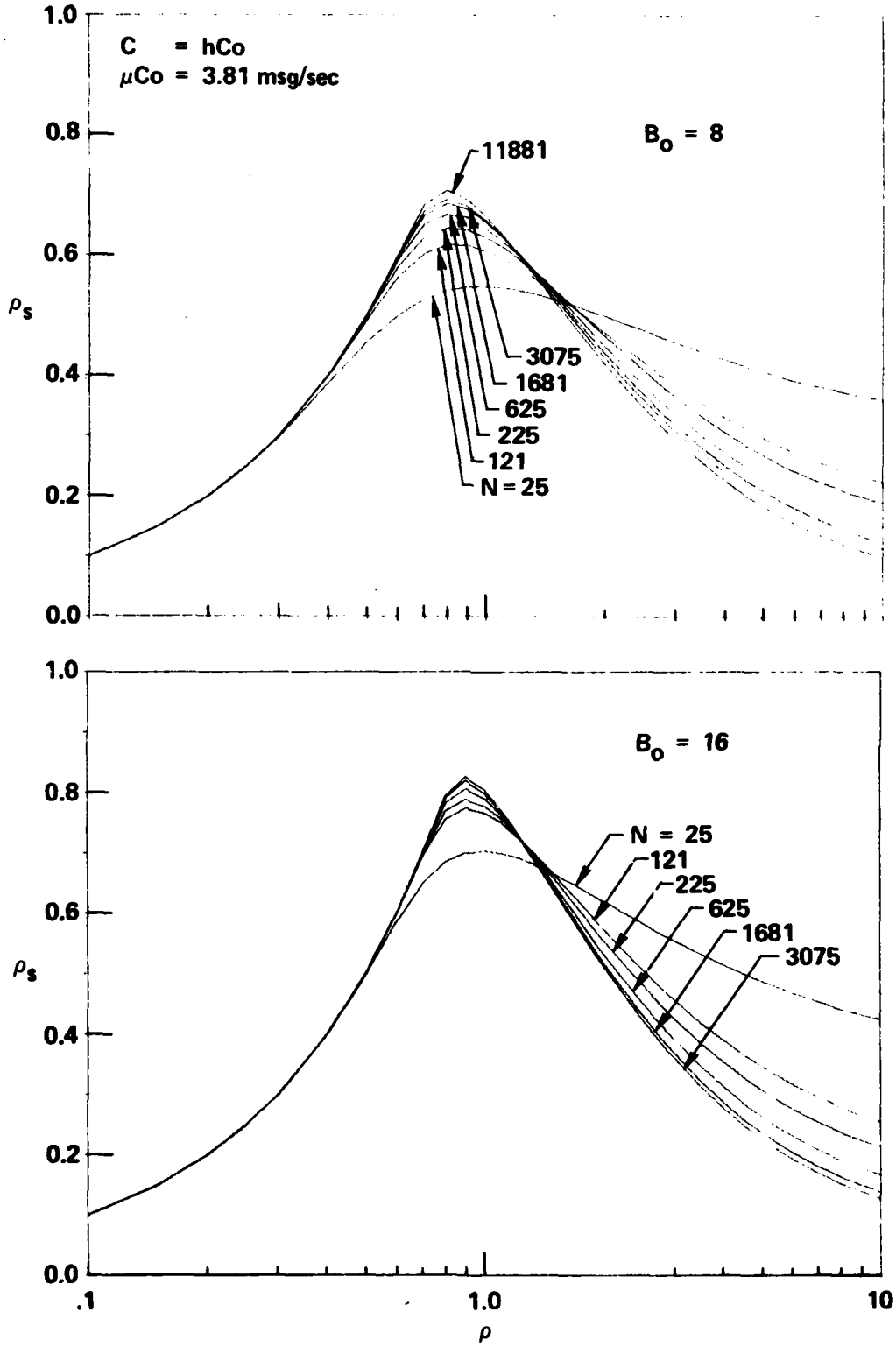


Figure 4.20 Buffer Scaling : Normalized Throughput versus Normalized Load.

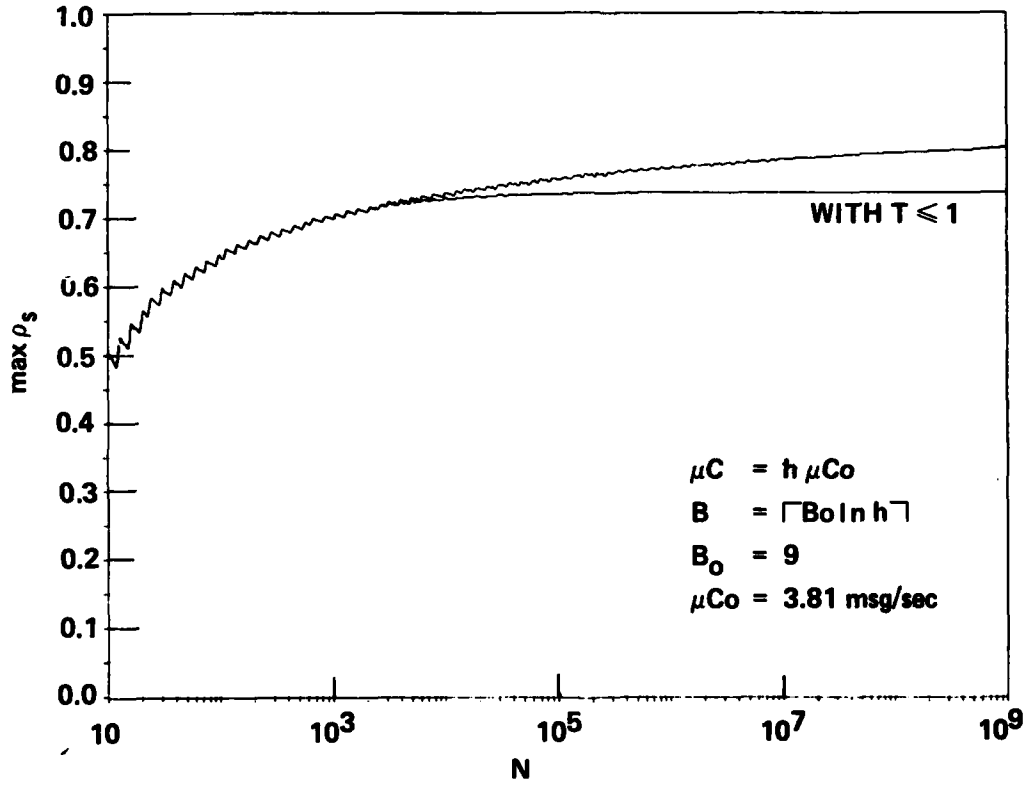


Figure 4.21. Buffer Scaling: Maximum Normalized Throughput versus Network Size.

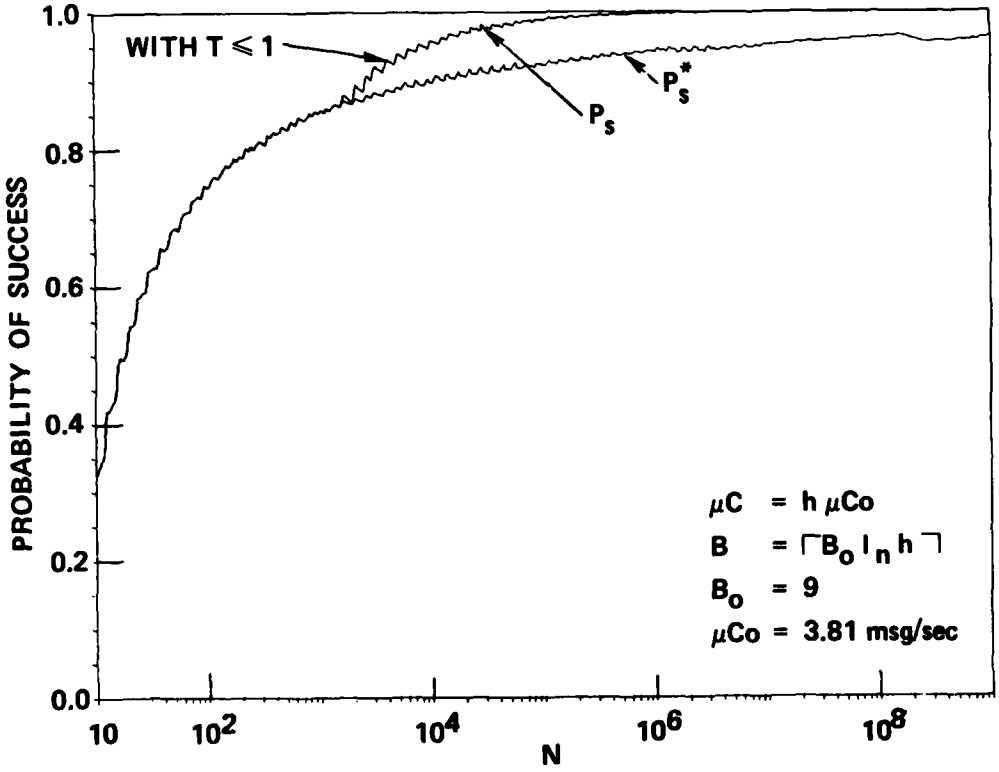


Figure 4.22. Buffer Scaling: Probability of Success.

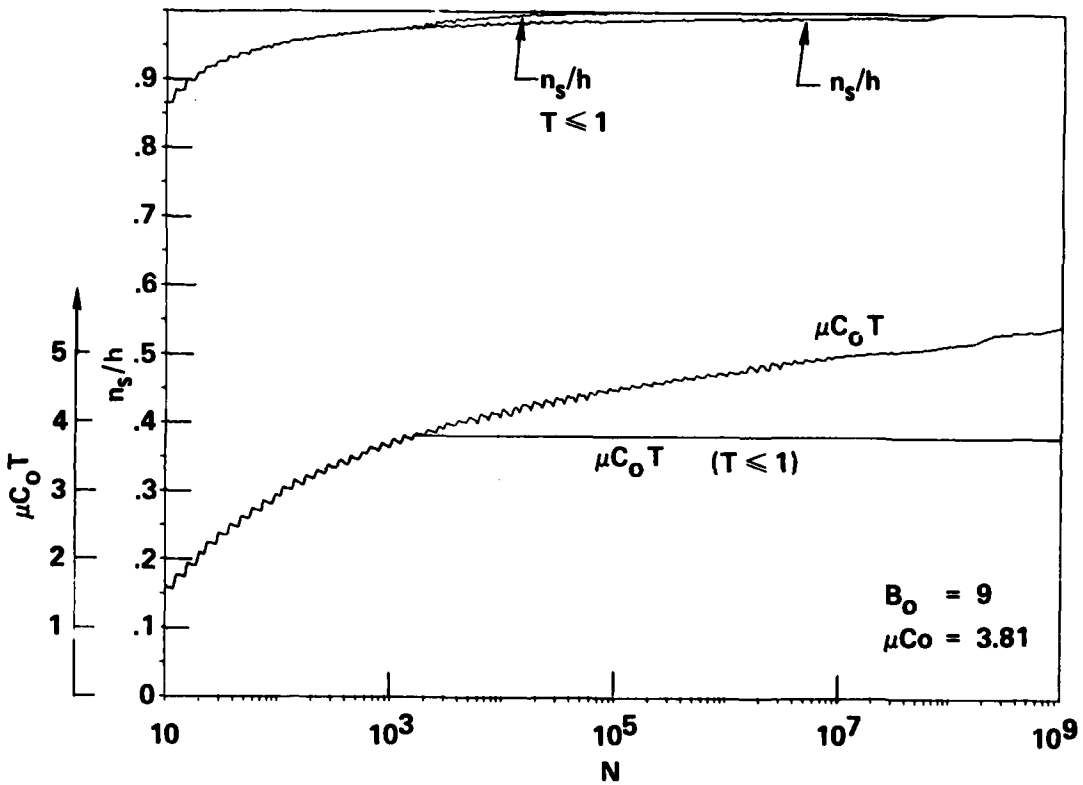
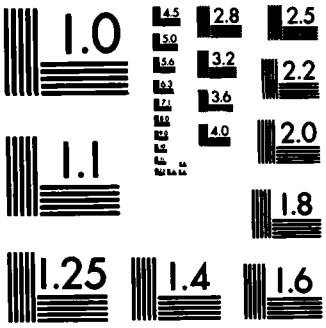


Figure 4.23. Buffer Scaling: Normalized Delay and Average Path Length of Successful Traffic.

AD-A122 995 DESIGN CONSIDERATIONS FOR LARGE COMPUTER COMMUNICATION NETWORKS(U) CALIFORNIA UNIV LOS ANGELES SCHOOL OF ENGINEERING AND APPLIED SCIENCE F KAMOUN APR 76 3/5

UNCLASSIFIED UCLA-ENG-7642 DAHC15-73-C-0368 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

and for a maximum ρ_s determined with or without a constraint on the average delay T. The maximum ρ_s is computed numerically using a Fibonacci search [ZANG 69]. The probability of success P_s the normalized delay $\mu CT/h$, and the normalized average path length n_s/h at those maxima of ρ_s, are shown respectively in Figs. 4.22 and 4.23. A few remarks emerge from the observations of those graphs:

i. The curves of max ρ_s with no constraint on T, and that of the corresponding T, show again that our scaling scheme is quite acceptable, especially when $N > 1000$. For that range of N, the maxima of ρ_s are between 0.7 and 0.8.

ii. The fact that max ρ_s with the constraint $T \leq 1$ becomes constant ($\simeq .74$) for N roughly greater than 10^5 can be explained as follows. Figs. 4.22 and 4.23 show that the corresponding P_s and n_s/h are equal to one for the same range of N. This means that at such a load, B is large enough to give a negligible loss; hence it is safe to use the earlier model based on the infinite nodal storage assumption see Section 4.1.2). That model predicts, with $C = hC_0$ and $T = T_0 = 1$, a value of $\Gamma/NA = \mu C_0 - T_0^{-1}$. Thus we expect max $\rho_s = \frac{h}{\mu C}\frac{\Gamma}{NA} = \frac{1}{\mu C_0}\left(\mu C_0 - \frac{1}{T_0}\right) = .74$, which checks with the value found in the plots.

A final question arises as to the sensitivity of the above results to the scaling of C. For that purpose, let us prove the following <u>proposition</u>.

Consider a symmetrical net of size N, which is such that <u>all</u> channels are of capacity C_1 or C_2. Then network loads proportional to the capacity assigned, result, for a fixed B, in throughputs proportional to the capacities, i.e.,

$$\frac{\Gamma_1}{\mu C_1} = \frac{\Gamma_2}{\mu C_2} \Rightarrow \begin{cases} \dfrac{\Gamma_s^{\,1}}{\mu C_1} = \dfrac{\Gamma_s^{\,2}}{C_2} \\ \\ \mu C_1 T_1 = \mu C_2 T_2 \end{cases} \tag{4.55}$$

where Γ_i, Γ_s^i and T_i are the load, throughput and delay with assignment C_i $(i = 1, 2)$

<u>Proof</u>

Let λ_1 and $P_B^{(1)}$ be the unique solution of Eq. (4.49) for $\Gamma = \Gamma_1$ and $C = C_1$. It is obvious that $\lambda_2 = \lambda_1 \mu C_2/\mu C_1$ and $P_B^{(2)} = P_B^{(1)}$ is the unique solution of Eq. (4.49) for $\Gamma = \Gamma_2$ and $C = C_2$. This fact combined with Eqs. (4.46) and (4.50) proves the above proposition.

As a result, the normalized notation (ρ, ρ_s) is not sensitive to the particular capacity assignment for a given N and B, i.e.,

$$\rho_1 = \rho_2 \Rightarrow \rho_s^1 = \rho_s^2 \quad .$$

The above property implies that the same curves as in Fig. 4.20 can be obtained with any capacity assignment for a given N and B. Moreover, the maxima of ρ_s are also insensitive to the particular choice of capacity (recall that all channel capacities are equal) at a given N and B. This implies that all the points in the curves in Figs. 4.21 - 4.23 depend only on N and the buffer scaling.

With the capacity scaling $C = hC_0$, ρ_s becomes

$$\rho_s = \frac{1}{\mu C_0} \frac{\Gamma_s}{NA} \tag{4.56}$$

which indicates that if the scaling of B keeps ρ_s constant, then the

scaling of C will keep the throughput per node constant, and that fulfills our objective.

4.4.2.2 Behavior of the Hierarchical Routing

Recall that Assumption 4.1 leads us to consider a fixed routing to model an adaptive hierarchical routing. Provided we know the distribution of the path length of the equivalent routing, we can use the loss model and scaling scheme to predict the behavior of such routing schemes as m and N vary. Once again, we use the bounds of Chapter 3 to characterize the message path lengths which result from a hierarchical routing.

Distribution of Path Length

Since the distribution of path length deals with paths on a node-pair basis, we can no longer use the bound E, which was only valid for the average distance. Fortunately Lemma 3.2 (see Eq. (3.29)) provides us with the more general bound on individual paths (to be denoted by Δ)

$$h^c_{st} - h_{st} \leq \Delta = \sum_{k=1}^{m-1} d_k \qquad \forall \ s,t \in S$$

However this bound, always true for CER, is only valid with OBR if s,t belong to a lower level cluster than C_m (see Eq. (3.31)). The fact that first, Δ is a very generous bound, and second that we expect OBR to behave better than CER makes us feel confident to use Δ as an approximation on paths for the OBR scheme. Note that a rigorous but extremely generous bound, may be obtained from Lemma 3.1 ($h^c_{st} \leq \sum_{k=1}^{m} d_k$).

From Eq. (3.46) and for our class of networks, Δ is simply equal to hF (see Eqs. (3.46) and (3.47)); and for an optimal clustering of degree m

$$\Delta = b \frac{N^v - N^{v/m}}{N^{v/m} - 1} + c(m - 1) \qquad (4.57)$$

Note again that $m = 1 \Rightarrow \Delta = 0$.

A "best case" and a "worst case" distribution can now be defined if we assume respectively that $h^c_{st} = h_{st}$ and $h^c_{st} = h_{st} + \Delta$.

To the "best case" distribution corresponds the z-transform $H(z)$ of the shortest paths in the network.

The z-transform $H_c(z)$ of the worst case distribution can be expressed in terms of $H(z)$ and Δ, as follows

$$H_c(z) = z^{\Delta} H(z) \tag{4.58}$$

To prove the above equation, let $\underset{\sim}{h}_c$ be the deterministic random variable representing the path length, then

$$P_r[\underset{\sim}{h}_c = k] = P_r[\underset{\sim}{h} = k - \Delta] \qquad \forall\, k \geq \Delta$$

Thus

$$H_c(z) = \sum_{k \geq \Delta} z^k P_r[h_c = k] = z^{\Delta} \sum_{i \geq 0} z^i P_r[\underset{\sim}{h} = i]$$

which proves Eq. (4.58).

The above remark that $m = 1 \Rightarrow \Delta = 0$ implies also that $m = 1 \Rightarrow H_c(z) = H(z)$. This fortunate property preserves the continuity of hierarchical routings.

Buffer Assignment and Feasibility

Recall that in this study we intend to account for the storage utilized by the routing tables. The size of such a storage is a linear function of the table length and counted in number of buffers it is equal to

$$\lceil \varepsilon_2 \ell \rceil$$

where $1/\varepsilon_2$ is the number of entries which fit in one buffer. As a consequence, if the total number of buffers to be shared between the routing table and the S/F function is as defined in Eq. (4.53), then the number of buffers strictly reserved for the S/F function is

$$B = \lceil B_0 \ln h \rceil - \lceil \varepsilon_2 \ell \rceil \tag{4.59}$$

With an optimal clustering of degree m, and for our class of symmetrical nets ($h = aN^v$), the above equation becomes

$$B = \left\lceil B_0(\ln a + v \ln N) \right\rceil - \left\lceil \varepsilon_2 m N^{1/m} \right\rceil \tag{4.60}$$

For a hierarchical routing to be feasible, B must be greater than or equal to one. Observing Eq. (4.60) we may conclude that:

i. For a fixed m, the routing becomes infeasible for networks of size larger than a critical number N_c. N_c is the solution of $B = 0$ in Eq. (4.60) and it is, obviously, an increasing function of m.

ii. For very large networks, and under the condition $B_0 v \geq \varepsilon_2 e$, only a hierarchical routing operating with a global minimum table length is feasible, i.e., $m = m_* = \ln N$; hence $\ell = mN^{1/m} = e \ln N$.

In summary, as N gets larger, it becomes imperative to move toward more clustering, eventually reaching a globally minimum table length. The decision to use a higher degree of clustering m should be weighed against the degradation incurred by the corresponding increase in network path length. This phenomenon is illustrated in the next section.

Numerical Application

From the above considerations, the application of the loss

model to a network operated with a hierarchical routing results in the evaluation of the worst and best case performances.

The worst case performance is characterized by

$$\Gamma_s = H_c(1 - P_B)\Gamma$$

The probability of blocking P_B is the solution of Eq. (4.49) where H is replaced by H_c. We note that because $H_c(z)$ is similar to H(z) the iterative algorithm devised to solve Eq. (4.49) is still valid, except that the initial value of the sequence $\{\lambda_k\}$ is now

$$\lambda_1 = \lim_{P_B \to 0} \frac{1 - H_c(1 - P_B)}{P_B} \frac{\Gamma}{NA} = H_c'(1) \frac{\Gamma}{NA}$$

with $H_c'(1) = H'(1) + \Delta = h + \Delta$. With respect to the delay T, let n_s^c be the average path length of the successful traffic, replacing H(z) by $H_c(z)$ in Eq. (4.48), we arrive at $n_s^c = n_s + \Delta$; therefore, $T = (n_s + \Delta)t$ instead of Eq. (4.50).

The "best case" performance is obtained by setting $\Delta = 0$.

Again we use the set of values of a, b, c, v as derived for the torus nets, Eq. (3.50), and the z-transform H(z) as given in Eq. (A.31). The scaling is specified by the usual $C = hC_0$ and Eq. (4.59) Also ε_2 is chosen to be equal to 1/64 (this value is motivated by the ARPANET where a buffer holds up to 64 words.)

We first evaluate the behavior of an MHR as applied to a specific network size N = 1681. Fig. 4.24 shows the plots of the lower bound on ρ_s, the normalized thoughput, versus ρ, the normalized load, and this for a set of values of m. In that example, B_0 is chosen equal to 9; with that value the number of S/F buffers at m = 1

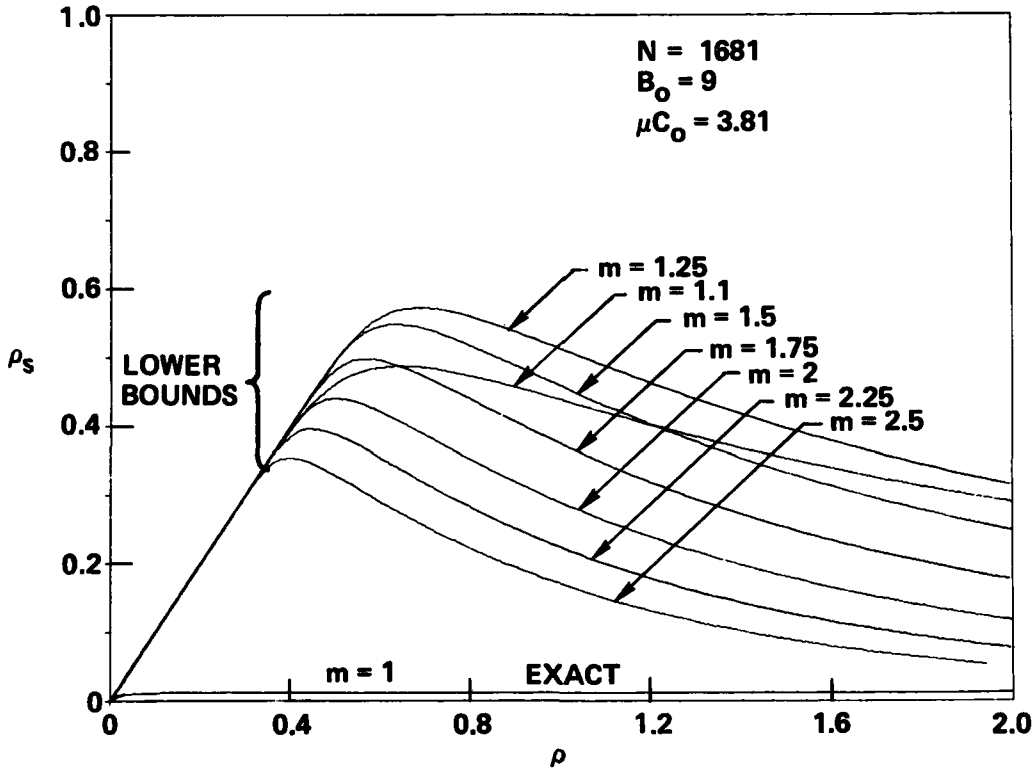


Figure 4.24. Throughput ρ_s, versus Load ρ, with Different Degrees of Clustering.

(no clustering) is equal to one. We observe that the best performance (lower bound) is obtained for m = 1.25 from among the set of m's selected here. With m = 1.25 the maximum ρ_s improves dramatically from $\rho_s = 0.02$ with m = 1, to a minimum of 0.58. We also notice that the performance improves as m varies from 1 to 1.25 and then it deteriorates for m > 1.25. Moreover, the curves with a smaller m exhibit a slower decay. This fact may be explained as follows: from Eq. (4.51), the limiting throughput is

$$\frac{B}{B + R - 1} H_c'(0)NA\mu C$$

but

$$H_c'(z) = \Delta z^{\Delta-1}H(z) + z^{\Delta}H'(z)$$

hence $z = 0 \Rightarrow H_c'(0) = 0$ (for $\Delta > 1$, which is usually the case, see Eq. (4.57)), and also the larger Δ is, the faster is the convergence of $H_c'(0)$ to zero. Furthermore, for a fixed N, Δ is an increasing function of m.

If we limit our considerations to an operational range of ρ (i.e., ρ is only allowed to vary from 0 up to a value slightly larger than the one producing the maximum ρ_s, roughly $0 \leq \rho \leq 1$), then the value of m which leads to the maximum ρ_s also leads to the best performance over that entire range.

As a consequence of the above considerations, from now on we will restrict our observations to the behavior of the maximum ρ_s (lower or upper bound) with respect to N and for a set of values of m. Furthermore the maximum ρ_s will be determined with or without a constraint on the average delay T.

Fig. 4.25 shows such a behavior of max ρ_s, with no constraint on T. Figs. 4.26 - 4.29 show respectively the corresponding average delay T, the probability of success P_s, the number of S/F buffers B and finally the corresponding relative bound on the increase in path length Δ/h.

Figs. 4.30 - 4.32 show similar plots for max ρ_s with a constraint on the maximum delay for T and P_s. A few remarks emerge from the observation of those figures. Those remarks are, in general, quite similar to the ones stated at the end of Section 4.3.2.2, namely with regard to the optimal degree of clustering m for a given N, and the feasibility and viability of hierarchical routing. Before we proceed, let us notice that the non-smoothness of the curves is due to the discrete changes of B (see Eq. (4.59)). This fact is more accentuated for the smaller values of N where B is small, and consequently a change of one unit is relatively noticeable. Round-off errors, as well as errors due to our numerical algorithms for finding P_B and especially max ρ_s (Fibonacci search), may also contribute to the non-smoothness of the curves.

Hierarchical routing with an appropriate degree of clustering, m, guarantees a behavior of the max ρ_s (with respect to N) to lie between the upper and lower bound envelopes. It is quite remarkable that the lower bound envelopes (with or without a constraint on T) remain relatively flat (around 0.6), for N beyond one hundred. Moreover the upper bound envelopes are very close to the curves obtained in Fig. 4.21; this means that at the point (N, m) corresponding to those envelopes, the storage required by the updates is relatively negligible.

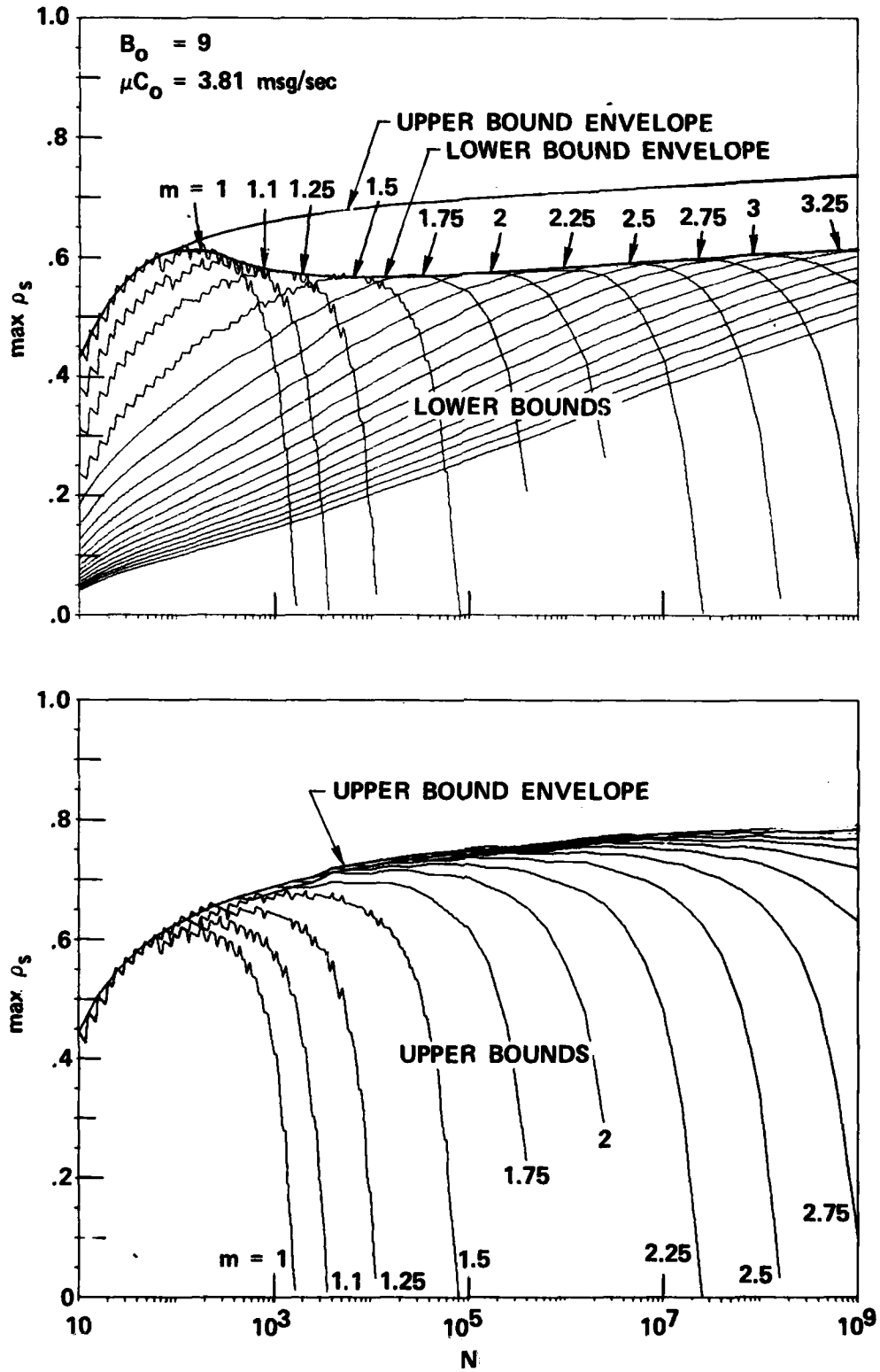


Figure 4.25. Maximum Throughput Obtained with the Hierarchical Routing Model with Updates and With Storage Limitation.

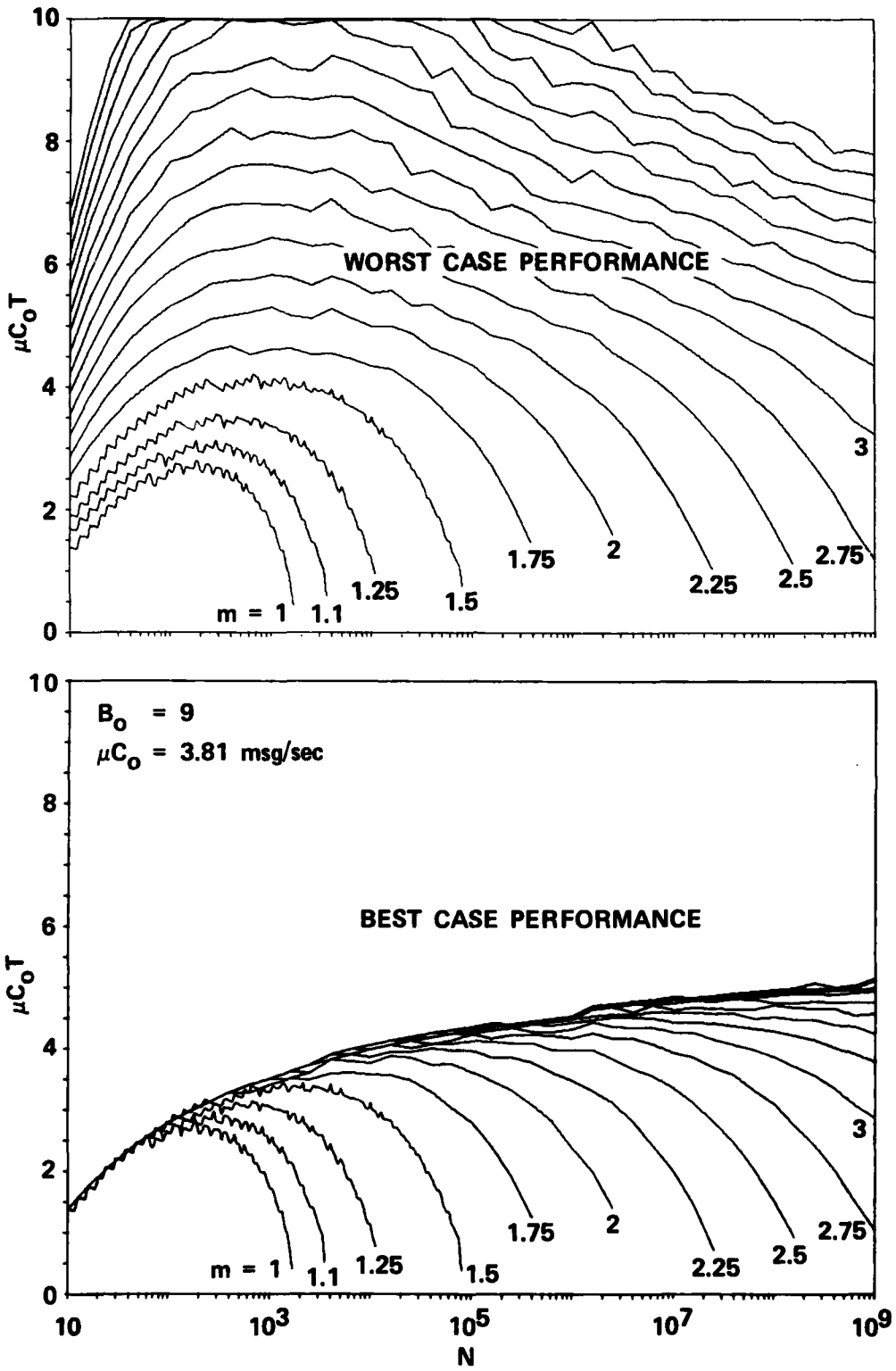


Figure 4.26. Network Delay at Maximum Throughput with the MHR.

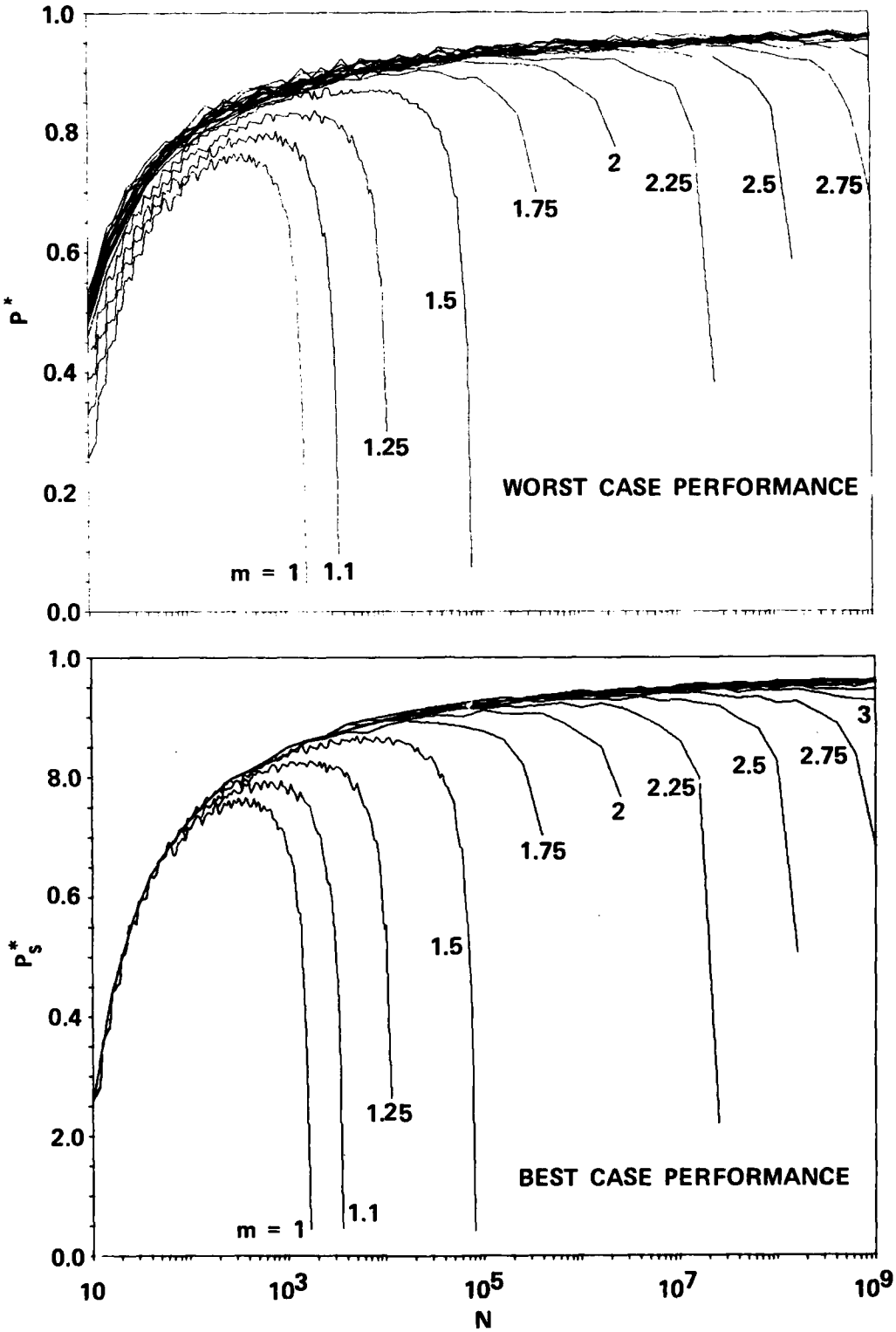


Figure 4.27. Probability of Success at Maximum Throughput.

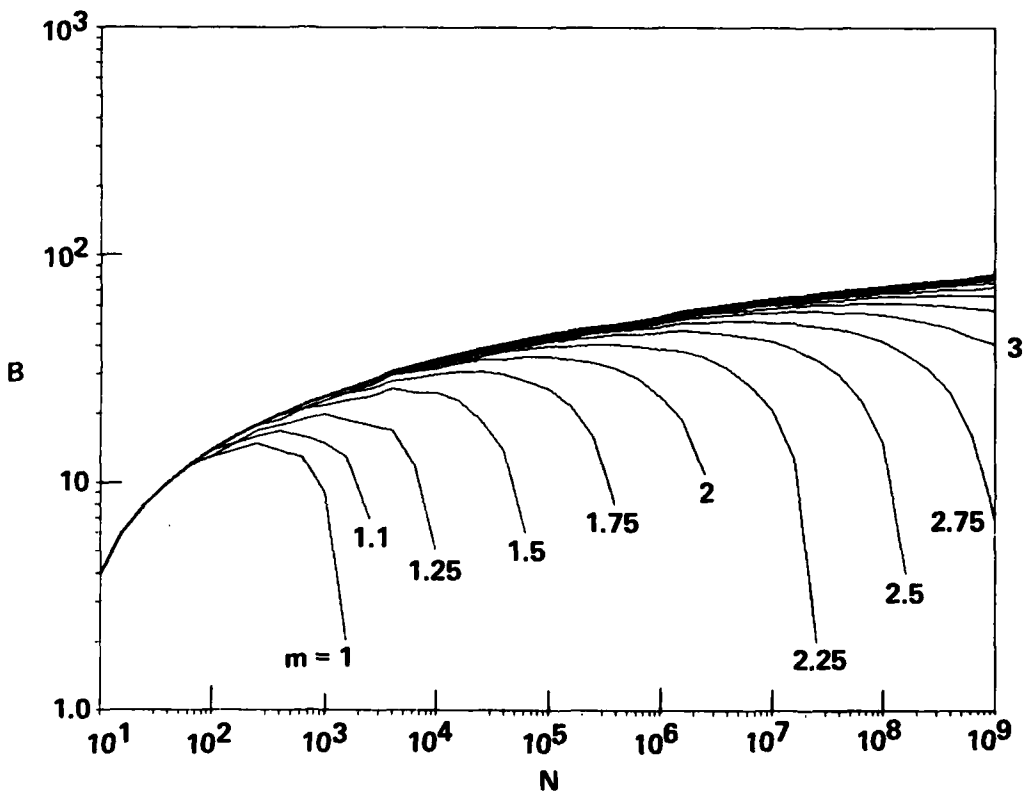


Figure 4.28. Behavior of the Number of S/F Buffers per Node, with the MHR.

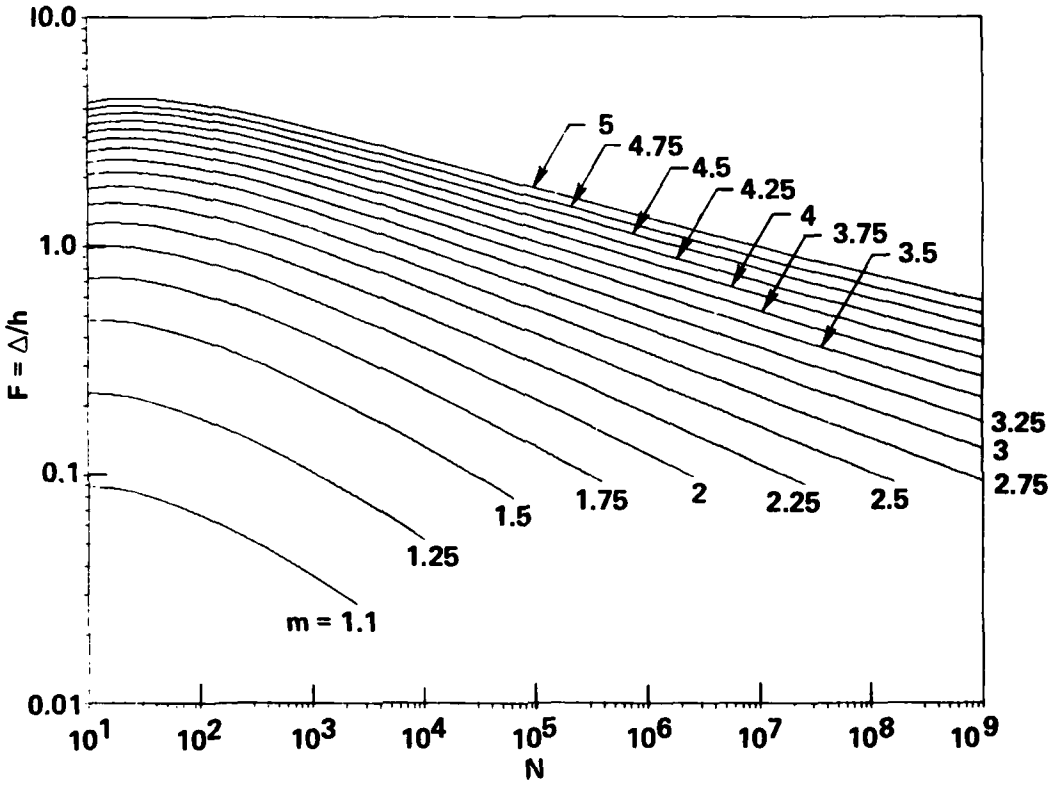


Figure 4.29. Behavior of the Relative Increase in Path Length with the MHR.

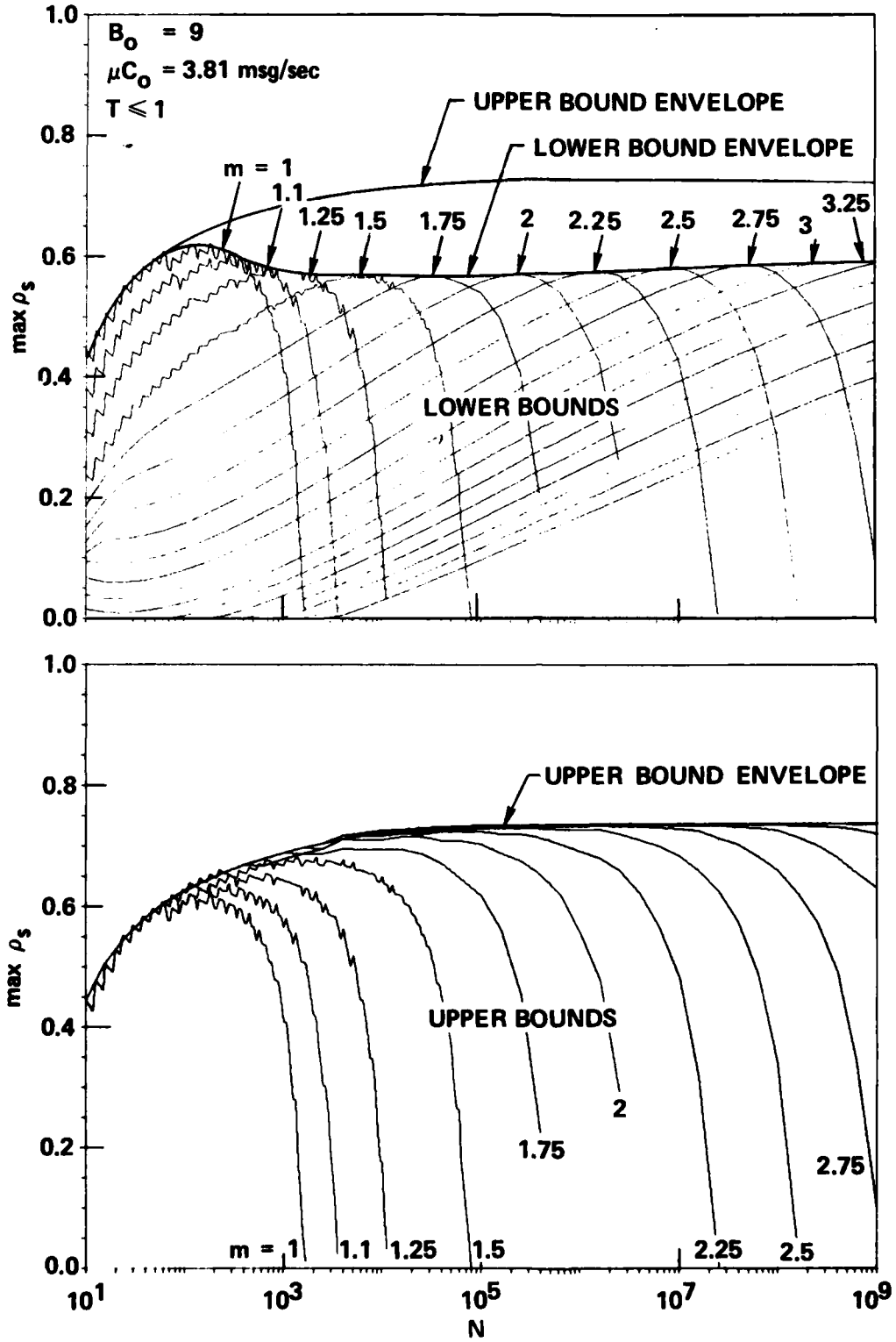


Figure 4.30. Maximum Throughput with a Delay Constraint, Performance of the MHR; Model with No Updates and With Storage Limitation.

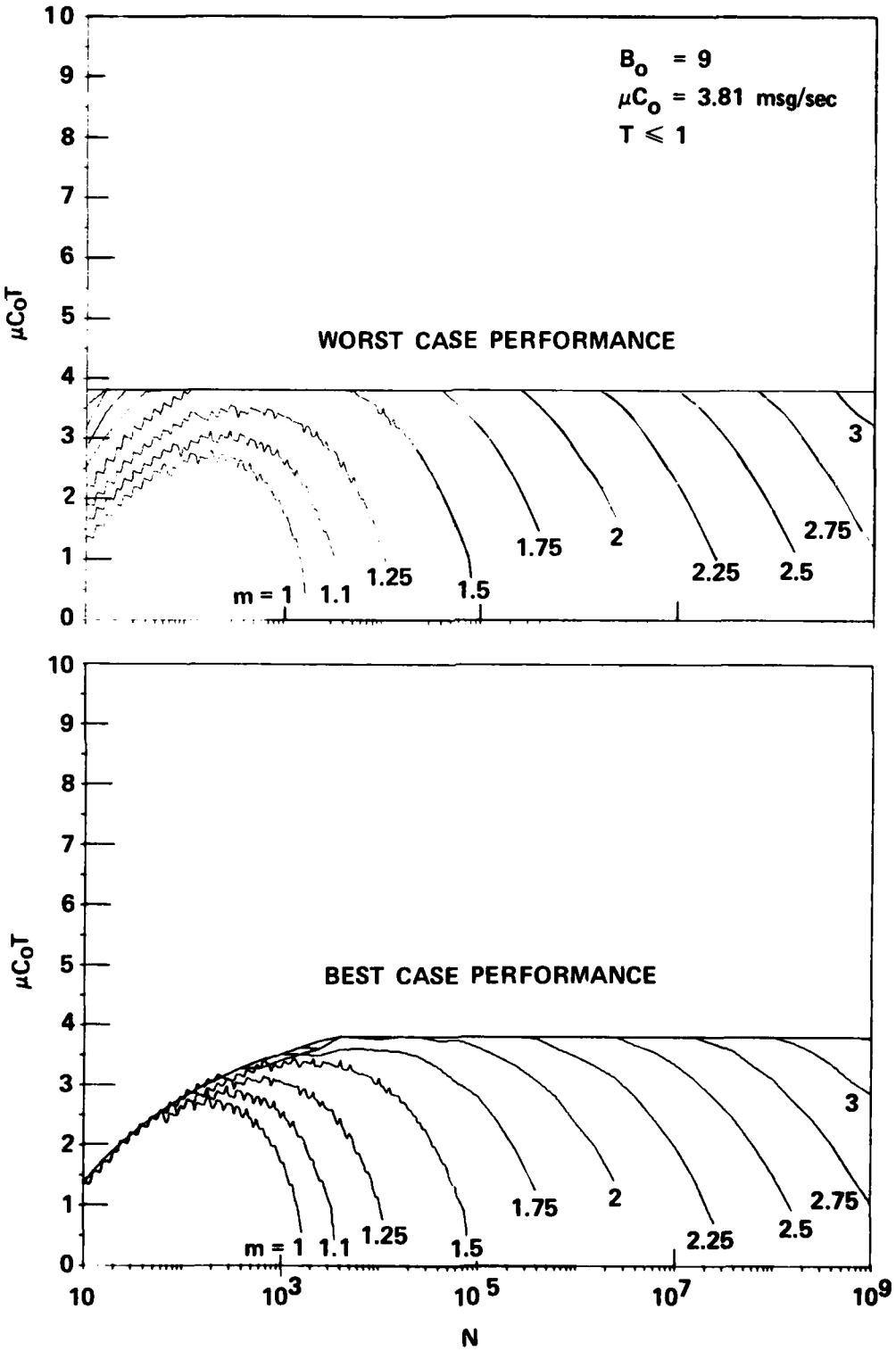


Figure 4.31. Network Delay at Maximum Throughput with a Delay Constraint.

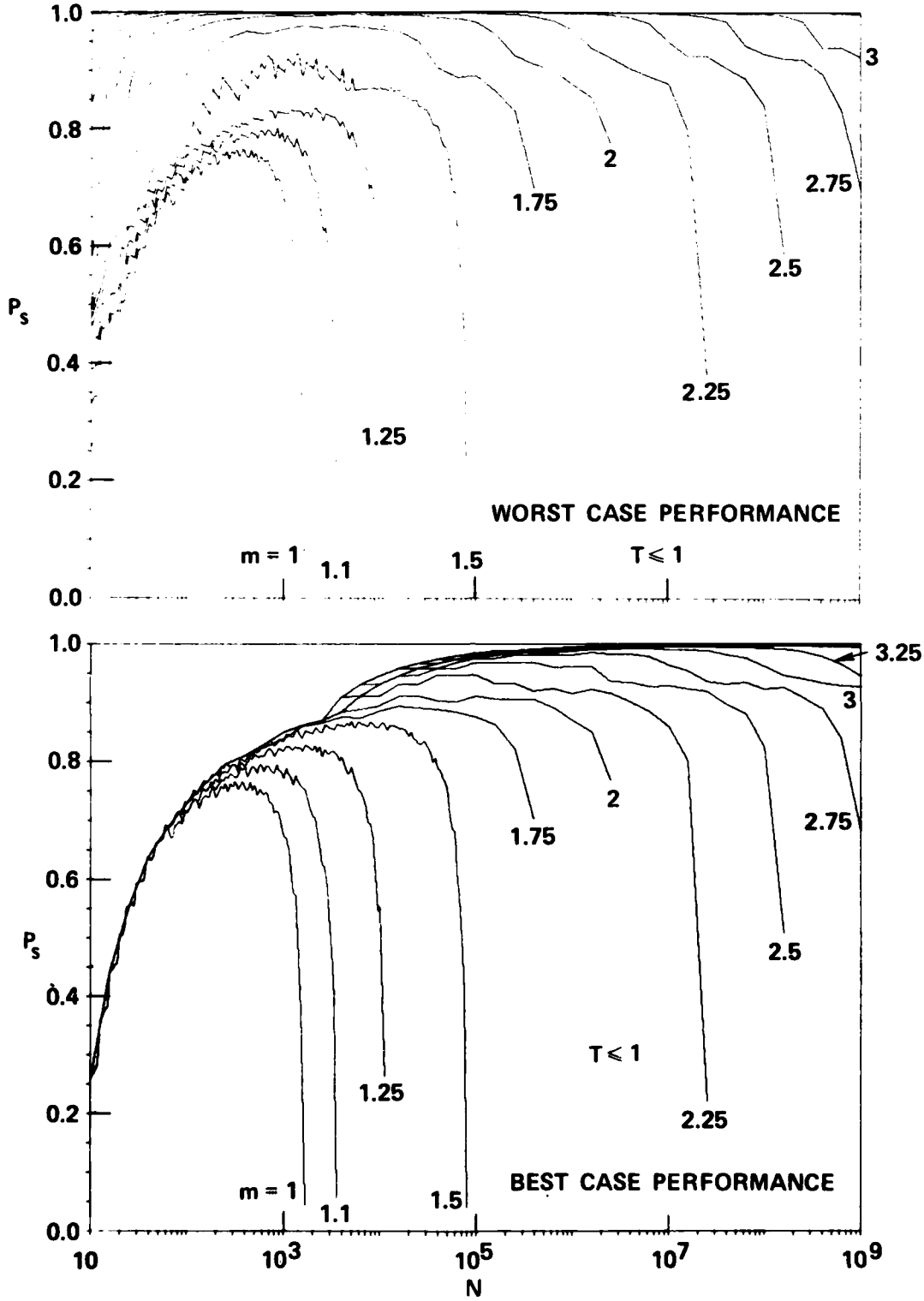


Figure 4.32. Probability of Success at Maximum Throughput with a Delay Constraint.

(This fact is also illustrated in Fig. 4.28). As a consequence, the gap existing between the lower and upper bound envelopes is mainly caused by the increase in the path length Δ.

Finally, let us notice that the performance of a non-hierarchical routing ($m = 1$) deteriorates very sharply for values of N around 1000, and that for N greater than roughly 250, hierarchical routing clearly becomes superior.

4.5 A Queueing Model with Updates and Storage Limitation

In this section, we intend to put our previous efforts together in order to devise a model whereby both line capacities and nodal storage used by the routing are accounted for.

The R $M|M|1$ single node model with a finite number of buffers B must now be modified in order to accomodate for the updates. Because of the results in Section 4.3.1, a channel can be modeled by a HOL priority queue ($M|M|1$, $D|D|1$). This is, however, a major obstacle in an analytical solution. A more careful observation of the analysis of the HOL system (Eq. (4.33)) shows that the effect of the updates is primarily to reduce the line capacity available for data traffic from C to $(1 - \rho_u)C$. The secondary effect is the added term $\lambda_u/2(\mu_u C)^2$ in the numerator of Eq. (4.33); it is this effect that we will neglect in the sequel. Moreover, we will assume that the handling of updates utilizes some storage (working storage) other than the S/F area.

As a result of the above considerations, our study here is reduced to the one performed in Section 4.4.2.2 where C is to be replaced by $(1 - \rho_u)C$.

The results of a numerical example are shown in Fig. 4.33. The same environment which generated Figs. 4.25 - 4.32 is assumed here, except that C is now

$$\mu C = h\mu C_0 - \frac{\lambda_u}{\mu_u}\mu$$

The rate λ_u selected corresponds to our earlier compromising choice, i.e., $\lambda_u = \frac{1}{2} N^{1/4}\lambda_u^0$, where $\lambda_u^0 = 0.14\ \mu C_0$. Also recall Eq. (4.35),

$$\frac{\mu}{\mu_u} = \frac{1}{64} mN^{1/m}$$

In addition, μ is set to one.

Fig. 4.33 shows only the worst case performance[1] of the hierarchical routing. The results are quite comparable to those in Figs. 4.25 and 4.30. The effect of updates can be seen in the drop of the minimum normalized throughput by roughly 0.05, except for very large N's where the drop becomes very small. In addition, hierarchical routing becomes superior to non-hierarchical routing for N at around 180 instead of the previous 250.

By comparing Figs. 4.7, 4.25 and 4.33, we note that in this particular example the network is more storage-bound than capacity-bound.

Finally, the fairly flat shape of the lower bound envelope of Fig. 4.33 shows that the effect of adaptive routing in terms of nodal storage and capacity requirements has been gradually reduced (as N grows) so as to keep a smooth network behavior.

---

[1]The upper bound envelope obtained in Fig. 4.25 is also valid with this application.

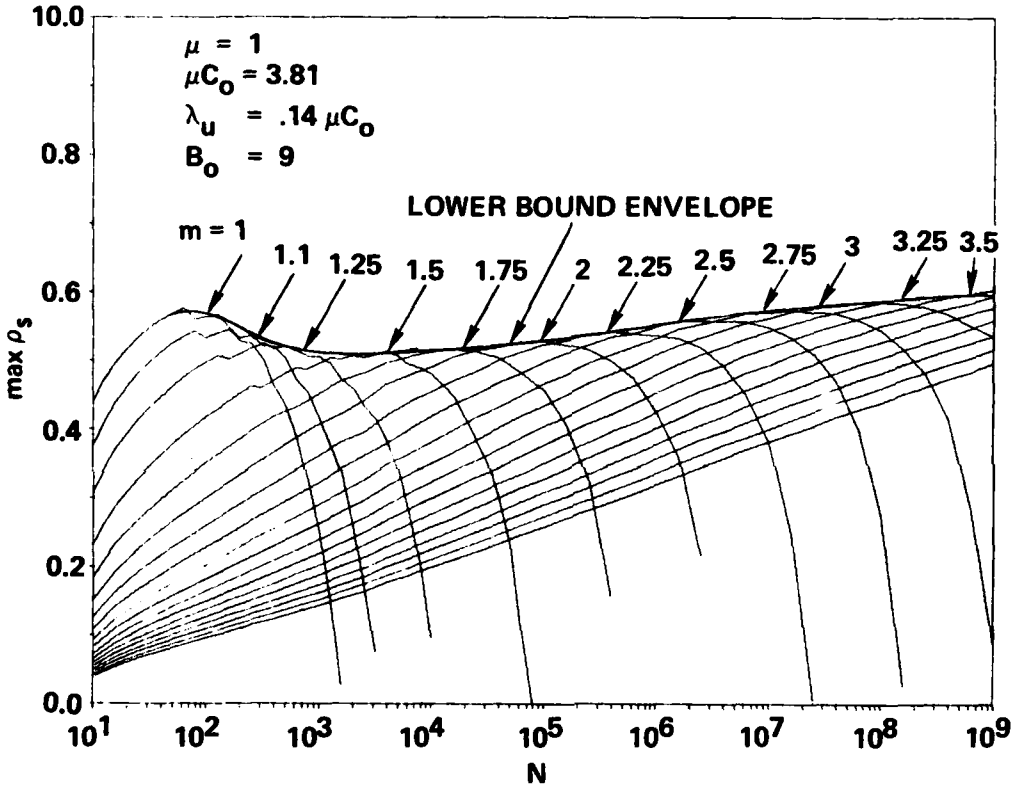


Figure 4.33. Maximum Lower Bound Throughput Obtained with the Hierarchical Routing; Model with Updates and with Storage Limitation.

4.6 Conclusion

In this chapter we were able to demonstrate that for a class of symmetrical and distributed networks:

i. In an ideal situation of sufficient storage and line capacity, the performance of a non-hierarchical routing is, in general, better than the one obtained with a hierarchical routing, except at the limit of very large nets where they become quite comparable, even for enormous table reductions.

ii. With a more realistic situation and with reasonable assumptions on network growth, the hierarchical routing becomes not only a necessity for large nets, but it also preserves a remarkably good network performance for a phenomenal range of the network size N.

The particular numerical examples studied in this chapter showed that the transition point where hierarchical routing becomes surely (because of lower bounds) better than a non-hierarchical one occurs for N between 100 and 200.

CHAPTER 5

CLUSTERING ISSUES AND SIMULATION OF THE MHR SCHEMES

In this chapter we address the following two issues:

(i) The assignment of nodes to clusters, clusters to superclusters, etc., given an arbitrary network and a clustering structure, (i.e., $\underset{\sim}{n}$, m).

(ii) The evaluation of the hierarchical routing schemes as applied to more general networks such as the ARPANET.

With respect to (i), first we define some nearness measures among the network nodes. Then, based on those measures, we choose (and slightly modify) the "Complete Linkage" (CL) technique as a solution for the clustering problem.

With respect to (ii), a simulation program has been developed in order to evaluate the performance of the MHR schemes (OBR, CER) under some fairly general operational environment.

5.1 Clustering Issues

The clustering problem is generally defined as one of finding natural groupings in a set of data, points, objects, etc. Numerous clustering techniques or procedures have been documented in the literature. We mention in particular the comprehensive book of Anderberg [ANDE 73] and other publications: [DUDA 73], [KERN 70] and the bibliographies therein.

In our context of computer networks, we need to define what we mean by a natural grouping. In what sense are we to say that certain

nodes are more likely to belong to the same cluster than others. This question actually involves two separate issues: how one should measure the similarity (nearness) between two nodes, and how one should carry out and evaluate the partitioning of a set of nodes into clusters. In this section, we address these issues based on some observations and previous conclusions related to what we consider as a good clustering.

5.1.1 Characterization of a "Good" Clustering

Let us temporarily restrict our considerations to a 2-level clustering, i.e., the partitioning of the set of nodes into disjoint subsets (clusters). Recall that with a hierarchical routing, routing tables contain one entry per node for nodes in the same cluster and one entry per cluster. Also, traffic between nodes in the same cluster must follow paths internal to that cluster. Furthermore, a given node must send all its traffic to a given cluster on the same path up to that cluster. The above rules lead us to distinguish two cases with regard to the internal and external properties a cluster should exhibit.

(a) Internal Characteristics. The internal behavior of a cluster should take advantage of the complete routing information at each node. As a result, cluster subnets must include most of the "best" paths between nodes in that cluster. More importantly, since internal traffic is confined to paths internal to the cluster, the cluster subnets must contain the shortest paths between its nodes. This last property was assumed to be true in the derivation of some of the earlier bounds on the increase in the network path length (see Chapter 3).

To take further advantage of the complete routing information, in case of non-uniform traffic conditions (i.e., γ_{jk}'s are not necessarily equal), clusters should be composed of nodes with higher intertraffic rates.

(b) External Characteristics. Due to the reduction of "external" routing information, a cluster, as seen from the outside, is equivalent to a single "super-node." An increase in the network path length has occured (Chapter 3) because of messages entering a cluster from a non-optimal entry point (exchange node). The single most important variable which affects this increase is the diameter of a cluster, or more precisely, it is the distance between the entry points of a cluster. As a result, clusters should be chosen so as to correspond to highly connected sets of nodes which thus lead to a small diameter and, moreover, to a small average internal path length. In case of low-connected nets, long chain clusters must be avoided.

Under non-uniform traffic conditions, clusters should also be selected so as to minimize intercluster traffic which is prone to utilize longer paths.

The above considerations can easily be extended to an m-level hierarchical clustering. Furthermore, they provide us with guidelines as to the choice of nearness measures and clustering techniques. Before we proceed, let us notice that existing nets (e.g., ARPANET, TRANSPAC) exhibit the above features to a certain extent. In particular, nodes tend to be localized in highly populated areas, hence they constitute natural groupings. This fact is more likely to become predominant for large networks, and this is a further motivation for the MHR schemes.

5.1.2 Nearness Measures

In order to cluster the set of network nodes, it is necessary to have some numerical similarity neasurements to characterize the relationships among the nodes. In other words, a measure of association must be computed for every pair of nodes (or for subsets of nodes in case of very large nets). Any such measure must reflect the above properties expected of a cluster, with the understanding that strongly associated nodes are more likely to belong to the same cluster.

The above properties of a good clustering may be simply summarized by stating that a cluster must preferably be composed of a subset of nodes, as highly connected as possible, and which interchange as much traffic as possible.

The above considerations lead to the following intuitive nearness measures. Let s_{ij} denote the similarity (nearness) between nodes i and j. The most obvious similarity measure between two nodes is the inverse of the hop distance between them, i.e.,

$$s_{ij} = \frac{1}{h_{ij}} \tag{5.1}$$

h_{ij} is as defined in Chapter 3.

In order to make the above measure sensitive to the traffic pattern, we modify Eq. (5.1) as

$$s_{ij} = \frac{\gamma_{ij}}{h_{ij}^{\alpha}} \tag{5.2}$$

Where α is a positive scaling exponent which reflects the importance of the variable h_{ij} as compared to γ_{ij} in the evaluation of s_{ij}. $\alpha = 0$ means that s_{ij} is only sensitive to the traffic rates.

The above external characteristic of a cluster motivates us to seek the grouping of nodes which, seen from the outside (i.e., from other nodes), exhibit some sort of unity. For this purpose, we define a matching variable between arbitrary pairs of nodes i,j as seen from an arbitrary node k: m_{ij}^k . The function of this variable is to, somehow, reflect to what extent the routing table at node k indicates the same route (i.e., same next-node) for traffic destined to i or j. m_{ij}^k is, therefore, closely related to the routing scheme utilized and can be determined either in a "static" way or through measurements.

The static evaluation of m_{ij}^k relies on the computation of the paths which result from a given <u>deterministic</u> routing policy (e.g., shortest path routing, flow deviation method [GERL 73A], etc.). If we consider a shortest path routing, then m_{ij}^k is set equal to one if node k indicates the same next node on the paths for i and j; and it is set equal to zero otherwise. Next, we define the nearness between nodes i and j as the average of m_{ij}^k over all nodes k,

$$s_{ij} = \frac{1}{N - 2} \sum_{\substack{k \neq i \\ k \neq j}} m_{ij}^k \qquad (5.3)$$

As mentioned above, m_{ij}^k can be determined through measurements such as the ones performed on the ARPANET [KLEI 74]. Such measurments (of interest to our study and others) have been carried out by the Network Measurement Center, NMC, at UCLA. They consist of collecting regular snapshots of the routing tables over a certain period of time. A matching value is then defined for each snapshot, say y, as $m_{ij}^k(y)$ which is then equal to one if node k's RT shows the same next node for

the routes to i and j (and zero otherwise). If Y is the total number of snapshots, then m_{ij}^k is defined as

$$m_{ij}^k = \frac{1}{Y} \sum_{y=1}^{Y} m_{ij}^k (y) \qquad (5.4)$$

Finally, s_{ij} is as defined in Eq. (5.3). These last two measures will be applied below to the ARPANET.

Further measures may be defined as a combination of Eqs. (5.2) and (5.3), or else as based on other network characteristics. In this study we restrict our considerations to the above intuitive and simple measures in order to achieve the network clustering.

5.1.3 Clustering Techniques

The nearness measures discussed above may be used as input to some clustering techniques which then realize the grouping of nodes into clusters. From among the existing clustering techniques [ANDE 73] the Complete Linkage (CL) method seems to be the most appropriate to our study. This method belongs to the family of hierarchical clustering techniques. It operates on a similarity matrix, $s = [s_{ij}]$, to construct a tree depicting specified relationships among the nodes. More precisely, with CL, each node starts as a cluster. Then, at each step the most similar pair of clusters, say p and q, are merged into a new cluster, called t. The similarity between the new cluster t and some other cluster r is determined as follows:

$$s_{tr} = \min \{s_{pr}, s_{qr}\} \qquad (5.5)$$

As such, s_{tr} is the similarity between the two most dissimilar pairs in clusters t and r. (Alternatively, if s_{ij} is a distance-like measure (e.g., $s_{ij} = h_{ij}$) then, $s_{tr} = \max \{s_{pr}, s_{tr}\}$).

The main property of this method is that all nodes in the same clusters are linked to each other at some maximum distance or minimum similarity. As a result, this method tries to identify "maximally connected subgraphs" which is precisely our main objective.

This last property can also be used as a stopping criterion by defining a threshold Th, beyond (or below) which no merging is allowed (i.e., if $s_{ij} < Th \ \forall \ i,j$ then STOP).

Another stopping criterion may consist of specifying the total number of clusters (n_2).

A shortcoming of CL is that it may lead to various non-optimal cluster sizes (see Proposition 2.1) which may considerably reduce the gains in table length, ℓ, obtained from optimal size clusters. Therefore some size limitation must be introduced. This can be realized by not allowing the merger of two clusters whose combined size is greater than the prespecified limit. This limit may be chosen as the optimal integer size n_1 (see Proposition 2.6). Furthermore, CL can be extended, as shown below, to accomodate a hierarchical clustering with more than 2 levels, provided we are given a vector of <u>maximum</u> degrees, $\underset{\sim}{n} = (n_1, \ldots, n_m)$.

Similar to the above, let p,q be the most similar pair of clusters, which are such that p is an ith level cluster and q, a jth level cluster. Also let $n_i(p)$ (and $n_j(q)$) be the number of $i\text{-}1^{st}$ level cluster ($j\text{-}1^{st}$) in p (q). Three cases to consider are,

(a) $\underline{i = j}$: Merge p and q into t;

If $n_i(p) + n_j(q) \leq n_i$, then t is an ith level cluster of degree $n_i(t) = n_i(p) + n_j(q)$.

Else, t is an $i+1^{st}$ level cluster of degree $n_{i+1}(t) = 2$.

(b) $\underline{i < j}$: The ith level cluster p is augmented with j-i levels, each of degree one, to become a jth level cluster; then Step (a) is performed.

(c) $\underline{j < i}$: The same as above for q.

Initially, each node, say p, is considered to be a 1st level cluster of degree $n_1(p) = 1$ (i.e., it contains one 0th level cluster). Furthermore, n_m, the maximum number of $m\text{-}1^{st}$ level clusters, is usually set to be large enough that the merging of two $m\text{-}1^{st}$ level clusters always results in a $m\text{-}1^{st}$ level cluster (see Step (a)). Either that, or the merging of the pair p,q is not allowed.

The above clustering technique does not guarantee that cluster subnets are connected. This is, however, to be expected of such a method (see below) To accomodate for this deficiency, a connectivity check may be performed before allowing the merger of two clusters. This check may be performed by using a distance-like measure d_{ij} which initially is set equal to h_{ij} and after a merging of clusters p,q into t, then

$$d_{tr} = \min \{d_{pr}, d_{qr}\}$$

As a result, a merging of p and q is allowed only if $d_{pq} = 1$.

Application to the ARPANET

The above modified complete linkage method (without a connectivity check) has been implemented and applied to the ARPANET. Fig. 5.1 shows a map of a 52 (duplex) channel, 46 node ARPANET as it was in operation on June 25, 1974. CL used as input the nearness measures of Eq. (5.3) as computed, first, from measurements performed at UCLA and second, from a shortest (hop) path routing.

(i) Clustering based on measurements. Fig. 5.2 illustrates the outcome of CL (as obtained using the measured nearness) in terms of the relative table length ℓ/N, and the threshold Th. Th is defined as the minimum nearness between any pair of nodes which belong to a common cluster at any level. The curves are shown for m = 2, m = 3, and for a variable m which achieves the lower envelope for ℓ/N. For $m \geq 3$, the maximum degrees were assumed to be: $n_1 = n_2 = \ldots = n_{m-1} = 3$, as motivated by Proposition 2.5; n_m was left unbounded.

The curve corresponding to m = 2 was obtained with no constraints on (n_1, n_2). It reflects the fact that with CL, initially, all nodes are considered to be clusters; hence Th = 1, $\frac{\ell}{N} = 1 + \frac{1}{N} \approx 1$, and at the last step of CL all nodes are in the same cluster. Thus, $\frac{\ell}{N} = 1 + \frac{1}{N} \approx 1$ with Th = 0.11. The minimum (with m = 2) of ℓ is equal to 17 (recall N = 46) and corresponds to the clustering structure shown in Fig. 5.3. This structure is composed of 11 clusters, with the largest cluster containing 6 nodes. The suboptimum $\ell = 17$ is fairly close to the theoretic minimum $\bar{\ell} = 2\lceil N^{1/2} \rceil = 14$. This shows the fairly good behavior of CL in this particular example. Let us also note that at that minimum point, the resulting cluster subnets are connected.

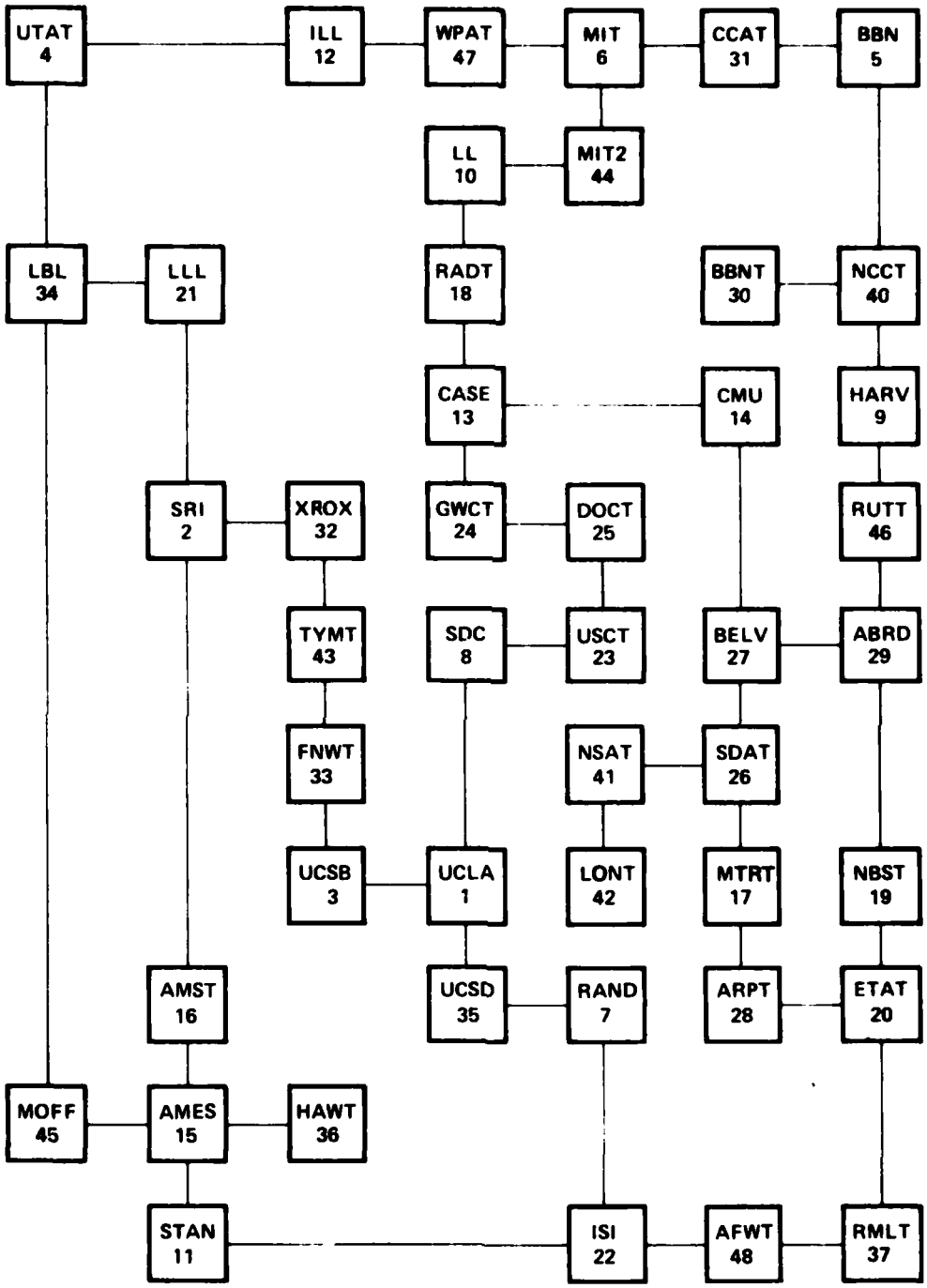


Figure 5.1. ARPANET, Planar Map as of June 1974.

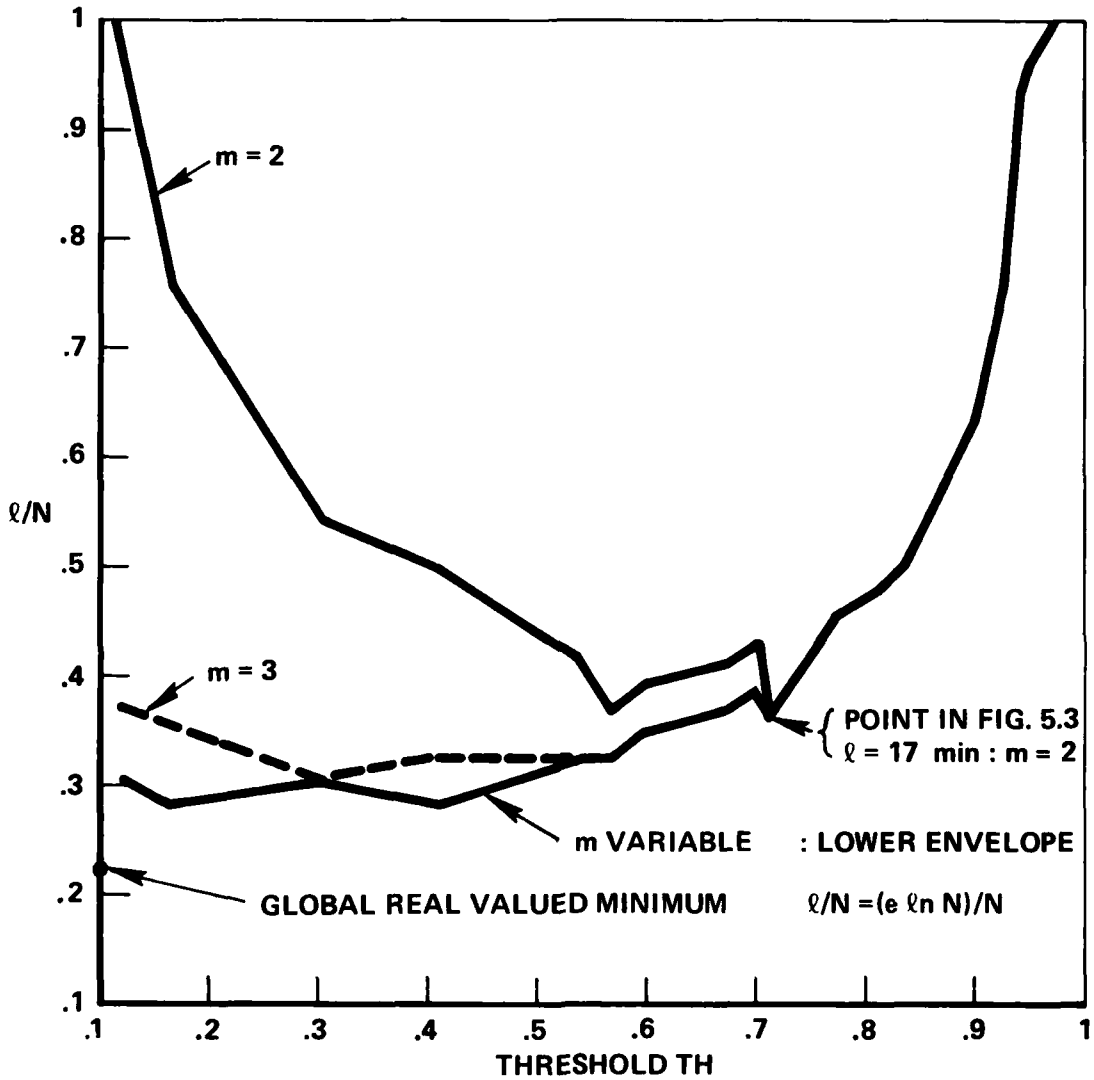


Figure 5.2. Minimum Nearness (Threshold) in the "Measured" Clustering of the ARPANET.

With a variable m, the global suboptimal point is reached for $\ell = 14$, whereas the theoretic global minimum is $\ell_* = 11$.

(ii) Clustering based on the shortest path matching. With this measure, CL leads to the following suboptimal points:

$$m = 2 \quad \Rightarrow \quad \ell = 15$$

$$m \text{ variable} \quad \Rightarrow \quad \ell = 13$$

They represent a slight improvement over the previous ones. For the sake of comparison with the earlier outcome, we show in Fig. 5.4, the clustering structure which yields $\ell = 17$; this structure is also composed of 11 clusters, with the largest containing 6 nodes. The clustering structures of Fig. 5.3 and 5.4 do not exhibit strong similarities. This can be explained by the quite asymetrical traffic pattern of the ARPANET [KLEI 74] which led to the measured nearness.

The evaluation of the MHR schemes based on such clustering outcomes is the object of the next section.

5.2 Simulation of the MHR Schemes

5.2.1 The Simulator

A PL1 simulation program has been developed[1] to help investigate the effects of hierarchical adaptive routing on the performance of arbitrary networks. The computer storage and computational requirements for simulating moderate size (a few hundred nodes) networks placed a

[1] A thorough documentation of this program is available at UCLA, Network Measurement Center.

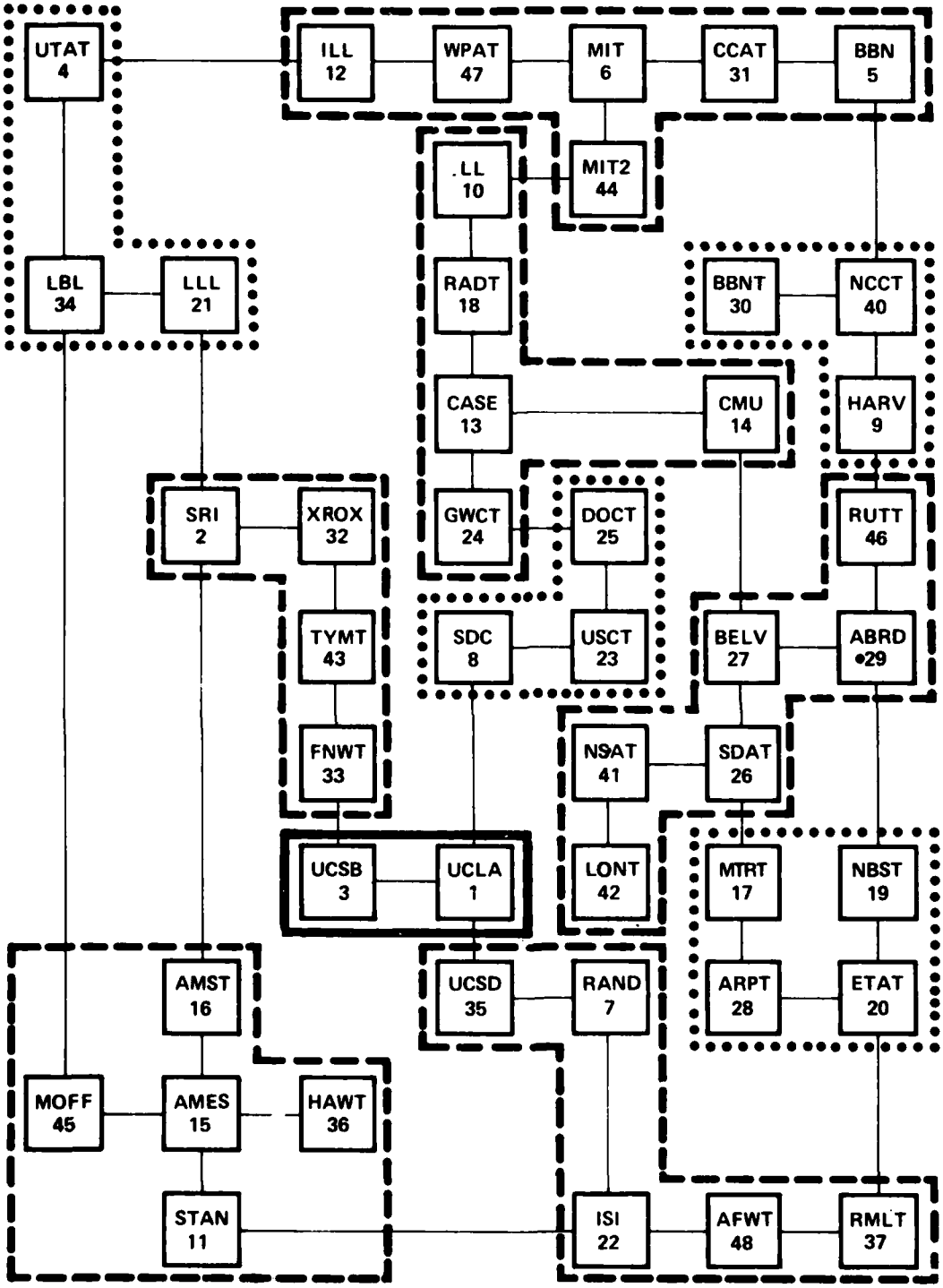


Figure 5.3 Sample Outcome of the Clustering Technique Based on "Measured" Nearness Measure; $\ell = 17$.

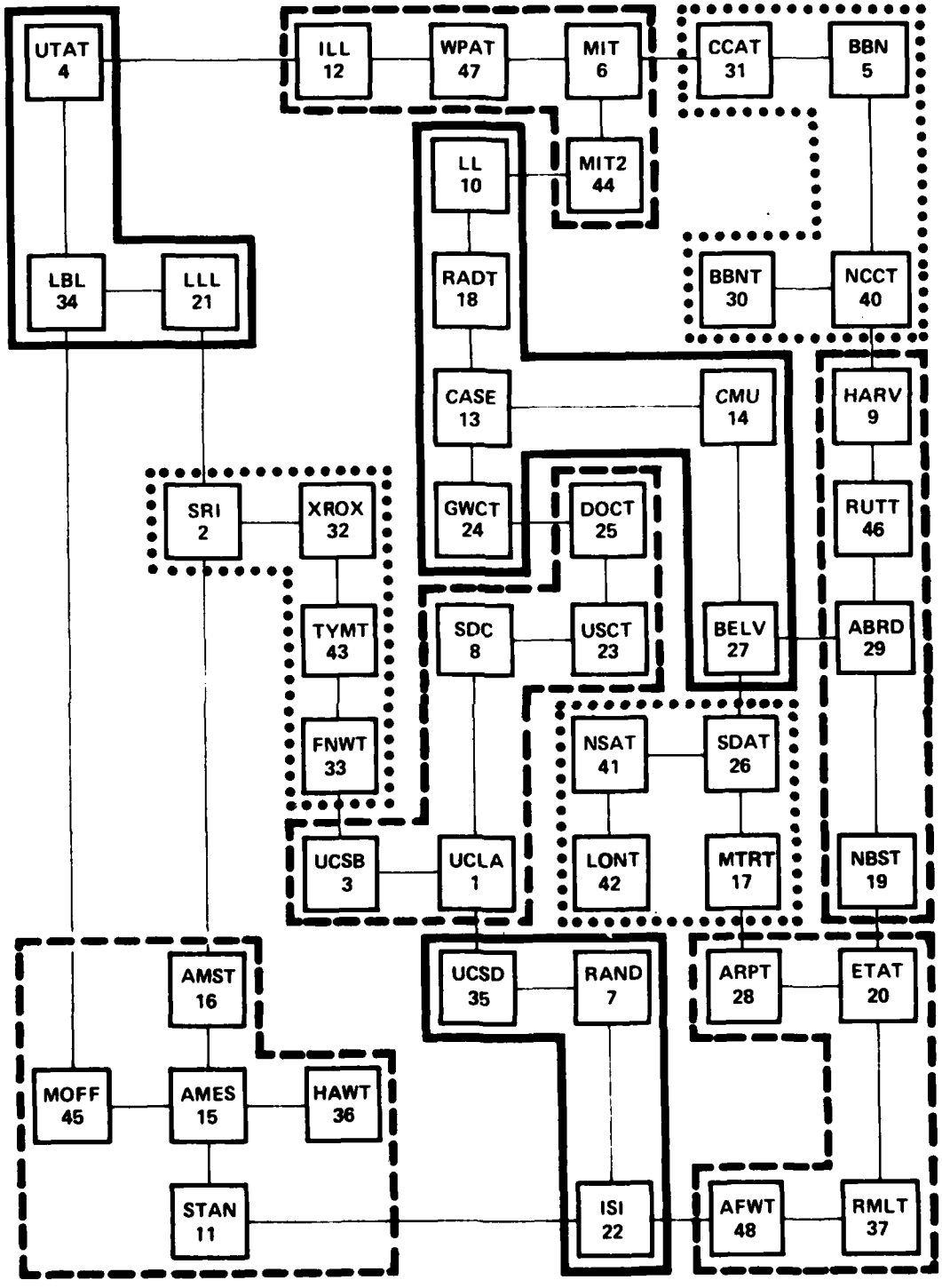


Figure 5.4. Sample Outcome of the Clustering Technique Based on "Static" Nearness Measure; $\ell = 17$.

restraint upon the amount of detail which could be implemented in the simulator. We list below the important features implemented in the simulator.

1. Messages are assumed to be single packet messages (hence, there will be no reassembly problems) whose lengths may be constant or drawn from a truncated exponential distribution.

2. All channels are assumed to be duplex channels and error free.

3. Each node (IMP) may be composed of a finite or infinite number of S/F buffers. In case of a finite storage, the sharing with maximum queue length (SMXQ) scheme is applied (see Appendix B).

4. The routing updates are periodic and occur every τ seconds. The updates are not synchronized among nodes. On the contrary, initial update times are selected randomly on a node pair basis (or simply on a node basis).

The update rule is performed as specified in Chapter 3 (for NCR and MHR schemes) except that in Algorithm 3.9, the channel length a_{ts} is now a random variable. When an update occurs, the channel length a_{ts} is estimated, as is presently done in the ARPANET. If we let k_1 be the line capacity index and k_2 the queue index ($k_1 = 4$, $k_2 = 1$ for the ARPANET), then

$$a_{ts} = k_1 + k_2 q_{ts}$$

where q_{ts} is the current queue length (in packets) for line (t, s).

A further modification is introduced to Algorithm 3.9 in order to avoid chain loops [NAYL 75]. A change in routing at node t is

allowed only if entry $C_j(i)$ at s is not pointing to t. Let $NN(s, C_j(i))$ denote the next node field at entry $C_j(i)$ of node s RT (see Fig. 3.1), then the first step of Algorithm 3.9 is modified as

$$\text{IF } H(t, C_j(i)) > a_{ts} + HF(s, C_j(i)) \text{ AND } NN(s, C_j(i)) \neq t.$$

5. Acknowledgements (ACK) are assumed to be instantaneous. A message forwarded from one node to another can be rejected if there is no space available (recall that S/F buffers are managed by the SMXQ scheme). This situation triggers a negative ACK (a departure from the ARPANET protocol) upon reception (instantaneous) of that ACK, the sending node reschedules the retransmission of that message. The message is deleted when a positive ACK is received.

6. With respect to the flow control, messages are not accepted in the net for either of the following reasons:

(i) There is no buffer space at the originating node.

(ii) The total number of messages in the network is equal to a given critical number, NC.

The rejected messages are considered to be lost.

This terminates the description of the main features of the simulator; we now proceed with some applications.

5.2.2 Simulation of the ARPANET

Our previous clustering structures of the ARPANET (see Figs. 3.1, 3.3 and 3.4) are now used as input to the simulator in order to compare the performance of the NCR and the MHR schemes (CER). More specifically, the operating conditions are as follows:

Topology: 46 node ARPANET, Fig. 3.1. It is composed of 52 duplex channels of capacity 50 kbit/sec per channel, except for lines (16, 15) and (45, 15) which are of capacity 230.4 kbit/sec. The propagation delay on the lines is set to zero.

MHR: The net is operated with a non-clustered (NCR) adaptive routing and with a 2-level closest entry routing (CER) scheme. The underlying clustering structures of CER are as shown in Figs. 5.3 and 5.4. In what follows, we will refer to the representation of Fig. 5.3 as AMC, which stands for ARPANET Measured Clustering, and ASC, ARPANET Static Clustering, for the representation in Fig. 5.4. The non-clustered version will be referred to as ANC (N for "non").

Traffic Characteristics: We assume a uniform traffic matrix, i.e., $\gamma_{ij} = \gamma \ \forall\, i, j$; therefore, the network throughput is equal to $N(N - 1)\gamma = 2070\ \gamma$. The message length is assumed to be fixed and equal to 1 kbit.

Update Information: $k_1 = 4$, $k_2 = 1$. Update interval, $\tau = 0.64$ sec. The initial update times are chosen randomly on a node-pair basis. The length of an update message is set equal to 1 kbit with or without clustering.

Node Operating Condition: We assume an infinite nodal storage, and a nodal processing time equal to zero.

The results of the simulation are shown in Fig. 5.5. Let us first notice that the maximum throughput is achieved for $\gamma_{max} = 0.274$ msg/sec. This value can be computed by considering the

cut set (line cut set) with minimal capacity and maximum through traffic. Channels (24, 13), (12, 47) and (48, 37) partition the net into two equal subsets of nodes (23 nodes each) and thus they constitute such a cut set. The traffic in one direction is equal to

$$(23)^2 \frac{\gamma}{\mu} + 3 \frac{\lambda_u}{\mu_u}$$

where $1/\mu = 1/\mu_u = 1$ kbit and $\lambda_u = 1/\tau = 1.562$. The capacity of this cut (in one direction) is equal to 150 kbits, and $\gamma = .274$ msg/sec achieves such a maximum traffic.

Fig. 5.5 shows that NCR leads to a better performance, mainly for $\gamma \geq .2$ msg/sec. This is to be expected, since this application did not account for the savings obtained from the MHR schemes. Such savings are, however, relatively small due to the small size of the ARPANET (46 nodes).

Furthermore, the static clustering, ASC, leads to a better performance than the measured clustering, AMC. This is mainly due to the uniform traffic assumption.

The difference in performance between the NCR and the CER may be further explained as due to the low connectivity of the net, which leads to chain cluster subnets. The average distance between entry points of a cluster is approximately 3 for an average network path length of approximately 6.

The above results still intuitively indicate that for an ARPANET with more than 64 nodes (at which point the RT uses more than one buffer), one may expect that an MHR with a table length $\ell \approx 64$ will start to show some improvement over the NCR scheme.

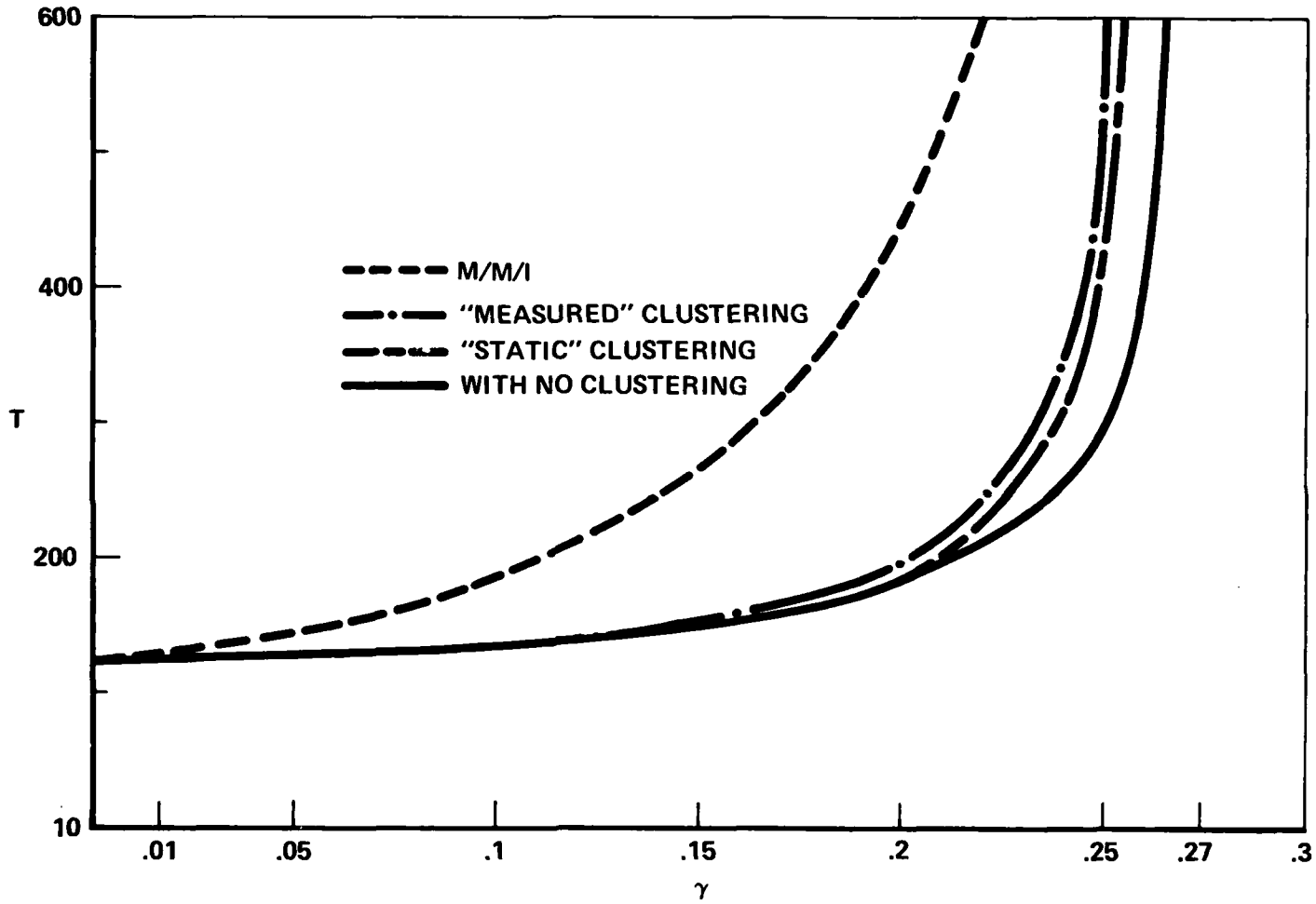


Figure 5.5. Simulation of the ARPANET as Operated with an MHR Scheme.

5.2.3 Simulation of a 64-Node Torus

We consider in this section a 64-node torus upon which we impose two clustering structures (see Fig. 5.6). In this particular example, we account for the savings obtained from the reduction in table length; furthermore, we use the overall best routing (OBR) scheme to operate the network. More specifically, the operational environment for the simulation is as follows:

Topology: We assume a 64-node torus, composed of 128 duplex channels of 20 kbit/sec each. The propagation delay on the lines is chosen as equal to zero.

MHR: The network is operated with an NCR and an OBR scheme. The OBR scheme is utilized with two clustering structures: a 2-level clustering composed of 16 clusters containing 4 nodes each (i.e., $n_2 = 16$, $n_1 = 4$), and a 3-level clustering with degree vector $n_1 = n_2 = n_3 = 4$. These structures are shown in Fig. 5.6; the first will be denoted by $16 \times 4C$ and the second by $4 \times 4 \times 4C$.

Traffic Characteristics: We assume a uniform traffic pattern. The offered load Γ is then equal to $\Gamma = N(N - 1)\gamma = 4032\gamma$. The message length is governed by a truncated exponential distribution. The mean length is set equal to 500 bits and the truncation point is at 1000 bits.

Update Information: Similar to the above, we choose $k_1 = 4$, $k_2 = 1$, $\tau = .64$ sec. The size of and update message $1/\mu_u$, is fixed and proportional to the table length; moreover, we assume each entry to be 16 bit long. Therefore,

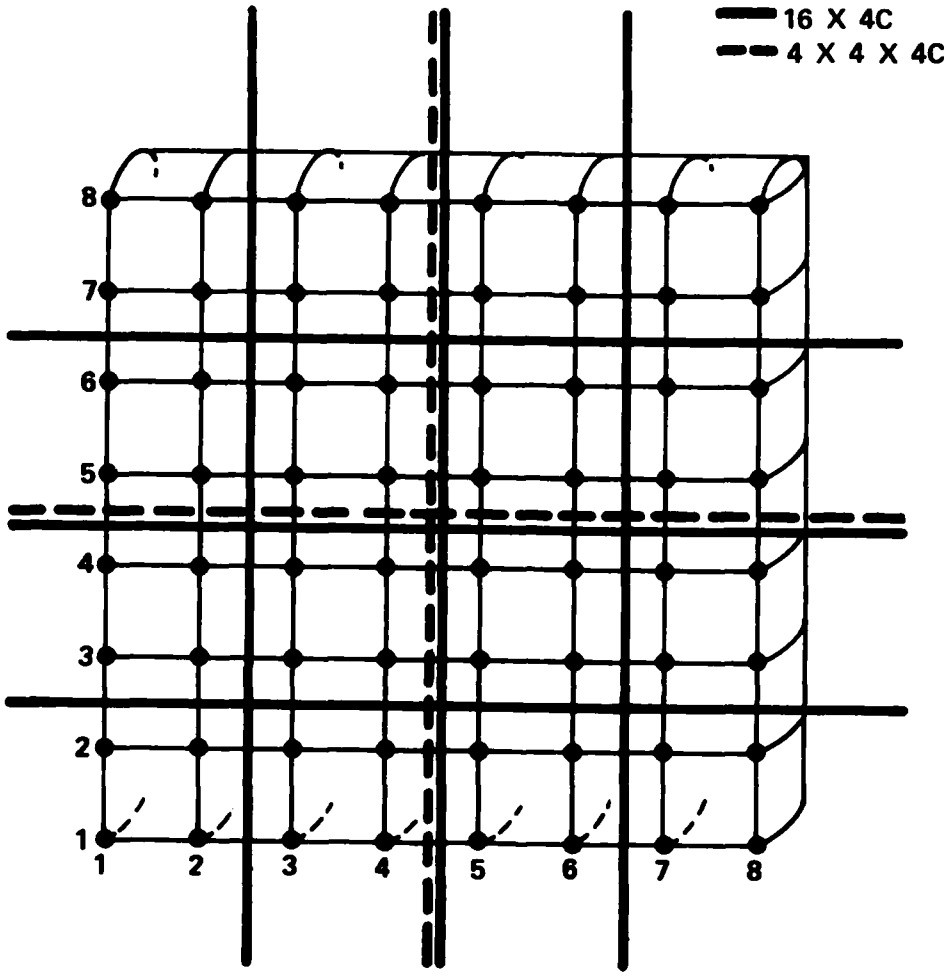


Figure 5.6. Clustered 64-Node Torus Network.

$$NC \Rightarrow \ell = 64 \Rightarrow 1/\mu_u = 1024 \text{ bits}$$

$$16 \times 4C \Rightarrow \ell = 20 \Rightarrow 1/\mu_u = 320 \text{ bits}$$

$$4 \times 4 \times 4C \Rightarrow \ell = 12 \Rightarrow 1/\mu_u = 192 \text{ bits}$$

Node Operating Condition: The nodal processing is set to zero. Each node is assumed to contain 16 S/F buffers. These buffers are shared between the 4 outgoing channels under the SMXQ sharing scheme, where 12 is the maximum length of any queue.

Results: Fig. 5.7 shows the results of the simulation under the above operating conditions. We observe that OBR with the "16 × 4" 2-level clustering achieves a better performance than NCR and OBR with 4 × 4 × 4C. The difference with NCR is not too considerable; this is due to the small size of our network (64 nodes) which thus does not yield to great savings in table reduction. However, this indicates that clustering starts to pay off, even for such a small network size. The OBR scheme with the 4 × 4 × 4C clustering structure achieves the worst performance; furthermore, is saturates faster than the two other cases. This is due to the fact that such a degree of clustering (high for this small net) leads to a significant increase in path length, hence in internal traffic.

Let us also note that for $\Gamma = 1700$, the network gets into a deadlock situation. This, and the bending of the curves at $\Gamma = 1600$, is due to the buffer limitation. A better flow control scheme is required to operate the net; recall that the only control introduced here is in terms of the total number of messages which can be in the

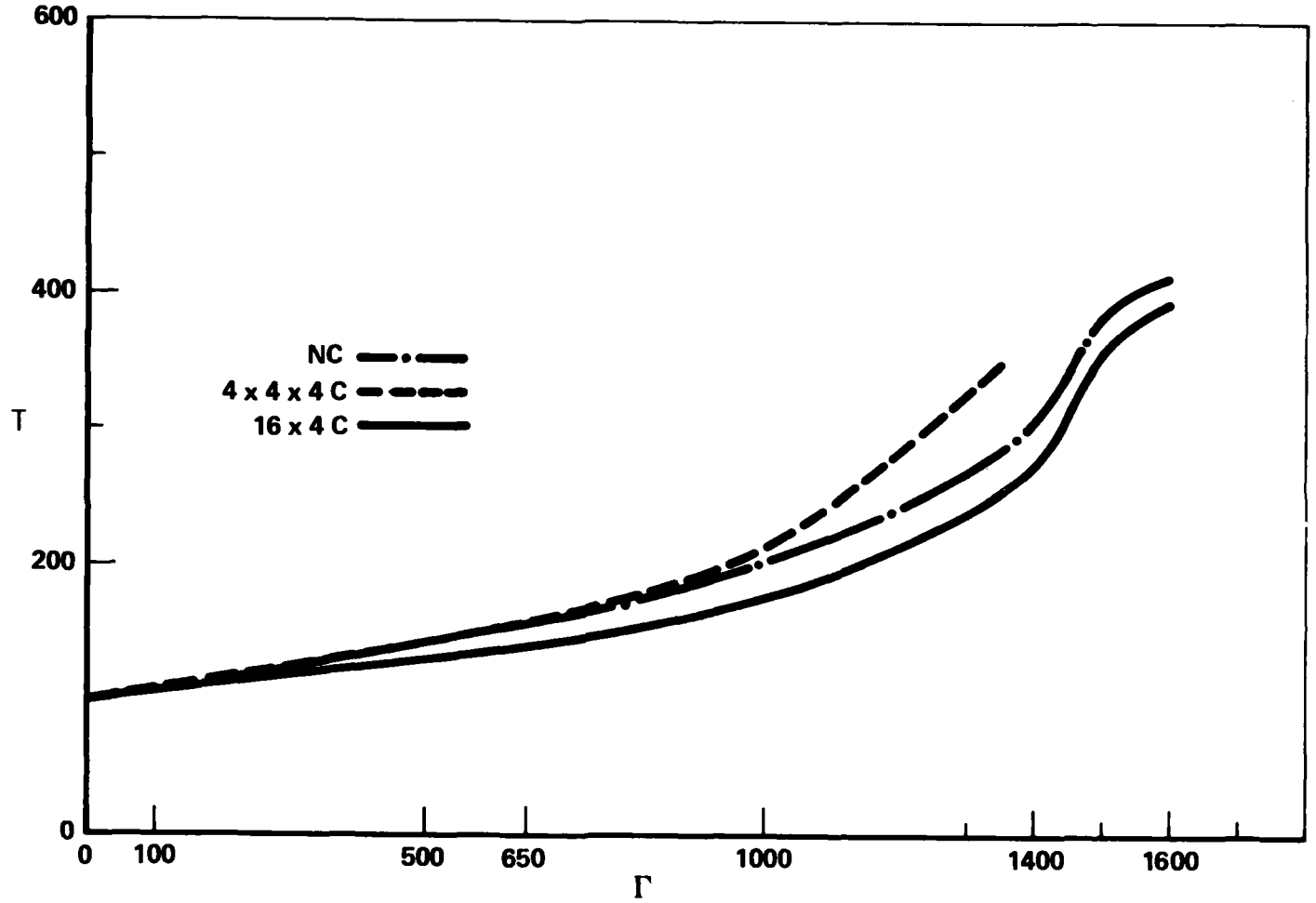


Figure 5.7. Simulation of a 64-Node Torus as Operated with the MHR Scheme.

network at any given time. That number was selected to be equal to 819 (approximately 4/5 of the total number of buffers in the network).

5.3 Conclusion

In this chapter we mainly laid the framework and introduced a methodology in defining nearness measures between pairs of nodes. Those measures were utilized with a modified complete linkage method to achieve the clustering of the ARPANET. Applications to larger networks should be investigated, along with other clustering techniques.

The simulation of a 64-node torus confirmed our earlier theoretic results which indicated that the MHR schemes with an appropriate clustering structure eventually (as the network grows in size) outperform the NCR scheme. The transition point was seen here to be even smaller, in the neighborhood of 64.

CHAPTER 6

TOPOLOGY DESIGN CONSIDERATIONS FOR LARGE COMPUTER COMMUNICATION NETWORKS

6.1 Introduction

Up to now, we were given a large distributed computer communication network, and the problem was to devise an appropriate adaptive routing scheme which would operate efficiently with a fairly small amount of routing information. The study of the MHR routing schemes showed that, indeed, they represent a class of very satisfactory solutions to the routing problem in large networks. Now, going one step back, the problem becomes that of designing the topology of a minimum cost communication network which connects a large set of nodes and which satisfies a set of given requirements (traffic, delay, reliability, etc.). Furthermore, it is desirable that the resulting network be fairly easy to operate (routing, flow control, adaptation to failures, etc).

The bursty nature of computer traffic, as well as the continuously decreasing cost of computer hardware [ROBE 74], very much favor packet switching, as currently used in the ARPANET [ROBE 70], [HEAR 70], [KLEI 70], [FRAN 70], [CARR 70], [MCQU 74], as the technology for us to consider. Several different formulations of the topology design problem can be found in the literature. Generally, they correspond to heuristic methods with different choices of performance measures, design variables and constraints. Along those formulations, several solutions have been proposed [GERL 73A]

[FRAN 72] and applied to ARPA-like network designs. However, for networks with more than a few hundred nodes, present procedures fail because of the large amount of computer time and storage needed to perform the optimization. Design procedures, based on hierarchical clustering, have been proposed [FRAN 73], [GERL 73B], [COVI 74] to substantially reduce the computational cost and the storage requirement involved in the optimization step. Generally speaking, in a two-level hierarchical design, the nodes will be grouped into clusters and "gates" (special cluster-exchange nodes) selected from each cluster. Cluster subnets will be designed separately, and then a supernet of gates will be designed to connect the clusters together. The assumption is that nodes in the same cluster are most likely to be very few hops apart in either a non-hierarchical or hierarchical design. The approach described briefly above could be easily extended to more than two levels in the hierarchy.

In this chapter, after stating the design problem in a more precise form, a class of hierarchical design procedures will be presented. The focus will be upon the determination of a certain optimum hierarchical clustering structure of the set of the nodes, to be used in the design phase. In other words, we will be concerned with the choice of the cluster sizes at any level and the number of levels which will minimize a certain objective function (e.g. computational cost). Some related questions, such as the decomposition of the global performance variables and requirements, will also be addressed.

Frank and others [FRAN 73] showed from a feasibility study of a 1000 node network that, indeed, hierarchical structures are desirable

for the design of large networks. They also posed the same questions concerning the clustering structure but failed to answer them for the general case (m levels).

6.2 The Topology Design Problem

The same network model as in Paragraph 6.1 is used in this section.

6.2.1 Delay Analysis

Recall Eq.(4.5)

$$T = \frac{1}{\Gamma} \sum_{i=1}^{NA} \frac{\lambda_i}{\mu C_i - \lambda_i}$$

More elaborate expressions for T [FULT 72], [KLEI 74] may be obtained to include overhead traffic, nodal processing delay, etc. For the purpose of illustrating the complexity of the design problem, we will limit our consideration to the simple expression above. In the design problem, T will appear either as the objective function to be minimized or as a variable constrained to be less than or equal to a given value, T_{max}.

Let $f_i^{(k,\ell)}$ be the average flow (bits/sec) produced in channel i by messages traveling from source k to destination ℓ (flow of commodity (k, ℓ)), and let f_i be the total average flow in channel i, given by:

$$f_i = \sum_{k=1}^{N} \sum_{\ell=1}^{N} f_i^{(k,\ell)}$$

Consequently $$f_i = \frac{\lambda_i}{\mu}$$

and $$T = \frac{1}{\Gamma} \sum_{i=1}^{NA} \frac{f_i}{C_i - f_i} \tag{6.1}$$

Let $\underset{\sim}{f} \triangleq (f_1, f_2, \ldots, f_{NA})$ be the flow vector, and let $\underset{\sim}{C} \triangleq (C_1, C_2, \ldots, C_{NA})$ be the capacity vector. For the above equation to be feasible, the vectors $\underset{\sim}{f}$ and $\underset{\sim}{C}$ must satisfy

$$\underset{\sim}{f} \leq \underset{\sim}{C} \qquad \text{i.e., } f_i \leq C_i \tag{6.2}$$

The above relation will serve as a constraint in the design problem.

6.2.2 The Communication Cost

To the ith link, we have assigned a channel capacity C_i. C_i is a discrete variable. However, for the development of efficient analytical techniques, it is often convenient to approximate C_i with a continuous variable [KLEI 64], [GERL 73A] during all (or part) of the optimization phase, after which these continuous values are discretized. Typical capacity options and their cost are listed in [GERL 74]. The cost of channel i, denoted $d_i(C_i)$, is a function of the capacity and the length of the channel[1]. In general, the cost shows economies of scale with respect to capacity, and it increases linearly with the length of the channel.

[1] More elaborate charging structures such as ATT's new DDS service have also been devised.

6.2.3 Traffic Requirement

Average "traffic requirements" between nodes can be represented by a matrix R

$$R \triangleq [\gamma_{jk}]$$

Recall that γ_{jk} (messages/sec) is the average transmission rate from source j to destination k. In general, R is given as an input parameter to the problem. In some cases, R is given as

$$R = \rho\overline{R}$$

where $\overline{R}$ is a known traffic pattern and ρ is a variable scaling factor, usually referred to as the traffic level.

As a result of the traffic requirements, the following multicommodity flow constraint [HU 69] must be satisfied.

$$\sum_{j=1}^{N} f_{ji}^{(k,\ell)} - \sum_{j=1}^{N} f_{ij}^{(k,\ell)} = \begin{cases} -\dfrac{\gamma_{k\ell}}{\mu} & \text{if} \quad i = k \\ +\dfrac{\gamma_{k\ell}}{\mu} & \text{if} \quad i = \ell \\ 0 & \text{otherwise} \end{cases} \qquad (6.3)$$

6.2.4 Reliability

Links and nodes in a real network can fail with non-zero probability, thus interrupting some communication paths. It is important to evaluate the overall network reliability in the presence of such failures. Several reliability measures are available: the probability of the network being disconnected, the fraction of communicating node pairs, the connectivity of the network, etc. [FRAN 72]. However, no simple satisfactory measure exists yet. In the design problem, for the

sake of simplicity, connectivity is usually used as a reliability measure.

This concludes our model description, and now we are ready to state the design problem more formally and then to discuss its complexity.

6.2.5 Topology Design Problem

The topological design problem can be defined as follows:

Given: Node locations

Requirement matrix R

Cost-capacity functions $d_i(C_i)$

Minimize: $D(A, \underset{\sim}{C}) = \sum_{i \in A} d_i(C_i)$

Over: $A, \underset{\sim}{C}, \underset{\sim}{f}$ (6.4)

where A is the set of arcs which corresponds to a specific topology, $\underset{\sim}{C}$ is a vector of capacities and $\underset{\sim}{f}$ is a multicommodity flow, such that

(a) $\underset{\sim}{f}$ satisfies the requirement matrix R (see Eq. (6.3))

(b) $\underset{\sim}{f} \leq \underset{\sim}{C}$ (see Eq. (6.2))

(c) $T = \frac{1}{\gamma} \sum_{i \in A} \frac{f_i}{C_i - f_i} \leq T_{max}$ (see Eq. (6.1))

(d) Reliability constraint: e.g., K-connectivity.

Notice that the above cost function associates the entire network cost with the channels themselves [KLEI 64], [GERL 73A]. This does not

ent a loss of generality with respect to the nodal costs which may be grouped with the channel costs.

In general, there are $2^{N(N-1)/2}$ possible topologies. Considering all possible designs by a computer is out of the question. Furthermore, capacities are available in discrete sizes. This means an enormous integer optimization problem must be solved.

The non-linearity of the time-delay functions and, in some cases, of the reliability measure add another dimension of complexity to the problem.

There exists no efficient technique for the exact solution of the topological design problem. Several heuristic procedures have been proposed and implemented. Among them, we mention the Branch X-change method [FRAN 72] and the Concave Branch Elimination method [GERL 73B]. They typically start with an initial topology over which they perform some alterations in the course of the optimization. Built into those procedures and inherent in the multicommodity nature of the flow, is the determination of the shortest path between any pair of nodes in the network. This operation requires approximately up to the order of N^3 operations (N^2 to N^3, N = number of nodes) and may be performed many times in the course of the optimization. The overall computational complexity corresponding to those heuristics is estimated to be on the order of N^3 to N^6 [FRAN 73], [FRAN 72].

For networks with more than a few hundred nodes, present procedures fail because of the large amount of computer time and storage needed to perform the suboptimization. Furthermore, for such

networks, incremental changes in the number of nodes ought to be considered on a local basis rather than on a global basis as it is performed now. In other words, addition or deletion of a node should not affect the topology of the entire network to a large degree. As a result, new approaches are needed to deal with the design of large networks.

6.3 The m-level Hierarchical Topology (MHT) Design Procedure

Various tools are available in the field of operations research to solve certain classes of large scale mathematical programming problems. Among those techniques are the Dantzig-Wolfe, Bender's, Lagrangian decomposition methods and the column generation method [LASD 70], [HIMM 72], [GEOF 74], [GRAV 72]. Due to the complexity of our design problem, the direct application of these techniques does not seem too promising and even less practical. Consequently, a good heuristic method, based on intuition and on the natural properties of the problem, is desirable. However, exact design procedures should be developed in order to serve as comparison for the evaluation of any heuristic design technique. Hierarchical design procedures are good candidates to alleviate the tremendous complexity of the problem and yet produce some good topological design. Such a procedure, the m-level hierarchical topology (MHT) design method, will be presented and studied in the rest of the chapter.

The idea behind the MHT is to impose a decomposable structure on the design problem which will result in a set of smaller sub-problems. In other words, we will introduce independencies among

subsets of design variables. The imposed independencies will substantially reduce the set of feasible solutions and also, as a direct consequence, the computational cost. In doing so, there is the risk of eliminating the optimal solution. Therefore, it is very important to seek "natural" decompositions.

The four major steps in an MHT are:

a. m-level hierarchical clustering (MHC) of the set of nodes.

b. Decomposition of the objective functions and the design constraints.

c. Decomposition of the optimization problem into a set of smaller subproblems. Solve for the set of subproblems with possible iterations.

d. Exact performance evaluation of the resulting network and comparison with optimum performance.

6.3.1 Step a: MHC of the Set of Nodes

The decomposition mentioned above will be realized through an m level hierarchical clustering of the set of nodes, based on some appropriate nearness measure. As defined in Chapter 2 (see Fig 2. and Paragraph 2.3.1 for definitions and notation), the MHC consists of grouping the network nodes (0th level clusters) into 1st level clusters which in turn are grouped into 2nd level clusters. This operation continues in a bottom up fashion until the grouping of the $m-2^{nd}$ level clusters into $m-1^{st}$ level clusters whose union constitute the mth level cluster. The mth level cluster is the highest level cluster, and as such, it includes all the nodes of the network. This partitioning, easily describable by a tree structure as in Fig. 2.4, could also

proceed in a top down manner.

The above structural similarities, with the MHC devised for the adaptive routing in large networks, do not quite carry over to the choice of the nearness measures and probably not to the clustering techniques, both of which will greatly affect the outcome of the MHT. The nearness measures must take into account the cost of the different components of a communication network (switching nodes, channels, etc.), the traffic and reliability requirements, the delay of a message in the net, etc. Some examples of simple nearness measures are:

(i) $s_{ij} = \frac{1}{d_{ij}}$ nearness between nodes i and j

(ii) $s_{ij} = \frac{\gamma_{ij}}{d_{ij}^{\varepsilon}}$

where d_{ij} is the geographical distance between nodes i and j;

γ_{ij} is the rate of traffic from i to j, and the exponent ε is between zero and one.

Here also, the direct application of the clustering techniques [ANDE 73], [DUDA 73] may lead to various non-optimal cluster sizes which will, in general, considerably reduce the computational gains [KERN 70] obtained from optimal size clusters. One way to decide upon an MHC structure is to choose the one which will minimize the computational cost incurred in the MHT. This problem will be posed and solved later in this chapter.

Along with the MHC, we must select the gates (exchange nodes) for all clusters at all levels (Fig. 6.1). The function of the <u>gates</u>

from a given cluster is to handle the traffic exchanged between the set of nodes in that cluster and those outside. More specifically, the assumption underlying the flow of messages is as follows.

Flow Assumption 6.1

a. Traffic between nodes in the same cluster, at any level, will only take paths which are internal to that cluster, i.e., paths contained in the corresponding local subnetwork.

b. Traffic between nodes in different k^{th} level clusters ($k = 1, \ldots, m - 1$), but which belong to the same $k+1^{st}$ level cluster, will first be channeled to a $k+1^{st}$ level gate of the originating cluster over its local subnetwork; then, it will take the $k+1^{st}$ layer subnetwork of gates to reach a $k+1^{st}$ level gate of the destination cluster, at which point it will be dispatched over the local subnetwork to finally reach the destination node. (This is the standard procedure in hierarchical networks.)

Fig. 6.1 illustrates some of the preceding and forthcoming definitions for a 3-level hierarchical design.

A local subnetwork of a kth level cluster, say C_k, is defined recursively as the collection of those local subnetworks associated with each $k-1^{st}$ level cluster $\{C_{k-1}(i)\}_i$ which belongs to C_k and which are connected together through a subnetwork of kth level gates called a kth layer subnetwork. Consequently, such a local subnetwork is composed of k hierarchical layers, each of which is composed of a certain number of layer subnetworks. As a reminder of the k layers, the subnetwork will also occasionally be called a k-level subnet or a k-level local subnet.

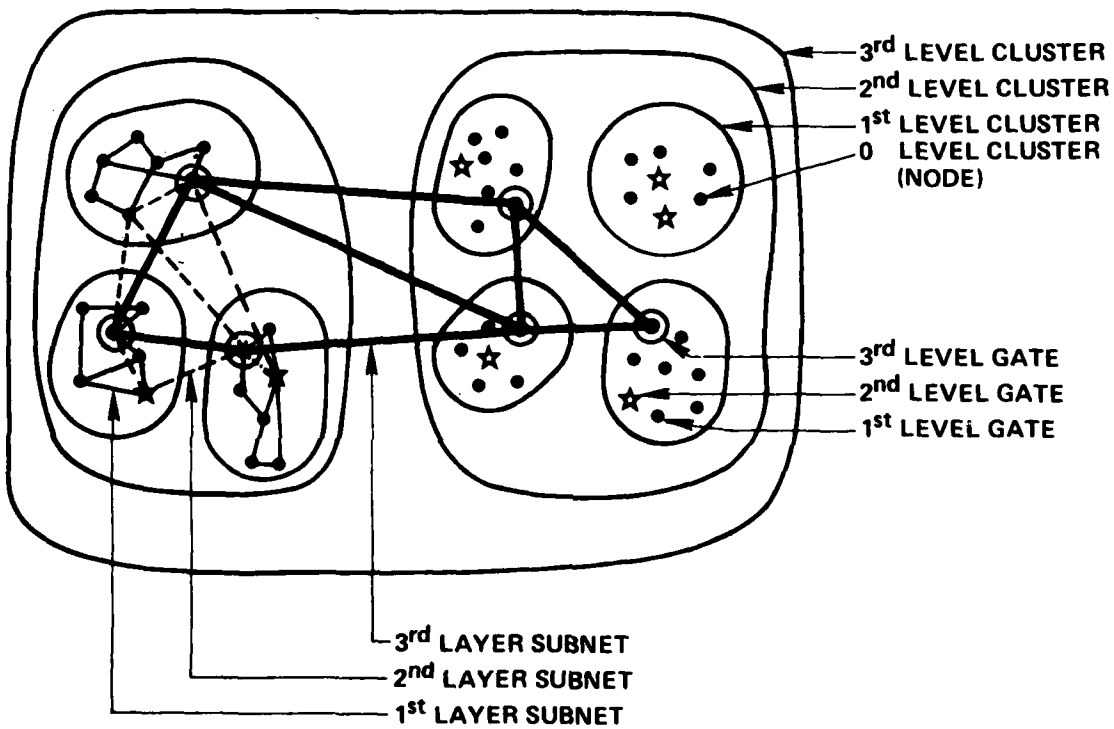


Figure 6.1. A 3-Level Hierarchical Network.

The selection of gates at all levels is governed by the following assumption.

Gate Assumption 6.2

Given an integer vector $\underset{\sim}{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)$ where $\beta_1 \triangleq 1$ and a selection rule, then, starting at k = 1, select β_{k+1} $k+1^{st}$ level gates among the set of kth level gates of each kth level cluster $C_k(i_m, \ldots, i_{k+1})$. Repeat this step sequentially until k = m - 1. A network node is considered to be a 1st level gate.

Let e_k be a kth level gate, and $E_k(i_m, i_{m-1}, \ldots, i_{k+1}) \triangleq \{e_k \,/\, e_k \in C_k(i_m, i_{m-1}, \ldots, i_k)\}$ be the set of all k^{th} level gates in the corresponding cluster. Also, let $g_k(i_m, i_{m-1}, \ldots, i_{k+1})$ be the size of the abovementioned set of gates. As a consequence of the gate assumption, the following relation exists between the size and the degree vectors.

$$g_k(i_m, i_{m-1}, \ldots, i_{k+1}) = \beta_k \, n_k(i_m, \ldots, i_{k+1}) \qquad (6.5)$$

Notice that for k = 1, the set of 1st level gates of a 1st level cluster is simply the set of all the network nodes contained in that cluster, hence

$$g_1(i_m, i_{m-1}, \ldots, i_2) = n_1(i_m, i_{m-1}, \ldots, i_2)$$

which explains the convention, $\beta_1 = 1$. For the above selection scheme to be feasible, the number of kth level gates of a kth level cluster, C_k, must be no less than β_{k+1}, i.e.,

$$g_k(i_m, i_{m-1}, \ldots, i_{k+1}) \geq \beta_{k+1} \qquad \forall\, k = 1, \ldots, m-1$$

The above relation, combined with Eq. (6.5), will henceforth introduce a constraint, to be referred to as "gate constraint," over the choice of the degree vector $\underset{\sim}{n}$ characterizing the MHC structure.

Gate Constraint

$$n_k(i_m, i_{m-1}, \ldots, i_{k+1}) \geq \frac{\beta_{k+1}}{\beta_k} \qquad \forall\, k = 1, 2, \ldots, m-1$$
$$\forall\, (i_m, i_{m-1}, \ldots, i_{k+1}) \quad (6.6)$$

The choice of the vector $\underset{\sim}{\beta}$ will be mainly related to the reliability constraint, Eq. (6.4). If a K-connectivity is to be imposed on the topology of the network, then the vector $\underset{\sim}{\beta}$ must be such that $\beta_i \geq K$ for $i = 2, 3, \ldots, m$. This is obvious since the set of the β_i i^{th} level gates of an $i\text{-}1^{st}$ level cluster represents a cut set [HARA 72] for the other nodes in that cluster.

Let a centralized hierarchical network be defined as a hierarchical network whose topology is such that all k^{th} level clusters (for all k's) communicate with the outside through only one $k+1^{st}$ level gate. Consequently, the corresponding vector $\underset{\sim}{\beta}$ is

$$\underset{\sim}{\beta} = (1, 1, \ldots. 1) \qquad \text{i.e., } \beta_i = 1 \qquad \forall\, i = 1, \ldots, m$$

This structure is commonly found in the hierarchical star networks and also in the organization of many business or government agencies.

6.3.2 Decomposition Step

The main computational gains introduced by the MHT come from the separate design of the different components of the networks, i.e., the design of the layer subnets at all levels. To perform this operation, it is necessary to define performance measures for those subnets as well as the corresponding constraints to be imposed on their design. Then, the global objective function and constraints must be expressed in terms of those associated with the layer subnets. As an example, assuming that messages travelling over k^{th} layer subnets will incur the same delay, T_k, then the problem is to express the global average delay, T, in terms of T_k's, i.e., an expression of the form

$$T = \sum_{i=1}^{m} a_i T_i$$

must be derived. A particular case is solved in Section 6.7 when an optimal MHC structure is considered. Similarly, given a certain dollar budget, the question may arise as to how much one should allocate for the design of each of the layer subnets at any level. Section 6.7 will address some of the above questions.

6.3.3 Design and Evaluation Steps

These two steps will not be studied in depth in this report. However, the economies of hierarchical networks, as well as their operational flexibility (routing, flow control, addition of nodes), will be pointed out in Chapter 7. Since the MHT results in the design of a set of much smaller subproblems, then present heuristic design

methods [GERL 73A], [FRAN 72] become feasible and quite attractive to perform the required job.

In order to evaluate the computational complexity in the design phase of the MHT, we will make the following assumption:

Computational Cost Assumption 6.3

a. The computational cost incurred in the design of a kth layer subnet connecting a set of n k^{th} level gates is equal to n^{α_k} $(k = 1, 2, \ldots, m)$.

b. The total computational cost involved in the design is equal to the sum of the costs induced in the design of all the layer subnets.

Recall that the polynomial form of the computational cost is the one normally used [FRAN 72] to characterize the computational complexity of most of the present design algorithms. The fact that different exponents, α_k's, could be selected, depending on the level of the hierarchy, is provided to allow the modelling of the design of hierarchical networks where different technologies or design algorithms or both are considered at each level or group of levels.

In summary, the main elements involved in the computational complexity of the MHT have been exposed under a set of assumptions which, hopefully, retain the essential character of the class of hierarchical design procedures.

6.4 Optimal Clustering Structure, General Case

Given the number of levels, m, the objective here is to determine the optimal clustering structure, i.e., the degree vector $\underset{\sim}{n}$,

which will result in a minimum computational cost. A similar problem has been solved in Chapter 2 but with a different objective function, i.e., the length of the routing table (see Eq. 2.3)).

6.4.1 Expression of the Computational Cost and Problem Statement

The model aimed at capturing the computational complexity of the MHT has been introduced and discussed in the previous paragraph.

Let $G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta})$ represent the computational cost incurred in the design step of the MHT. It is a function of the underlying clustering structure, the number of gates and the computational cost of the design of the different layer subnets which are respectively characterized by $(m, \underset{\sim}{n})$, $\underset{\sim}{\beta}$ and $\underset{\sim}{\alpha}$. Some of these variables will be dropped from the above notation when no confusion is possible. Notation such as G, $G(m, \underset{\sim}{n})$, etc., will be encountered in the rest of the chapter. Similarly, $G_k(m, \underset{\sim}{n}, \alpha, \beta)$ will denote the cost incurred in the design of all k^{th} layer subnets. We are now ready to evaluate G; we start with the simple case of a 3-level clustering.

6.4.1.1 Expression of G for a 3-Level Hierarchy

Fig. 6.1 shows a 3-level hierarchical clustering of a set of nodes. From Assumptions 6.2 and 6.3 and Eq. (6.5), the computational cost incurred in the design of all the 1st layer subnets is

$$G_1(3, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = \sum_{i_3=1}^{n_3} \sum_{i_2=1}^{n_2(i_3)} [\beta_1 n_1(i_3, i_2)]^{\alpha_1}$$

where the summations extend over all 1st level clusters of the network. (Recall that $\beta_1 = 1$). Similarly

$$G_2(3, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = \sum_{i_3=1}^{n_3} [\beta_2 n_2(i_3)]^{\alpha_2}$$

and

$$G_3(3, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = [\beta_3 n_3]^{\alpha_3}$$

Consequently, the global computational cost is

$$G(3, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = \sum_{i_3=1}^{n_3} \sum_{i_2=1}^{n_2(i_3)} [\beta_1 n_1(i_3, i_2)]^{\alpha_1} + \sum_{i_3=1}^{n_3} [\beta_2 n_2(i_3)]^{\alpha_2} + [\beta_3 n_3]^{\alpha_3} \qquad (6.7)$$

6.4.1.2 <u>Expression of G for an m-Level Hierarchy</u>

From Assumption 6.3 and Eq. (6.5), the computational cost of the design of a kth layer subnet is equal to

$$[g_k(i_m, i_{m-1}, \ldots, i_{k+1})]^{\alpha_k} = [\beta_k n_k(i_m, i_{m-1}, \ldots, i_{k+1})]^{\alpha_k}$$

The above expression, summed over all the kth level clusters, represents the computational cost incurred in the design of all the kth layer subnets, i.e.,

$$G_k(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = \sum_{i_m=1}^{n_m} \sum_{i_{m-1}=1}^{n_{m-1}(i_m)} \cdots \sum_{i_{k+1}=1}^{n_{k+1}(i_m, i_{m-1}, \ldots, i_{k+2})} [\beta_k n_k(i_m, i_{m-1}, \ldots, i_{k+1})]^{\alpha_k}$$

$$G_m(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = (\beta_m n_m)^{\alpha_m} \qquad (6.8)$$

Notice that for k = m, by convention, we will set the multiple summation to one.

From Assumption 6.3, the global computational cost is:

$$G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = \sum_{k=1}^{m} G_k(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) \tag{6.9}$$

6.4.1.3 <u>Problem Statement</u>

$$\begin{cases} \text{given:} & N, \underset{\sim}{\alpha}, \underset{\sim}{\beta} \\ \text{minimize:} & G(m, n, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) \quad \text{[see Eq. (6.9)]} \\ \text{over:} & m \text{ and } \underset{\sim}{n} \\ \text{subject to:} & \text{size constraint} \quad \text{[see Eq. (2.1)]} \\ & \text{gate constraint} \quad \text{[see Eq. (6.6)]} \\ & m \text{ positive integer variable} \\ & \underset{\sim}{n} \text{ vector of positive integer variables} \end{cases} \tag{6.10}$$

The precise problem expressed above is a non-linear integer programming problem which has not been solved in its entirety. In order to make progress toward a solution, we choose to relax some of the constraints, namely, the integer and gate constraints. It is also necessary to temporarily freeze the variable m. The relaxation of the gate constraint is not of too much consequence, since it will be shown that for practical values of $\underset{\sim}{\beta}$, $\underset{\sim}{\alpha}$, N, m, the optimal solution will satisfy that constraint. The added relaxation of the integer constraint will lead to a nice analytic solution when m is given; (for some particular properties of the vectors, $\underset{\sim}{\alpha}$ and $\underset{\sim}{\beta}$, an analytical solution is found for the optimal m.) The resulting real-valued solution is of considerable importance for at least two reasons: (1) The study of its behavior,

with respect to the variables m, $\underset{\sim}{\alpha}$, $\underset{\sim}{\beta}$, will provide us with insight as to how much computational gains can be obtained through the application of the MHT design procedure and into the choice of the appropriate clustering structure. (2) A sub-optimal integer solution could be directly obtained from the real-valued solution (thereby providing an upper bound on the cost). As a consequence of the relaxation of the integer constraint, one question arises as to what is the meaning of a discrete summation where the upper variable is not an integer. Consider as an example

$$G_2(3, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = \sum_{i_3=1}^{n_3} [\beta_2 n_2(i_3)]^{\alpha_2}$$

The summation is only meaningful in two instances:

(1) when n_3 is integer

(2) when $n_2(i_3) = n_2 \quad \forall \quad i_3$

Then the summation becomes:

$$\sum_{i_3=1}^{n_3} [\beta_2 n_2(i_3)]^{\alpha_2} \triangleq n_3[\beta_2 n_2]^{\alpha_2}$$

The solution of the optimization problem will show that clusters at the same level must be of equal degree (size); hence, the 2nd condition will always be satisfied a posteriori.

The optimization problem to be solved is reduced to:

$$\begin{cases} \text{given:} & N, \underset{\sim}{\alpha}, \underset{\sim}{\beta} \\ \text{minimize:} & G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) \\ \text{over:} & \underset{\sim}{n}, m \\ \text{s.t.:} & \text{size constraint} \end{cases}$$

$$\underset{\sim}{n} \geq 0 \qquad \text{real-valued vector}$$

$$m > 0 \qquad \text{integer} \qquad (6.11)$$

6.4.2 Optimal Solution

Proposition 6.1

Given m, the number of levels in the hierarchy, and assuming that $\alpha_i > 1$ for all $i = 1, \ldots, m$, then the solution of the optimization problem 6.11, i.e., the optimal clustering structure, is such that:

(a) All clusters at the same level, $k = 1, \ldots, m$, are composed of an equal number of lower level clusters; i.e., all nodes at the same level in the tree representation are of equal degree. The optimal degree vector reduces to an m-dimensional vector, $\underset{\sim}{n} = (n_1, n_2, \ldots, n_m)$ whose components are the solution of the following set of difference equations:

$$n_k(i_m, i_{m-1}, \ldots, i_{k+1}) = n_k \quad \forall \quad k = 1, \ldots, m \text{ and } (i_m, \ldots, i_{k+1})$$

$$\begin{cases} n_1 = \dfrac{N}{n_2 n_3 \cdots n_m} \\[2ex] n_k = \left[\dfrac{\prod_{i=1}^{k-1} (\alpha_i - 1)}{\alpha_k (\beta_k)^{\alpha_k}} \; \dfrac{B_k}{D_k} \left[\prod_{i=k+1}^{m} n_i \right]^{-\frac{\prod_{i=1}^{k-1} \alpha_i}{D_k}} \right]^{\frac{D_k}{D_{k+1}}} \end{cases} \qquad (6.12)$$

$$k = 2, 3, \ldots, m$$

where, by convention $\prod_{i=m+1}^{m} n_i \triangleq 1$; and D_k is the solution of:

$$\begin{cases} D_2 = 1 \\[2ex] D_k = \alpha_{k-1} D_{k-1} + \prod_{i=1}^{k-2} (\alpha_i - 1) \qquad k \geq 3 \end{cases} \qquad (6.13)$$

Also, B_k is the solution of

$$\begin{cases} B_2 = N^{\alpha_1} \\ \left[\dfrac{B_k}{D_k}\right]^{D_k} = \alpha_{k-1}^{-\alpha_{k-1}D_{k-1}} \left[\dfrac{\prod_{i=1}^{k-2}(\alpha_i - 1)}{(\beta_{k-1})^{\alpha_{k-1}}}\right]^{-\prod_{i=1}^{k-2}(\alpha_i - 1)} \left[\dfrac{B_{k-1}}{D_{k-1}}\right]^{\alpha_{k-1}D_{k-1}} \end{cases} \tag{6.14}$$

$$k \geq 3$$

(b) With this optimum solution, the minimum computational cost is:

$$\overline{G}(m, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = B_{m+1} \tag{6.15}$$

The proof of Proposition 6.1 is given in Appendix C. It consists of showing when all other variables are fixed, that the $n_1(\cdot)$'s must all be equal (to n_1). Then, n_1 is replaced by its optimal expression in the objective function, and the same operation is repeated for the degrees at the next level, and so forth, until all levels are exhausted.

As a consequence of Proposition 6.1, the optimal size vector also reduces to an m-dimensional vector, $\underset{\sim}{g} = (g_1, g_2, \ldots, g_m)$, whose components are given by

$$g_k = \beta_k n_k \qquad k = 1, \ldots, m \tag{6.16}$$

where n_k is the solution of Eq. (6.12).

The computation of the variables D_k, B_k, n_k may proceed sequentially by computing the sequences below.

(1) $D_3, D_4, \ldots, D_{m+1}$

(2) $\frac{B_3}{D_3}, \frac{B_4}{D_4}, \ldots, \frac{B_{m+1}}{D_{m+1}}$

(3) $n_m, n_{m-1}, \ldots, n_1$

where

$$n_m = \left[\frac{\prod_{i=1}^{m-1} (\alpha_i - 1)}{\alpha_m (\beta_m)^{\alpha_m}} \frac{B_m}{D_m} \right]^{\frac{D_m}{D_{m+1}}} \tag{6.17}$$

n_m is the first element to compute in the sequence $\{n_k\}$.

Notice that one can also solve the difference equations, (6.13) and (6.14) to obtain explicit expressions for D_k and B_k/D_k.

Explicit Expression of D_k

$$D_k = \sum_{j=2}^{k} \left(\prod_{i=j}^{k-1} \alpha_i \right) \left(\prod_{i=1}^{j-2} (\alpha_i - 1) \right) \qquad \forall\, k \geq 1 \tag{6.18}$$

By convention, when the lower index in the product exceeds the upper index, then the product is set to be equal to one. If the same situation arises with a summation sign, then the sum is set to be equal to zero. This convention will apply throughout the chapter.

Explicit Expression for B_k/D_k

$$\left[\frac{B_k}{D_k} \right]^{D_k} = \prod_{j=2}^{k-1} \left[\alpha_j^{-\alpha_j D_j} \left[\frac{\prod_{i=1}^{j-1} (\alpha_i - 1)}{(\beta_j)^{\alpha_j}} \right]^{-\prod_{i=1}^{j-1} (\alpha_i - 1)} \right]^{\prod_{i=j+1}^{k-1} \alpha_i} (N)^{\prod_{i=1}^{k-1} \alpha_i}$$

$$\forall\, k \geq 2 \tag{6.19}$$

The proof follows immediately from Eq. (6.14).

The combination of Eqs. (6.18) and (6.19), at $k = m + 1$, will provide an explicit expression for the minimum computational cost, $\overline{G}(m, \underset{\sim}{\alpha}, \underset{\sim}{\beta})$. The next natural step to take is to perform the <u>optimization of $\overline{G}$</u> with respect to m. This may easily be done numerically when m is restricted to a certain range of integer values.

<u>Numerical Examples</u>

Define $\underset{\sim}{G} \triangleq (G_1, G_2, G_3, G_4)$. Recall that G_k is the computational cost for the design of the kth layer subnets.

$\underset{\sim}{\alpha} = (2, 2.5, 3, 3.5)$

(i) $\underset{\sim}{\beta} = (1, 1, 1, 1)$

$m = 1$ $\quad \underset{\sim}{n} = (10^3)$ $\quad$ i.e., $\underline{N = 10^3}$

$G = 10^6$ $\quad (G \triangleq N^{\alpha_1}$ for $m = 1)$

$m = 2$ $\quad \underset{\sim}{n} = (25.08 \quad , 39.86 \quad)$ Recall that n is assumed to be continuous.

$\underset{\sim}{G} = (2.51\ 10^4, 1.00\ 10^4)$

$G = 3.51\ 10^4$

$m = 3$ $\quad \underset{\sim}{n} = (8.65 \quad , 9.63 \quad , 12.00 \quad)$

$\underset{\sim}{G} = (8.646\ 10^3, 3.453\ 10^3, 1.728\ 10^3)$

$G = 1.38\ 10^4$

$m = 4$ $\quad \underset{\sim}{n} = (5.46 \ , 5.22 \quad , 5.58 \quad , 6.29 \quad)$

$\underset{\sim}{G} = (54.61\ 10^2, 21.85\ 10^2, 10.92\ 10^2, 6.24\ 10^2)$

$G = 9.36\ 10^3$

(ii) $\underset{\sim}{\beta} = (1, 2, 2, 2)$

$m = 1 \qquad \underset{\sim}{n}\ (10^3)$

$G = 10^6$

$m = 2 \qquad \underset{\sim}{n} = (41.16 \quad , 24.30 \quad)$

$\underset{\sim}{g} = (41.16 \quad , 48.60 \quad)$

$\underset{\sim}{G} = (4.117\ 10^4, 1.647\ 10^4)$

$G = 5.76\ 10^4$

$m = 3 \qquad \underset{\sim}{n} = (17.29, 7.65 \quad , 7.56 \quad)$

$\underset{\sim}{g} = (17.29, 15.30 \quad , 15.12 \quad)$

$\underset{\sim}{G} = (1.73\ 10^4, .69\ 10^4, .34\ 10^4)$

$G = 2.77\ 10^4$

$m = 4 \qquad \underset{\sim}{n} = (12.26 \quad , 4.83 \quad , 4.26 \quad , 3.96 \quad)$

$\underset{\sim}{g} = (12.26 \quad , 9.66 \quad , 8.52 \quad , 7.92 \quad)$

$\underset{\sim}{G} = (1.22\ 10^4, .49\ 10^4, .25\ 10^4, .14\ 10^4)$

$G = 2.10\ 10^4$

6.5 Optimal Clustering Structure with Uniform Design Strategy and Gate Assignment

This section deals with the special case where the same design procedure is used at all levels of the hierarchy. Also, an equal number of gates is selected from clusters at all levels, i.e.,

$$\alpha_k = \alpha \qquad \forall\, k \geqq 1, \qquad \alpha > 1$$
$$\beta_k = \beta \qquad \forall\, k \geqq 2, \qquad \beta_1 \triangleq 1 \tag{6.20}$$

As a result, much simpler expressions are obtained for $\underset{\sim}{n}$ and G at optimality, given m. Furthermore, it is possible, analytically, to find the optimum number of levels in the hierarchy. At global optimality, all the layer subnets must be of equal size. This surprising result has an intuitive explanation given below.

In what follows, first, the expressions for $\underset{\sim}{n}$ and G will be derived at optimality, given m. Then, the global optimum solution will be determined and its peculiar properties will be studied, as well as its behavior with respect to α and β. Finally, comparisons with intuitive feasible solutions will be made.

6.5.1 Optimal Expressions Given m

Corollary 6.1

Given m, the number of levels in the hierarchy, and assuming[1] that $\alpha_i = \alpha$ for $i = 1, 2, \ldots, m$; $\alpha > 1$; and $\beta_i = \beta$ for $i = 2, 3, \ldots, m$; then, the optimal solution of Problem 6.11 is

$$\begin{cases} n_1 = \beta \dfrac{\alpha}{\alpha - 1} \left[\left(\dfrac{\alpha - 1}{\alpha} \right)^m \dfrac{N}{\beta} \right]^{\frac{(\alpha-1)^{m-1}}{D_{m+1}}} \\[2ex] n_k = \dfrac{\alpha}{\alpha - 1} \left[\left(\dfrac{\alpha - 1}{\alpha} \right)^m \dfrac{N}{\beta} \right]^{\frac{(\alpha-1)^{m-k}\alpha^{k-1}}{D_{m+1}}} \qquad k = 2, \ldots, m \end{cases} \tag{6.21}$$

[1] $\alpha \leq 1$ is treated in Section 6.5.4.4.

With this solution, the minimum computational cost is:

$$\overline{G}(m, \alpha, \beta) = D_{m+1} \left[\left(\beta \frac{\alpha}{\alpha - 1} \right)^{\alpha(\alpha-1)D_m} \left(\frac{(\alpha - 1)^{(\alpha-1)^m}}{\alpha^{\alpha^m}} \right)^{m-1} N^{\alpha^m} \right]^{\frac{1}{D_{m+1}}} \tag{6.22}$$

where $$D_k = \alpha^{k-1} - (\alpha - 1)^{k-1} \quad \text{for} \quad k \geq 1 \tag{6.23}$$

Proof:[1]

To prove Corollary 6.1, we need to find the expression for D_k and B_k/D_k.

Expression for D_k

From Eqs. (6.18) and (6.20)

$$D_k = \sum_{j=2}^{k} \alpha^{k-j} (\alpha - 1)^{j-2} \tag{6.24}$$

Now $$D_k = (\alpha - 1)^{k-2} \sum_{i=0}^{k-2} \left(\frac{\alpha}{\alpha - 1} \right)^i$$

After summing on i, we find Eq. (6.23).

Expression for B_k/D_k

From Eq.s (6.19) and (6.20)

$$\left[\frac{B_k}{D_k} \right]^{D_k} = \alpha^{-\alpha \sum_{i=0}^{k-3} \alpha^i D_{k-i-1}} (\alpha - 1)^{-\sum_{i=1}^{k-2} i\alpha^{k-i-2}(\alpha-1)^i} \beta^{\sum_{i=2}^{k-1} \alpha^{k-i}(\alpha-1)^{i-1}} N^{\alpha^{k-1}}$$

To evaluate the exponents involved in the above expression, we need the following identity.

[1]The same result as in Eq. (6.21) has been derived in [FRAN 72] for m = 2.

$$\sum_{i=1}^{n} ix^i = \frac{x(nx^{n+1} - (n+1)x^n + 1)}{(x-1)^2} \quad \text{for } x \neq 1 \qquad (6.25)$$

From Eqs. (6.23), (6.24) and (6.25):

$$\alpha \sum_{i=0}^{k-3} \alpha^i D_{k-i-1} = \sum_{i=0}^{k-3} [\alpha^{k-1} - \alpha^{i+1}(\alpha - 1)^{k-i-2}]$$

$$= (k-2)\alpha^{k-1} - \alpha(\alpha - 1)D_{k-1} .$$

Also $$\sum_{i=1}^{k-2} i\alpha^{k-i-2}(\alpha - 1)^i = \alpha^{k-2} \sum_{i=1}^{k-2} i\left(\frac{\alpha - 1}{\alpha}\right)^i$$

$$= (\alpha - 1)\left(\alpha D_{k-1} - (k-2)(\alpha - 1)^{k-2}\right)$$

Finally, the exponent of β is equal to $\alpha(\alpha - 1)D_{k-1}$. After substituting these exponents into the above expression of B_k/D_k and grouping terms together, we get:

$$\left[\frac{B_k}{D_k}\right]^{D_k} = \left[\beta \frac{\alpha}{\alpha - 1}\right]^{\alpha(\alpha-1)D_{k-1}} \left[\frac{(\alpha - 1)^{(\alpha-1)^{k-1}}}{\alpha^{\alpha^{k-1}}}\right]^{k-2} N^{\alpha^{k-1}} \qquad (6.26)$$

From the above equation at $k = m + 1$ and Eq. (6.15) we obtain Eq. (6.22).

The proof of Eq. (6.21) will proceed by induction. First, let us observe that from Eqs. (6.12), (6.20) and (6.26)

$$n_k^{D_{k+1}} = \left[\frac{(\alpha-1)^{k-1}}{\alpha\beta^\alpha}\right]^{D_k} \left[\beta \frac{\alpha}{\alpha-1}\right]^{\alpha(\alpha-1)D_{k-1}} \left[\frac{(\alpha-1)^{(\alpha-1)^{k-1}}}{\alpha^{\alpha^{k-1}}}\right]^{k-2} \left[\frac{N}{\prod_{i=k+1}^{m} n_i}\right]^{\alpha^{k-1}}$$

Factoring out the terms in α, $\alpha - 1$, β and using Eq. (6.23) to evaluate their exponents (to be denoted by exp ( )), we find

$$\begin{aligned}
\exp(\beta) &= \alpha(\alpha-1)D_{k-1} - \alpha D_k &&= -\alpha^{k-1} \\
\exp(\alpha) &= \alpha(\alpha-1)D_{k-1} - D_k - (k-2)\alpha^{k-1} &&= -k\alpha^{k-1} + D_{k+1} \\
\exp(\alpha-1) &= (k-1)D_k - \alpha(\alpha-1)D_{k-1} + (k-2)(\alpha-1)^{k-1} &&= k\alpha^{k-1} - D_{k+1}
\end{aligned}$$

After substitution, we get

$$n_k = \frac{\alpha}{\alpha - 1}\left[\left[\frac{\alpha - 1}{\alpha}\right]^k \frac{N}{\beta}\right]^{\frac{\alpha^{k-1}}{D_{k+1}}} \left[\prod_{i=k+1}^{m} n_i\right]^{-\frac{\alpha^{k-1}}{D_{k+1}}} \qquad k \geq 2 \tag{6.27}$$

For $k = m$, the product term in Eq. (6.27) vanishes to one, and the resulting expression of n_m satisfies Eq. (6.21). Assuming that Eq. (6.21) is true for $i = m, m - 1, \ldots, k + 1$, let us show that it is true for $i = k$.

From the induction hypothesis and Eq. (6.24),

$$\prod_{i=k+1}^{m} n_i = \left(\frac{\alpha}{\alpha - 1}\right)^{m-k}\left[\left(\frac{\alpha - 1}{\alpha}\right)^m \frac{N}{\beta}\right]^{\frac{\alpha^k D_{m-k+1}}{D_{m+1}}} \tag{6.28}$$

Substituting Eq. (6.28) into Eq. (6.27), and using Eq. (6.23) to simplify the resulting exponents, we arrive at the expression of n_k as given in Eq. (6.21). Since Eq. (6.27) is only true for $k = 2, 3, \ldots, m$ then it remains to be proven that n_1 satisfies Eq. (6.21). This is obvious from the expression of n_1 in Eq. (6.12) and from Eq. (6.28) for $k = 1$.

Remarks

(i) Size vector: From Eqs. (6.16) and (6.20)

$$g_1 = n_1, \quad g_k = \beta n_k \qquad k = 2, 3, \ldots, m$$

Substituting Eq. (6.21) into the above expression, we get a unique expression for g_k, valid for $k = 1, 2, \ldots, m$.

(ii) Gate Constraint: To satisfy the gate constraint, Eq. (6.6), the vector $\underset{\sim}{n}$ must be such that

$$n_1 \geq \beta, \quad n_k \geq 1 \qquad k = 2, \ldots, m$$

This condition will always be satisfied if

$$\left(\frac{\alpha - 1}{\alpha}\right)^m \frac{N}{\beta} \geq 1 \qquad \text{i.e.,} \qquad m \leq \frac{\ln \frac{N}{\beta}}{\ln \frac{\alpha}{\alpha-1}}$$

In the next section, we will show that the region of interest for m will effectively correspond to the above condition.

6.5.2 Global Optimum

So far, we have solved for the optimal clustering structure when m, the number of levels, is fixed. We now intend to let m vary and, consequently, to solve for the global optimum.

Proposition 6.2

Under the condition of Eq. (6.20) and m being a real variable, the global optimum[1] clustering structure is achieved for a number of levels

[1] "Global optimum" is used to distinguish the optimum solution with respect to m. Also, * indicates values at global optimality.

$$m_* = \frac{\ln \frac{N}{\beta}}{\ln \frac{\alpha}{\alpha - 1}} \tag{6.29}$$

and a degree vector n*

$$\begin{cases} n_1^* = \beta \dfrac{\alpha}{\alpha - 1} \\ n_k^* = \dfrac{\alpha}{\alpha - 1} \qquad k = 2, 3, \ldots, m_* \end{cases} \tag{6.30}$$

The corresponding minimum computational cost is

$$G_*(\alpha, \beta) = \left(\frac{N}{\beta} - 1\right)\beta^{\alpha} \frac{\alpha^{\alpha}}{(\alpha - 1)^{\alpha-1}} \tag{6.31}$$

<u>Proof</u>:

Since Problem 6.11 has been solved when m is fixed, there only remains to minimize $\overline{G}(m, \alpha, \beta)$ over m. Eq. (6.22) may be rewritten as

$$\overline{G} = D_{m+1}[H(m)]^{\frac{1}{D_{m+1}}}$$

where H(m) represents the expression between the outside brackets. Differentiating $\overline{G}$ with respect to m,

$$\overline{G}' \triangleq \frac{d\overline{G}}{dm} = [H(m)]^{\frac{1}{D_{m+1}}} \left[D'_{m+1} + D_{m+1} \left[\frac{\ln H(m)}{D_{m+1}} \right]' \right]$$

$$D_{m+1} \left[\frac{\ln H(m)}{D_{m+1}} \right]' = \frac{D_{m+1}[\ln H(m)]' - [\ln H(m)]D'_{m+1}}{D_{m+1}}$$

which we define as $\frac{U}{D_{m+1}}$. From Eq. (6.22)

$$\ln H(m) = \alpha(\alpha - 1)D_m \ln \beta\frac{\alpha}{\alpha - 1} - (m - 1)\alpha^m \ln \alpha$$

$$+ (m - 1)(\alpha - 1)^{m-1} \ln (\alpha - 1) + \alpha^m \ln N$$

Differentiating ln H(m) with respect to m, and substituting into U, we arrive at

$$U = \alpha(\alpha - 1)\left(\ln \beta \frac{\alpha}{\alpha - 1}\right) [D_{m+1}D'_m - D_m D'_{m+1}]$$

$$- (m - 1)\alpha^m(\ln \alpha)[D_{m+1} \ln \alpha - D'_{m+1}]$$

$$+ (m - 1)(\alpha - 1)^m(\ln (\alpha - 1))[D_{m+1} \ln (\alpha - 1) - D'_{m+1}]$$

$$+ \alpha^m(\ln N)[D_{m+1} \ln \alpha - D'_{m+1}]$$

$$+ D_{m+1}[(\alpha - 1)^m \ln (\alpha - 1) - \alpha^m \ln \alpha]$$

Also, from Eq. (6.23)

$$D'_m = \alpha^{m-1} \ln \alpha - (\alpha - 1)^{m-1} \ln(\alpha - 1)$$

$$D'_{m+1} = \alpha^m \ln \alpha - (\alpha - 1)^m \ln (\alpha - 1) \qquad (6.32)$$

Evaluating the terms between brackets in U, we obtain

$$D_{m+1}D'_m - D_m D'_{m+1} = \alpha^{m-1}(\alpha - 1)^{m-1} \ln \frac{\alpha}{\alpha - 1}$$

$$D_{m+1} \ln \alpha - D'_{m+1} = -(\alpha - 1)^m \ln \frac{\alpha}{\alpha - 1}$$

$$D_{m+1} \ln (\alpha - 1) - D'_{m+1} = -\alpha^m \ln \frac{\alpha}{\alpha - 1}$$

Replacing those expressions in $U/D_{m+1} + D'_{m+1}$, we notice that the last term of U will cancel with D'_{m+1}, and we are left with

$$\frac{dG}{dm} = [H(m)]^{\frac{1}{D_{m+1}}} \alpha^m(\alpha - 1)^m \left(\ln \frac{\alpha}{\alpha - 1}\right) \ln \frac{\beta}{N}\left(\frac{\alpha}{\alpha - 1}\right)^m \qquad (6.33)$$

Since $\alpha > 1$, $\overline{G}'$ is of the same sign and has the same roots as

$$\ln \frac{\beta}{N}\left(\frac{\alpha}{\alpha - 1}\right)^m$$

for all values of m. Equating the above equation to zero, we find m_* as given in Eq. (6.29). Also, for $m < m_*$, $\overline{G}'$ is negative, and for $m > m_*$, it is positive. Therefore, we conclude that G is minimum for $m = m_*$, and m_* is the optimal number of levels.

In order to find the corresponding optimal vector $\underset{\sim}{n}$, let us notice that from Eq. (6.29)

$$\left(\frac{\alpha - 1}{\alpha}\right)^{m_*} \frac{N}{\beta} = 1 \tag{6.34}$$

Substituting Eq. (6.34) into Eq. (6.21), we arrive at Eq. (6.30).

There are two possible ways to evaluate the global minimum cost G_*.

(i) <u>Replace m by m_*</u> in Eq. (6.22). After some algebra, using Eq. (6.23), Eq. (6.22) may be rewritten as

$$\overline{G} = \frac{\alpha^\alpha}{(\alpha - 1)^{\alpha-1}} D_{m+1}\left(\beta^{\alpha(\alpha-1)D_{m_\alpha - m\alpha^m}}(\alpha - 1)^{m(\alpha-1)^m} N^{\alpha^m}\right)^{\frac{1}{D_{m+1}}} \tag{6.35}$$

Then from Eqs. (6.23) and (6.34)

$$D_{m_*+1} = (\alpha - 1)^{m_*}\left(\frac{N}{\beta} - 1\right)$$

and

$$N^{\alpha^{m_*}} = \beta^{\alpha^{m_*}}\left(\frac{\alpha - 1}{\alpha}\right)^{m_*\alpha^{m_*}}$$

Substituting the last two equations into Eq. (6.35) at $m = m_*$, we get Eq. (6.31), after simplification.

(ii) When all the kth level clusters are of equal degree, the computational cost, Eq. (6.9), becomes

$$G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = \sum_{k=1}^{m} \left(\prod_{i=k+1}^{m} n_i \right) (\beta_k n_k)^{\alpha_k} \tag{6.36}$$

Substituting Eq. (6.30) into Eq. (6.36), we find

$$G_* = (n_1^*)^{\alpha} \frac{(n_1^*/\beta)^{m_*} - 1}{(n_1^*/\beta) - 1} \tag{6.37}$$

Replacing n_1^* in Eq. (6.37) by $\beta \frac{\alpha}{\alpha - 1}$, we get Eq. (6.31). This last operation terminates the proof of Proposition 6.2.

Remarks

1. Size vector: At global optimality, the size vector is such that

$$g_k^* = \beta \frac{\alpha}{\alpha - 1} \qquad k = 1, 2, \ldots, m_* \tag{6.38}$$

Eq. (6.38) indicates that at optimality, all the layer subnets are of equal size which depends only on α and β. The explanation of this very simple and interesting property will be the object of Section 6.5.3.

2. Gate constraint: From Eq. (6.38), since $\alpha > 1$,

$$g_k^* \geq \beta \qquad k = 1, 2, \ldots, m$$

Hence, the gate constraint, Eq. (6.6), is always satisfied at optimality. Moreover, as noticed at the end of Section 6.5.1, the optimal solution, given $m \leq m_*$, will also satisfy the gate constraint.

3. Number of layer subnets: Eq. (6.31) may be rewritten as

$$G_* = \left(\frac{N}{\beta} - 1\right)(\alpha - 1)\left(\beta \frac{\alpha}{\alpha - 1}\right)^{\alpha} \tag{6.39}$$

From Eq. (6.38), we conclude that the number of layer subnets is

$$NL_* = \left(\frac{N}{\beta} - 1\right)(\alpha - 1) \tag{6.40}$$

The above results could also be derived by counting the number of clusters in the tree structure (Fig. 2.4). Let NC_k be the number of kth level clusters; then, from Eq. (6.12)

$$NC_k = \prod_{i=k+1}^{m} n_i \qquad k = 1, 2, \ldots, m \tag{6.41}$$

At optimality, given m, NC_k is given by Eq. (6.28); and at global optimality

$$NC_k^* = \left(\frac{\alpha}{\alpha - 1}\right)^{m_* - k} \tag{6.42}$$

Hence, the total number of layer subnets is

$$NL_* = \sum_{k=1}^{m} NC_k^* = (\alpha - 1)\left(\left(\frac{\alpha}{\alpha - 1}\right)^{m_*} - 1\right)$$

The substitution of Eq. (6.34) into the above equation gives Eq. (6.40).

4. <u>Cost distribution</u>: The computational cost incurred in the design of all the kth layer subnets is

$$G_k^* = NC_k^*\left(\beta \frac{\alpha}{\alpha - 1}\right)^{\alpha} \qquad k = 1, \ldots, m_* \tag{6.43}$$

which is to be compared with the total cost G_*. For practical situations, $N/\beta >> 1$; hence, from Eq. (6.31)

$$G_* \simeq N\beta^{\alpha-1} \frac{\alpha^\alpha}{(\alpha - 1)^{\alpha-1}} = \alpha \frac{N}{n_1^*} \left(\beta \frac{\alpha}{\alpha - 1}\right)^\alpha$$

Since $NG_1^* = N/n_1^*$, then substituting Eq. (6.43) for k = 1 into the above equation, we get

$$G_* \simeq \alpha G_1^* \tag{6.44}$$

which says that the design of the 1st layer subnets represents approximately $1/\alpha$ of the total computational cost.

5. Behavior of the optimal tree structure with respect to α: We will assume that $\beta = 1$. Let us define a regular tree of degree K as a tree whose nodes are all of equal downward degree K, except for the leaves (downward degree zero), [KNUT 69]. As an example, a binary tree is a regular tree with degree 2.

The global optimum solution given in Eq. (6.30) becomes, for $\beta = 1$:

$$n_k^* = \frac{\alpha}{\alpha - 1} \qquad k = 1, 2, \ldots, m_* \tag{6.45}$$

We are interested in the set of α's which yield to integer solutions, i.e., to regular tree structures. If K is the degree of such trees, then α must be such that

$$\frac{\alpha}{\alpha - 1} = K \quad \Longleftrightarrow \quad \alpha = \frac{K}{K - 1} \tag{6.46}$$

Moreover, if a regular tree of degree K is composed of m_* levels, then it contains K^{m_*} leaves. Consequently, there is a one to one correspondence between the set of regular trees of degree K ($K \geq 2$ integer) whose number of levels is m_* (integer ≥ 2) and the global optimal solutions of Problem 6.11, where $\alpha = \frac{K}{K - 1}$ and $N = K^{m_*}$ for

$K = 2, 3, \ldots, \infty$. Notice that the above set of α's is contained in the interval $(1, 2]$ of real values, i.e., $1 < \alpha \leq 2$. Also, $\underline{\alpha = 2}$ corresponds to a <u>binary tree</u> representation.

6.5.3 Irreducibility

The simplicity of the solution, Eq. (6.38), obtained at global optimality leads us to consider a more intuitive approach toward its derivation, based on irreducibility considerations.

<u>Definition</u>: An <u>irreducible</u> set of nodes is such that no computational gain can be obtained through the application of the MHT for the design of the corresponding communication network.

<u>Lemma 6.1</u>: At global optimality, each layer subnet to be designed must correspond to an irreducible set of nodes.

<u>Proof</u> (by contradiction): Assume that at optimality, there exists at least one reducible unit. Then, the application of the MHT to design the network corresponding to that specific unit will reduce the computational cost for that unit and consequently for the total design. This contradicts the fact that we have reached the global optimum.

Notice that the application of the MHT to a reducible unit will preserve the hierarchical structure of the network as defined in Section 6.3. This would not be true if the α's or β's, or both, were not equal at all levels.

<u>Assumption</u>: Our intuition leads us to assume that for a given α and β, the size of an irreducible unit is unique.

Fact 6.1: At optimality, all layer subnets must be of equal size, q, and the number of levels must be equal to m_*,

$$q = \beta \frac{\alpha}{\alpha - 1} \qquad \text{and} \qquad m_* = \frac{\ln \frac{N}{\beta}}{\ln \frac{\alpha}{\alpha - 1}} \tag{6.47}$$

Proof: As a consequence of Lemma 6.1 and the above assumption, at optimality all the layer subnets are of equal size. Let q be that size; then from Eq. (6.5)

$$\begin{cases} n_1 = q \\ n_k = q/\beta \qquad k = 2, \ldots, m \end{cases} \tag{6.48}$$

From the size constraint, Eq. (2.1)

$$q = \beta \left(\frac{N}{\beta}\right)^{1/m} \tag{6.49}$$

Substituting Eqs. (6.48) and (6.49) into Eq. (6.9), we arrive at

$$G = \left(\frac{N}{\beta} - 1\right) \beta^{\alpha} \frac{(N/\beta)^{\alpha/m}}{(N/\beta)^{1/m} - 1} \tag{6.50}$$

In order to find the optimal number of levels, there remains to minimize G with respect to m. Differentiating G with respect to m, we get

$$\frac{dG}{dm} = \beta^{\alpha}\left(\frac{N}{\beta} - 1\right) \frac{\left(\frac{N}{\beta}\right)^{\alpha/m} \ln \frac{N}{\beta}}{m^2\left[\left(\frac{N}{\beta}\right)^{1/m} - 1\right]^2} \left[\alpha - (\alpha - 1)\left(\frac{N}{\beta}\right)^{1/m}\right] \tag{6.51}$$

In practical situations, $N/\beta > 1$. Under this assumption, dG/dm is of the same sign and has the same roots as $\alpha - (\alpha - 1)(N/\beta)^{1/m}$. Equating this last term to zero, we get $m = m_*$, as given in Fact 6.1.

From the sign of dG/dm, we note that G is minimum for $m = m_*$. This terminates the proof of Fact 6.1.

Conclusion: The assumption introduced in this section is intuitively justified a posteriori, since the solution of

$$\frac{\ln \frac{N}{\beta}}{\ln \frac{\alpha}{\alpha - 1}} = 1$$

with respect to N is unique and equal to $N_0 = \beta\alpha/(\alpha - 1)$. In other words, the optimal number of levels for a design of a network with $\beta\alpha/(\alpha - 1)$ nodes is equal to one.

6.5.4 Variations and Limiting Behavior of the Optimal Solution with Respect to the Design Variables

The behavior of the computational cost and the degree vector at optimality, given m, will be studied with respect to m. Of importance is the fact that $G(m)$ converges fairly fast to its minimum value, versus m, and remains very close to that value as m grows to infinity. A similar phenomenon will characterize the behavior of the n_k's versus m. We will also notice that, as α goes to infinity, the optimal solution, given m, is such that all layer subnets must be of equal size; such a property was, for finite α, only true at global optimality, i.e., $m = m_*$.

The variations of G_* and m_* with respect to α, β, N, do not disclose any remarkable property, except that m_* rapidly reaches its asymptotic value as α becomes greater than 2.

In what follows, we will restrict the study to the practical

situation where

$$\beta \geqq 1, \quad \frac{N}{\beta} > 1 \tag{6.52}$$

6.5.6.1 Behavior of the Optimal Solution and Objective Function, Given m

a. $\underline{G}(m)$ versus m: Section 6.5.2 showed that as m increases from zero to infinity, $\underline{G}(m)$ decreases, reaching its minimum at $m = m_*$ (note $m_* > 0$ because of Eq. (6.52) and $\alpha > 1$), and then increases. The limit of $\underline{G}(m)$, as m goes to infinity, is:

$$G_\infty \triangleq \lim_{m \to \infty} \underline{G}(m) = \beta^{\alpha-1} \frac{\alpha^\alpha}{(\alpha - 1)^{\alpha-1}} N \tag{6.53}$$

Proof

Let us first derive some intermediary results. From Eq. (6.23)

$$D_{m+1} = \alpha^m \left[1 - \left(1 - \frac{1}{\alpha}\right)^m\right] \qquad \forall\, m \geqq 1$$

Hence, when m goes to infinity

$$\begin{cases} D_{m+1}/\alpha^m \to 1 \\ D_m/D_{m+1} \to \dfrac{1}{\alpha} \\ \dfrac{m(\alpha - 1)^m}{D_{m+1}} \to 0 \\ D_{m+1}\alpha^{-(m\alpha^m)/D_{m+1}} \to 1 \end{cases} \tag{6.54}$$

Consequently, substituting these limits into Eq. (6.35), we arrive at Eq. (6.53). Notice that the difference

$$G_\infty - G_* = \beta^\alpha \frac{\alpha^\alpha}{(\alpha - 1)^{\alpha-1}}$$

is independent of N, and that the relative difference

$$\frac{G_\infty - G_*}{G_*} = \frac{\beta}{N - \beta}$$

goes to zero as N becomes large.

In what follows, four sets of figures (Figs. 6.2 - 6.9), each corresponding to a specific value of the pair (α, β), $\{(3,1), (3,3)$ $(5,1)$ $(5,3)\}$, will be shown. In each set, the functions $\underline{G}(m)$ and $\underline{G}(m)/G_*$ are plotted with respect to m and for several values of N, $N = \{50, 10^2, 10^3, 10^4, 10^5, 10^6\}$. The curves, $\underline{G}(m)$ versus m, illustrate the initially decreasing, then slightly increasing and asymptotic behavior of the optimal computational cost for a fixed m. By comparing $\overline{G}(m)$ to N^α, we are able to appreciate the enormous computational gains obtained through the application of the MHT.

The curves, $G(m)/G_*$ versus m, illustrate the "<u>locking</u>" effect whereby, once G(m) reaches its minimum value of $m = m_*$, Eq. (6.29), it will appear as if it remains indefinitely at that value. They also illustrate the fairly fast convergence of $\underline{G}(m)$toward a value close to the minimum, for a value of m relatively smaller than m_*. This indicates that we may actually obtain most of the computational gains with hierarchical structures whose number of levels (m) is much <u>smaller</u> than the optimal ones (m_*). Finally, the four sets are present in order to indicate the effect of the design parameters α and β on the behavior of the aforementioned functions.

b. <u>n_k versus m</u>

Differentiating Eq. (6.21) with respect to m, we find, after simplification

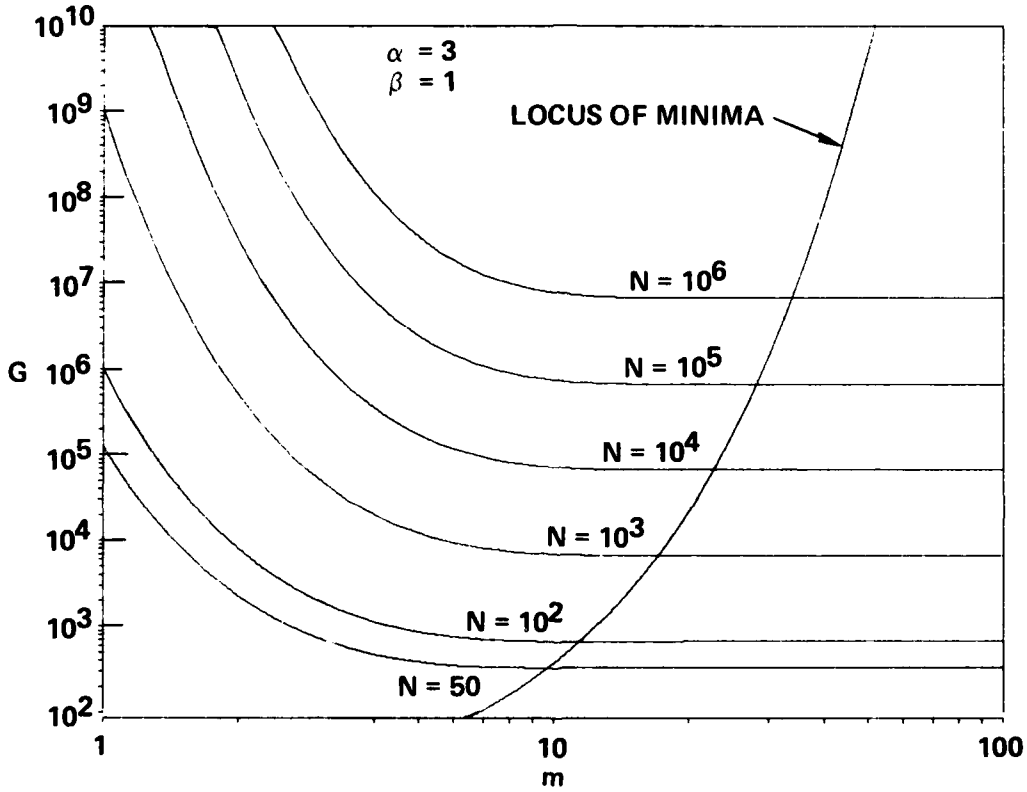


Figure 6.2. Minimum Computational Cost G(m, α, β), Given m; α= 3, β = 1.

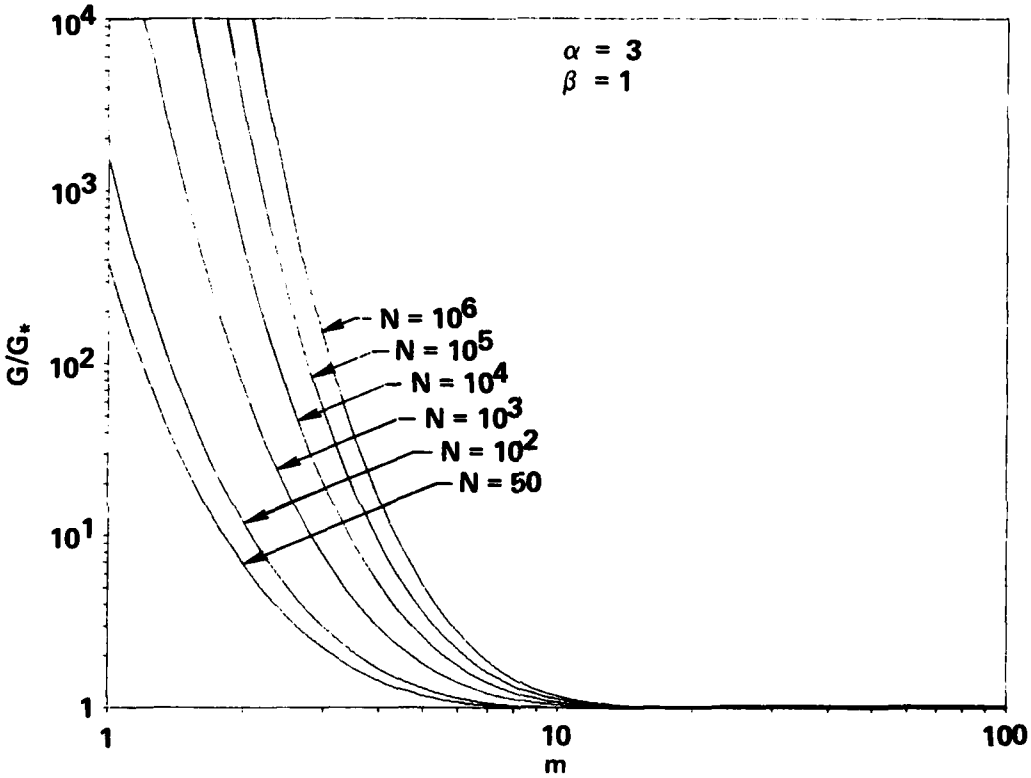


Figure 6.3. Ratio of Computational Cost at Optimality Given m, and at Global Optimality, G/G_*; $\alpha = 3, \beta = 1$.

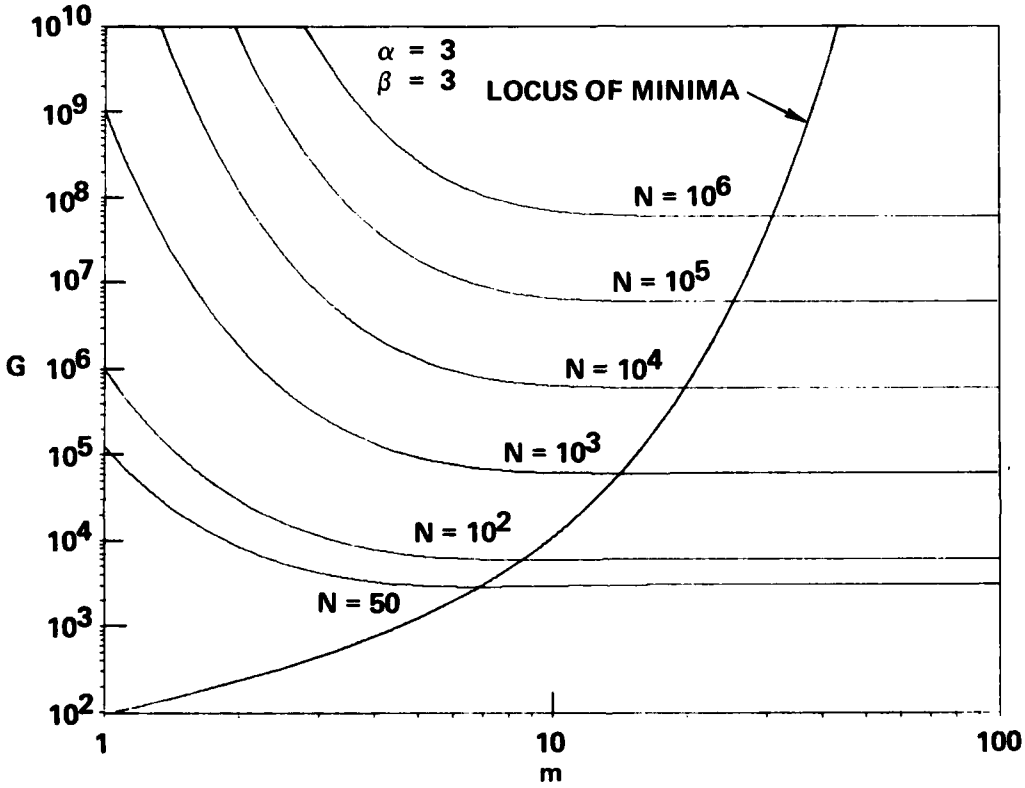


Figure 6.4. Minimum Computational Cost G(m, α, β) Given m; $\alpha = 3$, $\beta = 3$.

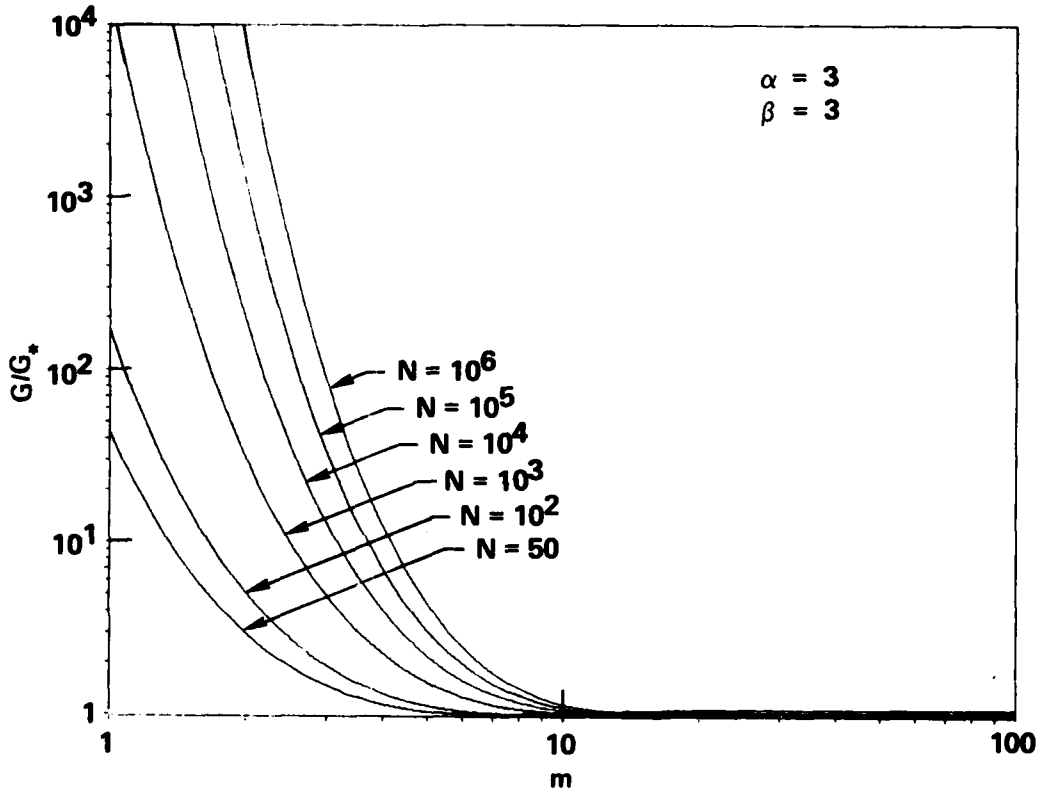


Figure 6.5. Ratio of Computational Cost at Optimality Given m, and at Global Optimality; $\alpha = 3$, $\beta = 3$.

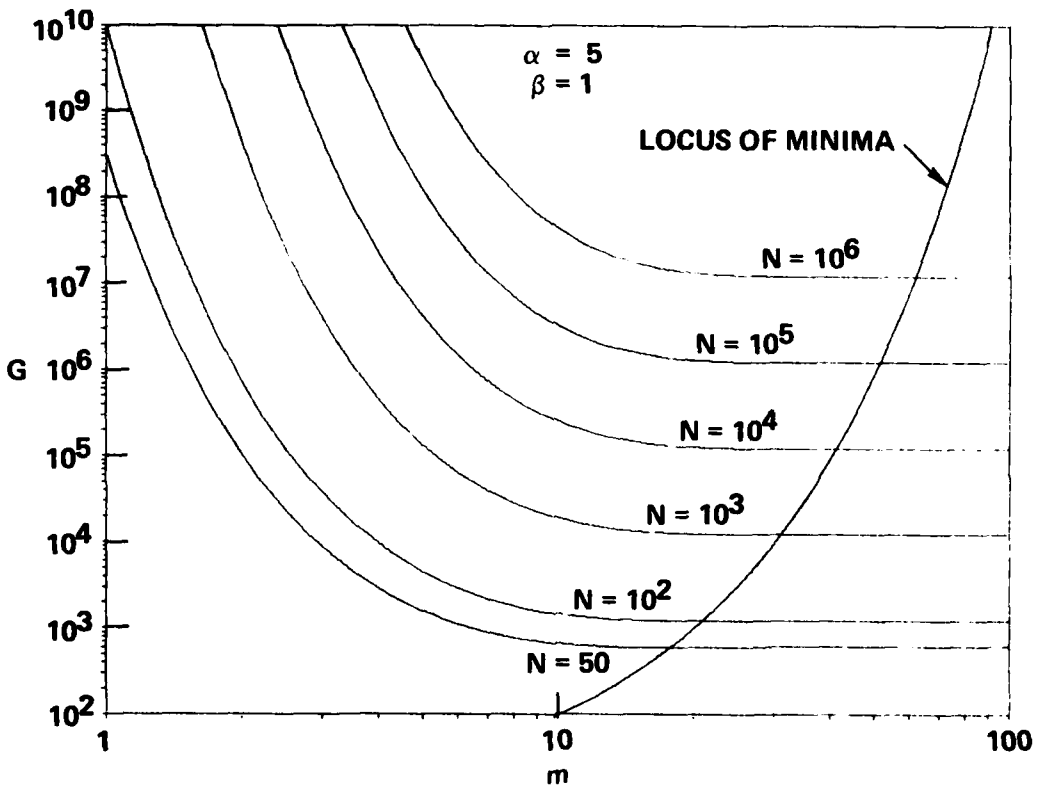


Figure 6.6. Minimum Computational Cost G(m, α, β), Given m; α = 5, β = 1.

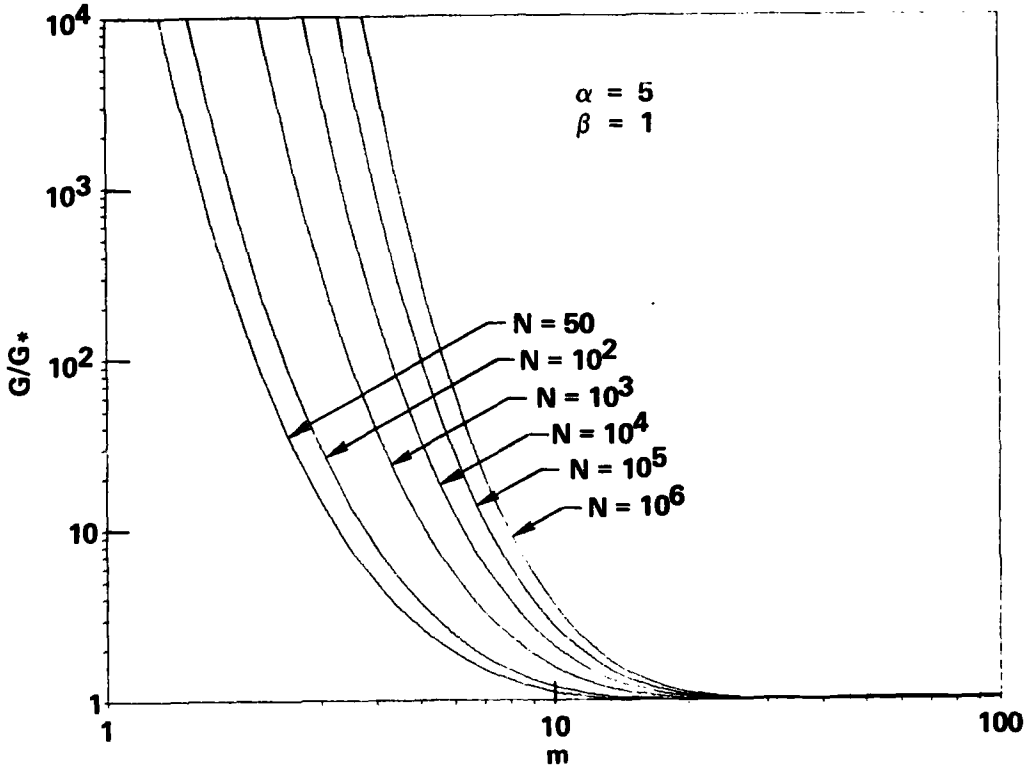


Figure 6.7. Ratio of Computational Cost at Optimality Given m, and at Global Optimality, G/G_*; $\alpha = 5$, $\beta = 1$.

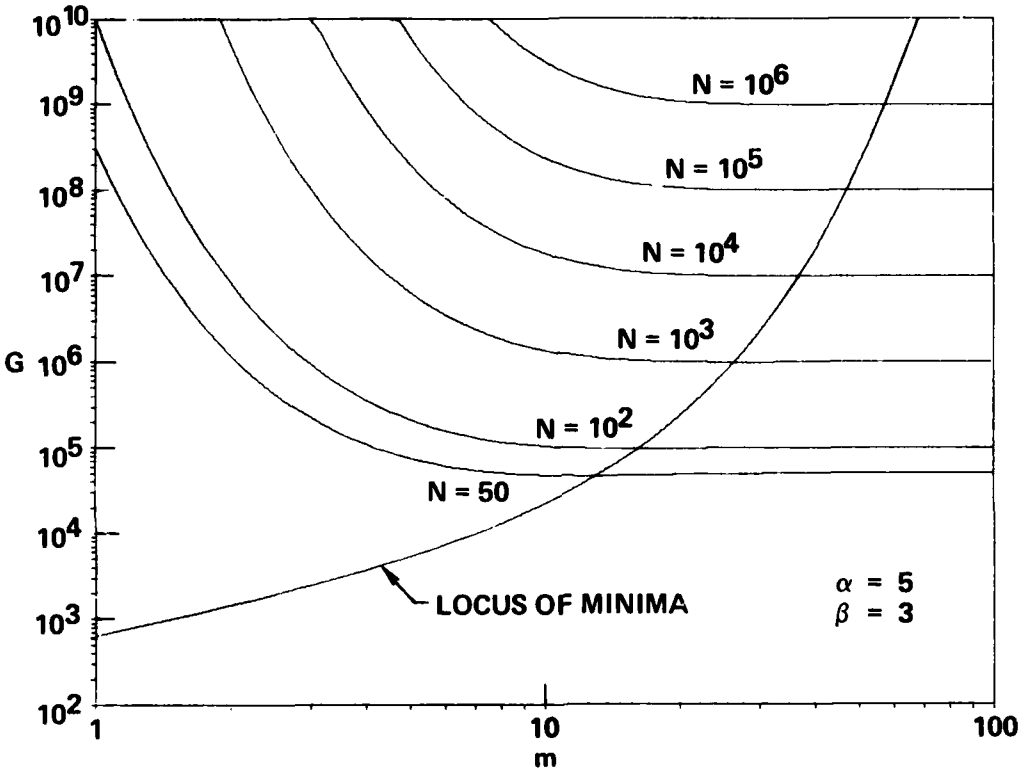


Figure 6.8. Minimum Computational Cost G(m, α, β), Given m; α = 5, β = 3.

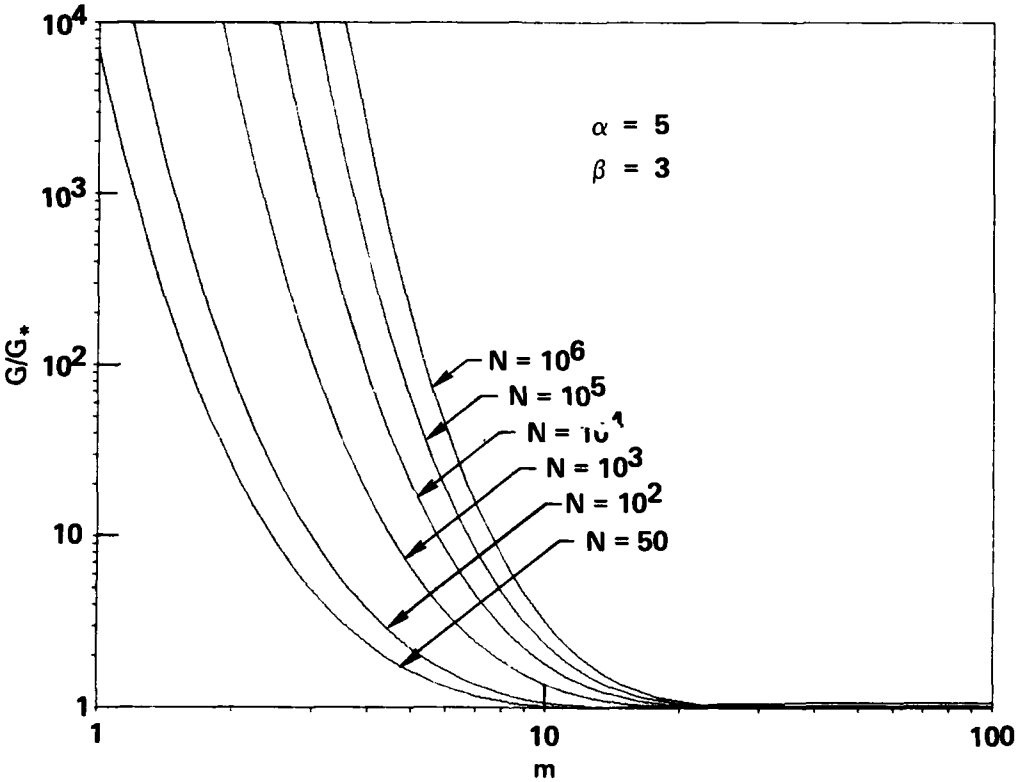


Figure 6.9. Ratio of Computational Cost at Optimality Given m, and at Global Optimality, G/G_*; $\alpha = 5, \beta = 3$.

$$\frac{dn_k}{dm} = n_k \frac{(\alpha - 1)^{m-k}\alpha^{m+k-1}}{D^2_{m+1}} \left(\ln \frac{\alpha}{\alpha - 1}\right)\left[\left(\frac{\alpha - 1}{\alpha}\right)^m - 1 - \ln \frac{N}{\beta} + m \ln \frac{\alpha}{\alpha - 1}\right]$$

For $\alpha > 1$, dn_k/dm is of the same sign as the expression in brackets (to be denoted by Z). From Fig. 6.10, we notice that Z has a unique root, m_0, which is such that

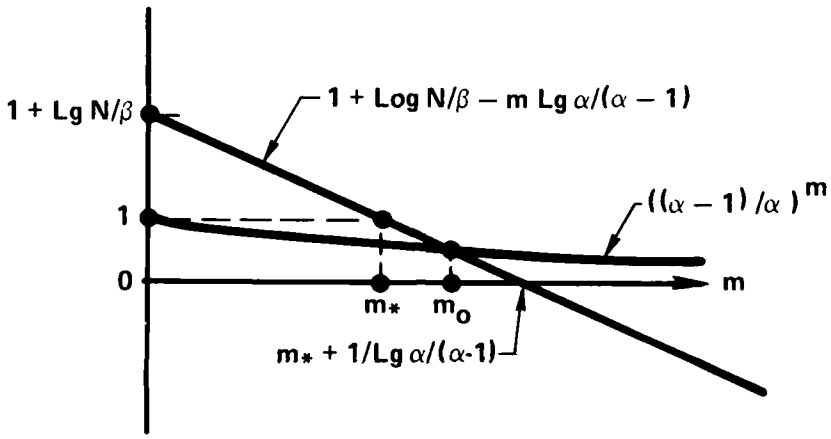


Figure 6.10. Root of dn_k/dm.

$$m_* \leq m_0 \leq m_* + \frac{1}{\ln \frac{\alpha}{\alpha - 1}}$$

Also, for $m < m_0$, Z is negative and, conversely, for $m > m_0$, Z is positive. Z is equal to zero at a value of m, m_0, which is <u>independent</u> of k. Consequently, as m varies from zero to infinity

(i) all n_k's, $k < m_0$, decrease, reach a minimum and then increase toward their limits.

(ii) all n_k's, $k > m_0$, increase toward their limits.

The limit of n_k for k fixed, as m goes to infinity, is

$$\begin{cases} \lim\limits_{m \to \infty} n_1 = \beta \dfrac{\alpha}{\alpha - 1} \\ \lim\limits_{m \to \infty} n_k = \dfrac{\alpha}{\alpha - 1} \qquad k \geq 2,\ k \ \underline{\text{fixed}} \end{cases}$$

The proof is immediate, after observing that

$$\lim_{m \to \infty} m \frac{(\alpha - 1)^{m-k} \alpha^{k-1}}{D_{m+1}} = \lim_{m \to \infty} \frac{m}{\alpha} \frac{\left(\frac{\alpha - 1}{\alpha}\right)^{m-k}}{1 - \left(\frac{\alpha - 1}{\alpha}\right)^{m}} = 0$$

and then, substituting this limit into Eq. (6.21). If we let k vary, more particularly, if we let k = m, then

$$\lim_{m \to \infty} \frac{m\alpha^{m-1}}{D_{m+1}} = +\infty$$

Hence

$$\lim_{m \to \infty} n_m = 0$$

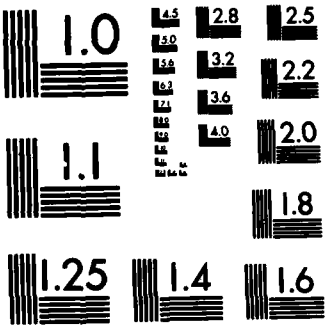
Moreover, $\ln n_m$ has as an asymptote a straight line of equation

$$-\frac{m}{\alpha} \ln \frac{\alpha}{\alpha - 1} + \ln \frac{\alpha}{\alpha - 1} \left(\frac{N}{\beta}\right)^{1/\alpha}$$

The behavior of n_k versus m, for $\underline{k \leq m_*}$, exhibits the "locking" effect as previous described. Moreover, this phenemonon is quite remarkable here, since the limit of n_k, as m goes to infinity, is equal to the value of n_k at global optimality. Fig. 6.11 illustrates the above properties.

AD-A122 995 DESIGN CONSIDERATIONS FOR LARGE COMPUTER COMMUNICATION NETWORKS(U) CALIFORNIA UNIV LOS ANGELES SCHOOL OF ENGINEERING AND APPLIED SCIENCE F KAMOUN APR 76 #/5

UNCLASSIFIED UCLA-ENG-7642 DAHC15-73-C-0368 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

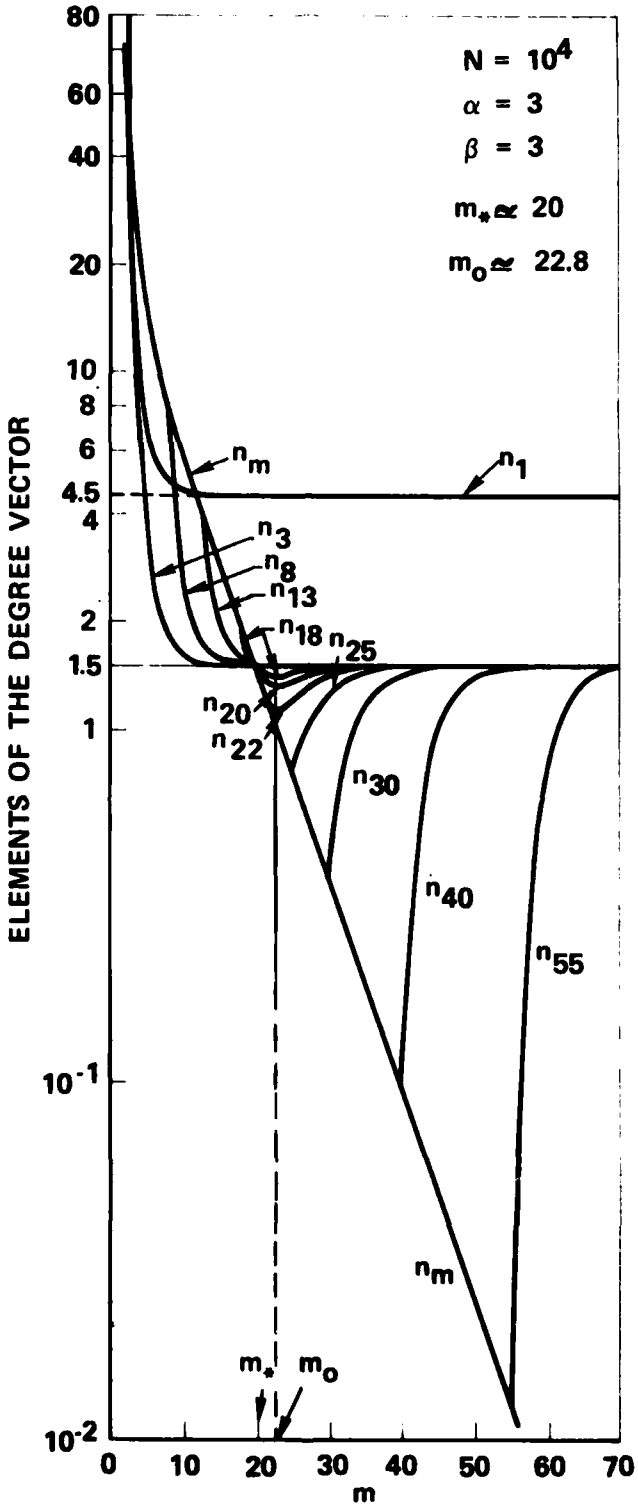


Figure 6.11. Variations of the Optimal Degrees with Respect to m.

6.5.4.2 Behavior of G_* and m_* with Respect to α, β, N

a. G_* versus α: Differentiating Eq. (6.31) with respect to α, we find

$$\frac{dG_*}{d\alpha} = \left(\frac{N}{\beta} - 1\right)\beta^{\alpha} \frac{\alpha^{\alpha}}{(\alpha - 1)^{\alpha-1}} \ln \beta \frac{\alpha}{\alpha - 1}$$

Under the conditions of Eq. (6.52) and $\alpha > 1$, G_* is an increasing function of α. As α goes to infinity, G_* will be asymptotic to the expression, $(N/\beta - 1)e(\alpha - 1/2)\beta^{\alpha}$.

If $\beta = 1$ the asymptote is a straight line, which implies that at the limit, G_* will show a linear growth with α.

b. G_* versus N: Eq. (6.31) shows that G_* varies linearly with N. It is also important to notice that the optimal hierarchical structure reduces the computational cost from the order of N^{α} steps to the order of N steps!

c. G_* versus β: Differentiating Eq. (6.31) with respect to β, we arrive at

$$\frac{dG_*}{d\beta} = \frac{\alpha^{\alpha}}{(\alpha - 1)^{\alpha-1}} \beta^{\alpha-2}[(\alpha - 1)N - \alpha\beta].$$

Consequently, G_* is an increasing and then decreasing function, as β varies from 1 to N (for $N > \frac{\alpha}{\alpha - 1}$ and under the conditions of Eq. (6.52)). Equating the derivative to zero, we find

$$\beta_0 = \frac{\alpha - 1}{\alpha} N \tag{6.55}$$

Substituting Eq. (6.55) into Eq. (6.31), we arrive at the maximum value of G_* (with respect to β), $G_*(\beta = \beta_0) = N^{\alpha}$. This maximum cost corresponds to a design with no hierarchical structure. This last equation

checks with the fact that for a set α, β, N, satisfying Eq. (6.55), the optimal number of levels, m_*, is equal to one (Eq. (6.29)), i.e., for such a set, a non-hierarchical design is optimal.

d. m_* versus N and β: Recall Eq. (6.29), $m_* = (\ln N/\beta)/\left(\ln \alpha/(\alpha - 1)\right)$. Hence, m_* varies as a logarithm of N/β. This logarithmic behavior appears to be characteristic of hierarchical structures [CHOW 74], (see Chapter 2).

e. m_* versus α: Differentiating Eq. (6.29) with respect to α, we find

$$\frac{dm_*}{d\alpha} = \frac{\ln N/\beta}{\alpha(\alpha - 1)[\ln (\alpha/(\alpha - 1))]^2}$$

Consequently, for $\alpha > 1$, m_* is an increasing function of α. The limiting values of m are

$$\alpha \to 1^+ \implies \begin{cases} m_* \to 0 \\ \dfrac{dm_*}{d\alpha} \to \infty \end{cases}$$

$$\alpha \to +\infty \implies \begin{cases} m_* \to +\infty \\ \dfrac{dm_*}{d\alpha} \to \ln N/\beta \end{cases}$$

The asymptote, as α goes to infinity, is the straight line

$$m_* = \left(\alpha - \frac{1}{2}\right) \ln N/\beta$$

Fig. 6.12 shows the plots of m_* versus α for several values of N/β. Notice that m_* rapidly reaches its asymptotic value as α becomes greater than 2. Also,

$$m_* = 1 \quad \text{for} \quad \alpha = \frac{N/\beta}{N/\beta - 1}$$

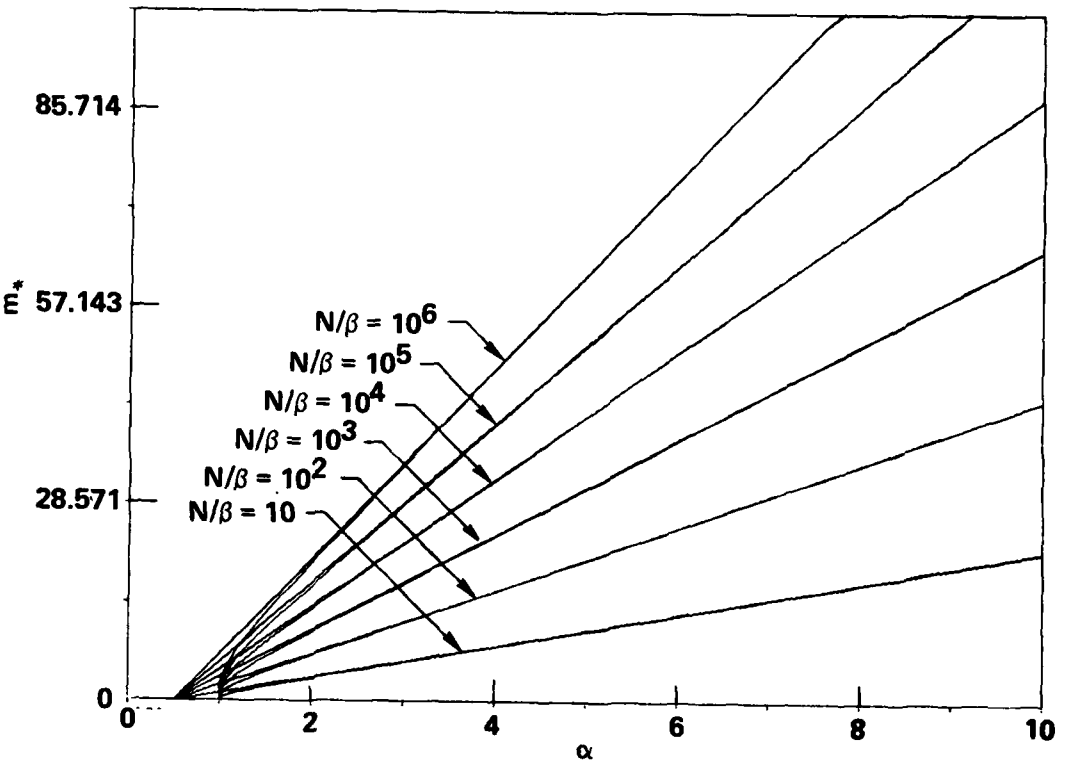


Figure 6.12. Optimal Number of Levels in the Hierarchical Design.

which means that for $1 \leq \alpha \leq \frac{N/\beta}{N/\beta - 1}$, <u>no partitioning</u> is required: furthermore, a loss will be incurred if we try to do so.

6.5.4.3 Limiting Behavior of the Optimal Solution, Given m, with Respect to α

a. $\underline{\alpha \to 1^+}$: From Eqs. (6.21) and (6.23), we obtain

$$\alpha \to 1^+ \quad \Rightarrow \quad \begin{cases} n_k \to +\infty & \text{for all k's, } k \neq m \\ n_m \to 0 \end{cases}$$

In other words, since the limit of n_m is zero, for any given m, $m \geq 2$, then, no clustering must be allowed. This confirms the earlier result obtained when looking at the behavior of m_* with respect to α.

b. $\underline{\alpha \to +\infty}$: From Eqs. (6.21) and (6.23), we find

$$\lim_{\alpha \to \infty} \frac{D_{m+1}}{m\alpha^{m-1}} = 1$$

Hence,

$$\alpha \to +\infty \quad \Rightarrow \quad \begin{cases} n_1 \to \beta \left(\frac{N}{\beta}\right)^{1/m} \\ n_k \to \left(\frac{N}{\beta}\right)^{1/m} & k = 2, \ldots, m \end{cases} \tag{6.56}$$

Thus,

$$\alpha \to +\infty \quad \Rightarrow \quad g_k \to \beta \left(\frac{N}{\beta}\right)^{1/m} \qquad k = 1, 2, \ldots, m$$

Consequently, as α goes to infinity, the optimal solution, given any m, is such that all layer subnets are of equal size. Such a property was found to be true for finite α, only at global optimality.

6.5.4.4 Optimal Clustering when $\alpha \leq 1$

Notice that if

$$1 < \alpha \leq \frac{N/\beta}{N/\beta - 1}$$

then, from Eq. (6.29), we conclude that $m_* \leq 1$, which means that no clustering is needed for α satisfying the equation above. Let us show that this result holds true if $0 < \alpha \leq 1$.

Proposition 6.3

Under the condition, $0 < \alpha \leq 1$, the global optimum of Problem 6.11 (with equal α's) is achieved for $m = 1$, i.e., no clustering is required.

Proof:

By contradiction, assume that $m > 1$; then, similar to the proof of Proposition 6.1 (Appendix C), let us fix m ($m > 1$, integer) and all degrees, $n_k(i_m, \ldots, i_{k+1})$, ($k = 2, \ldots, m$), and solve the reduced problem with respect to $n_1(i_m, \ldots, i_2)$'s. i.e.,

$$\begin{cases} \text{min:} \quad G_1 = \sum_{i=1}^{NC_1} [n_1(i)]^{\alpha} & \text{(see Eq. (6.8))} \\ \text{over:} \quad n_1(i) & \\ \text{s.t.:} \quad \sum_{i=1}^{NC_1} n_1(i) = N & \text{(see Eq. (2.1))} \end{cases}$$

$$n_1(i) \geq 0 \text{ real variable}$$

where NC_1 denotes the number of 1st level clusters, and $n_1(i)$ denotes the size of an arbitrary 1st level cluster. Since $\alpha \leq 1$, then $[n_1(i)]^{\alpha}$ is a concave function; hence, the objective function above, G_1, is a

concave function. Also, the set of feasible vectors, $\underset{\sim}{n}_1 = \{n_1(i)\}$ is a bounded convex polyhedron whose vertices correspond to vectors with all zeros except for one component equal to N, i.e., (N, 0, 0, ..., 0) and its permutations. As a result, we are faced with the optimization of a concave function over a bounded convex polyhedron, whose optimal solution is well known to be at a vertex. At any vertex, $G_1 = N^{\alpha}$; consequently, it is the minimum. Notice that this result is true for any NC_1, i.e., for any $n_k(i_m, \ldots, i_{k+1})$, $(k = 2, \ldots, m)$. In other words, for any vector $\underset{\sim}{n}$ satisfying the size constraint, Eq. (2.1), $G_1 \geq N^{\alpha}$, which, combined with Eq. (6.9), gives

$$G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) \geq N^{\alpha} + \sum_{k=2}^{m} G_k(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) \quad \forall\, \underset{\sim}{n}, \text{ feasible}$$

Hence, $$G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) > N^{\alpha} \quad \forall\, \underset{\sim}{n}, \text{ feasible}$$

Consequently, the optimal solution of Problem 6.11, for integer $m > 1$, must also satisfy the above inequality which is a contradiction, since, if $m = 1$, $G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = N^{\alpha}$.

Remark

Proposition 6.3 holds true for the general case where the α_k's are arbitrary, if $\alpha_1 \leq 1$.

6.5.5 Comparison of the Optimal Solution with Two Feasible Solutions

Two intuitive solutions emerge when dealing with hierarchical structures. The R-solution (R stands for root) corresponds to a regular tree representation, i.e., all degrees, at all levels, are equal.

$$r \triangleq n_k \qquad k = 1, 2, \ldots, m \tag{6.57}$$

The Q-solution models the situation where all layer subnets are of equal size.

$$q \triangleq g_k \qquad k = 1, 2, \ldots, m \tag{6.58}$$

Notice that the two solutions are identical when $\beta = 1$. The optimal solution, given m, will be referred to as the G-solution.

6.5.5.1 The Q-Solution

This solution has been studied in detail in the course of the proof of Fact 6.1 (Section 6.5.3). Of significance is the property that the Q-solution satisfies Proposition 6.2 at global optimality. The corresponding computational cost (denoted by Q) is given by Eq. (6.50). There remains to find the limit of Q as m goes to infinity. From Eq. (6.50),

$$\lim_{m \to \infty} Q = +\infty$$

The corresponding asymptote is a straight line whose equation is

$$Q_\infty = \left(\frac{N}{\beta} - 1\right)\left[\frac{m}{\ln N/\beta} + \alpha - 1\right] \tag{6.59}$$

6.5.5.2 The R-Solution

From Eq. (6.57) and the size constraint, $r = N^{1/m}$. With this assignment, the computational cost is

$$R = N^{(\alpha-1)/m}\left[N + \beta^\alpha \frac{N - N^{1/m}}{N^{1/m} - 1}\right] . \tag{6.60}$$

Differentiating Eq. (6.60) with respect to m, we find

$$\frac{dR}{dm} = \frac{N^{(\alpha-1)/m} \ln N}{m^2 (N^{1/m} - 1)^2} \Big[\beta^\alpha (N - 1) N^{1/m}$$

$$- (\alpha - 1)(N^{1/m} - 1) \left(\beta^\alpha (N - N^{1/m}) + N(N^{1/m} - 1) \right) \Big] \quad (6.61)$$

If we let Y represent the expression in brackets and $X = N^{1/m}$, then

$$Y = -(\alpha - 1)(N - \beta^\alpha) X^2$$

$$+ \left(2(\alpha - 1)N + \beta^\alpha \left(2N - \alpha(N + 1) \right) \right) X + (\alpha - 1)N(\beta^\alpha - 1) \quad (6.62)$$

$\frac{dR}{dm}$ is of the same sign as Y. Notice that

$$\begin{aligned} X \to -\infty &\Rightarrow Y \to -\infty \\ X = 1^+ &\Rightarrow m \to +\infty \Rightarrow Y = (N - 1)\beta^\alpha \geq 0 \\ X \to +\infty &\Rightarrow m \to 0^+ \Rightarrow Y \to -\infty \end{aligned} \quad (6.63)$$

Thus, there exists a unique root, X_R, which is greater than one. Let m_R be such that

$$N^{1/m_R} = X_R \Rightarrow m_R = \frac{\ln N}{\ln X_R} \quad (6.64)$$

From the above remarks, we conclude that R is minimum for $m = m_R$. There remains to find the limit of R as m goes to infinity. From Eq. (6.60), $\lim_{m \to \infty} R = +\infty$. The corresponding asymptote is

$$R_\infty = (N - 1) \left[\frac{\beta^\alpha}{\ln N} m + 1 + (\alpha - 2)\beta^\alpha \right]. \quad (6.65)$$

6.5.5.3 Comparison of the Three Solutions

Let us first summarize the properties of the three solutions.

1. All three solutions possess an optimal number of levels,

m_* (Eq. (6.29)) for G and Q, and m_R (Eq. (6.64)) for R.

2. For large values of m, the G-solution is asymptotic to a constant; whereas, the Q and R solutions asymptotically grow linearly with m. In other words, for $m \geq m_*$, G, contrary to Q and R, is not sensitive to m.

3. The G and Q solutions meet, for any α, at $m = m_*$ and, for $\alpha \to \infty$, at any given m.

4. The Q and R solutions are identical for $\beta = 1$.

Figs. 6.13 and 6.14 illustrate the above properties. Moreover, we notice that, as expected, G is always smaller than or equal to Q and R, and that, for $\beta \neq 1$, Q is certainly better than R, for values of m in the neighborhood of m_*, but not necessarily outside.

6.5.6 Suboptimal Integer Solution

We propose, below, a heuristic algorithm which generates an integer solution from the optimal real-valued solution.

Algorithm: Given N, m, β

1. $M \leftarrow m$, $\text{OLD } N \leftarrow N$ [save old values]

2. Compute n_m from Eq. (6.21)

   $nd_m \leftarrow \lceil n_m \rceil$ ceiling operation

   $N \leftarrow \frac{N}{nd_m}$ adjust size constraint

   $m \leftarrow m - 1$

   If $m > 1$ GO TO 2.

   Else $nd_1 \leftarrow \lceil N \rceil$, $ND \leftarrow \prod_{i=1}^{m} nd_i$

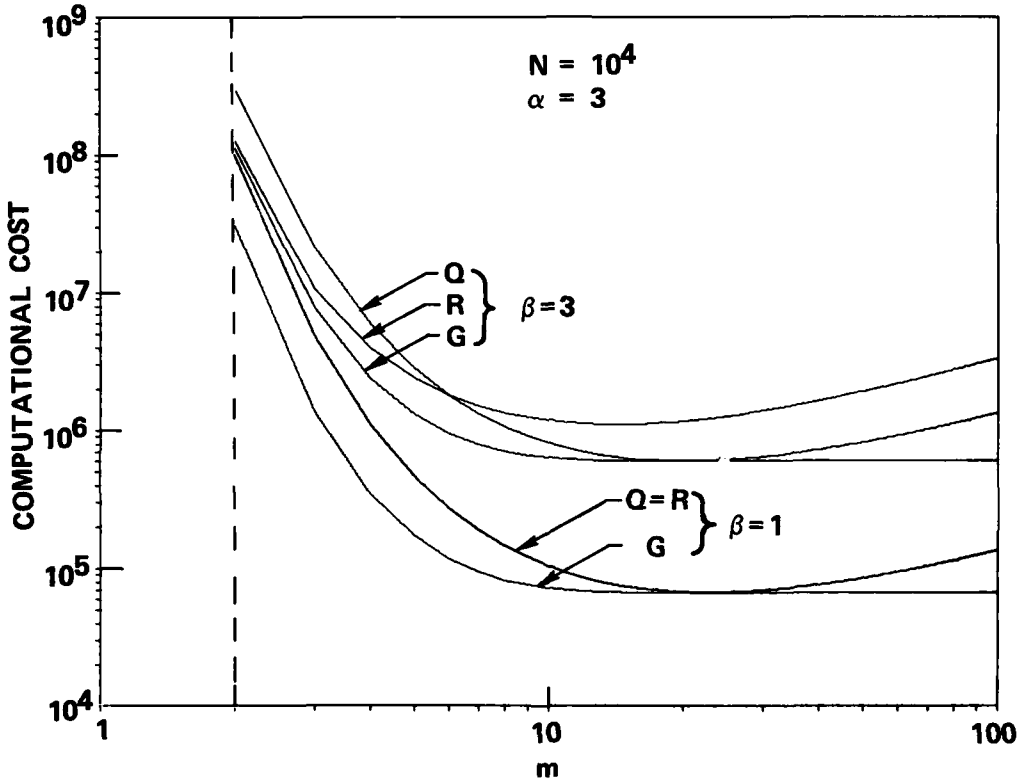


Figure 6.13. Comparison of the Optimal Solution with Two Feasible Solutions; $N = 10^4$, $\alpha = 3$.

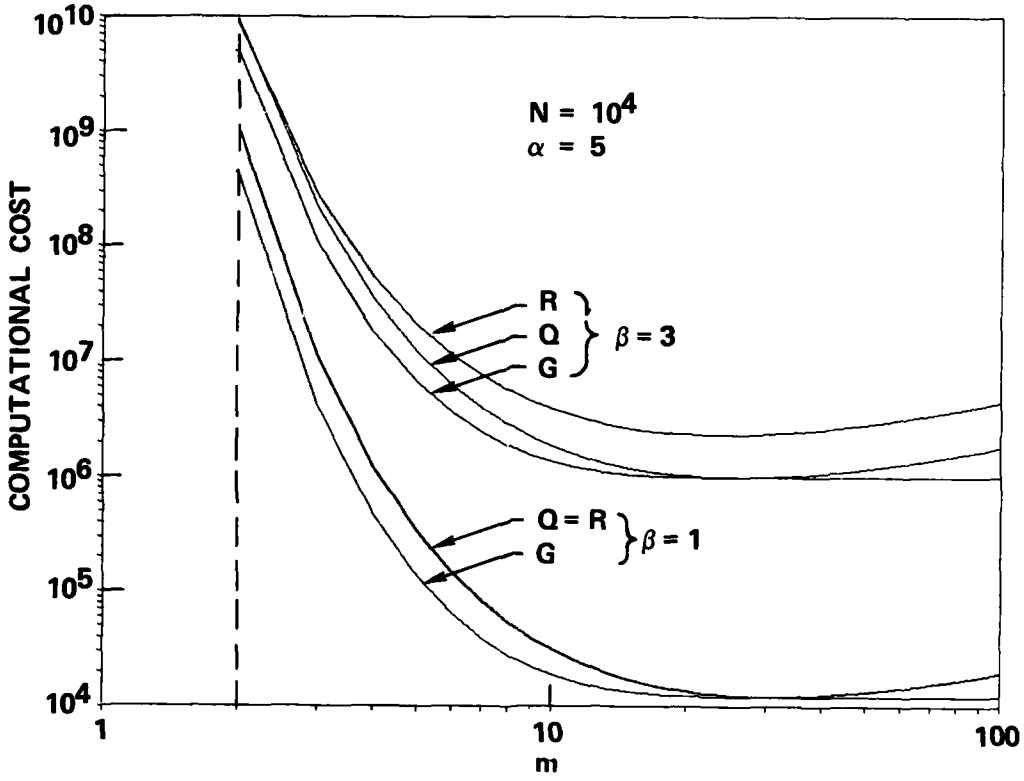


Figure 6.14. Comparison of the Optimal Solution with Two Feasible Solutions; N = 10^4, $\alpha = 5$.

3. $m \leftarrow M, \quad N \leftarrow \text{OLD } N$

4. DO $i = 2, 3, \ldots, m$

$\text{OLD } nd_i \leftarrow nd_i$

$nd_i \leftarrow \left\lceil \frac{N}{ND} nd_i \right\rceil$

$ND \leftarrow \frac{(ND)(nd_i)}{\text{OLD } nd_i}$

5. Compute corresponding computational cost

$$GD = \sum_{k=1}^{m} \left(\prod_{i=k+1}^{m} nd_i \right) \left(\beta_k (nd_k)^{\alpha} \right)$$

The main idea underlying the above algorithm is to always compute the degree of the highest level cluster; then after readjustment of the number of nodes, N, the operation is repeated for an equivalent problem with one less level. This idea is incorporated in Step 2. The purpose of Step 4 is to adjust the integer solution, $\{nd_i\}$, obtained in Step 2, in order to reduce the slack in the size constraint, i.e., in order to reduce the difference between ND and the given value of N.

The global minimum may easily be obtained when m is restricted to a certain range of values. This could be done by computing the cost incurred for each m (using the above algorithm) and then choosing the number of levels which corresponds to the minimum cost.

The performance of the above algorithm is illustrated in the following sets of figures, Figs. 6.15 - 6.18. These figures must be compared to those corresponding to the real-valued solution, Figs. 6.2 - 6.9. We notice that the plots of G(GD) versus m roughly exhibit the same shape as found for the real-valued solution. The two solutions

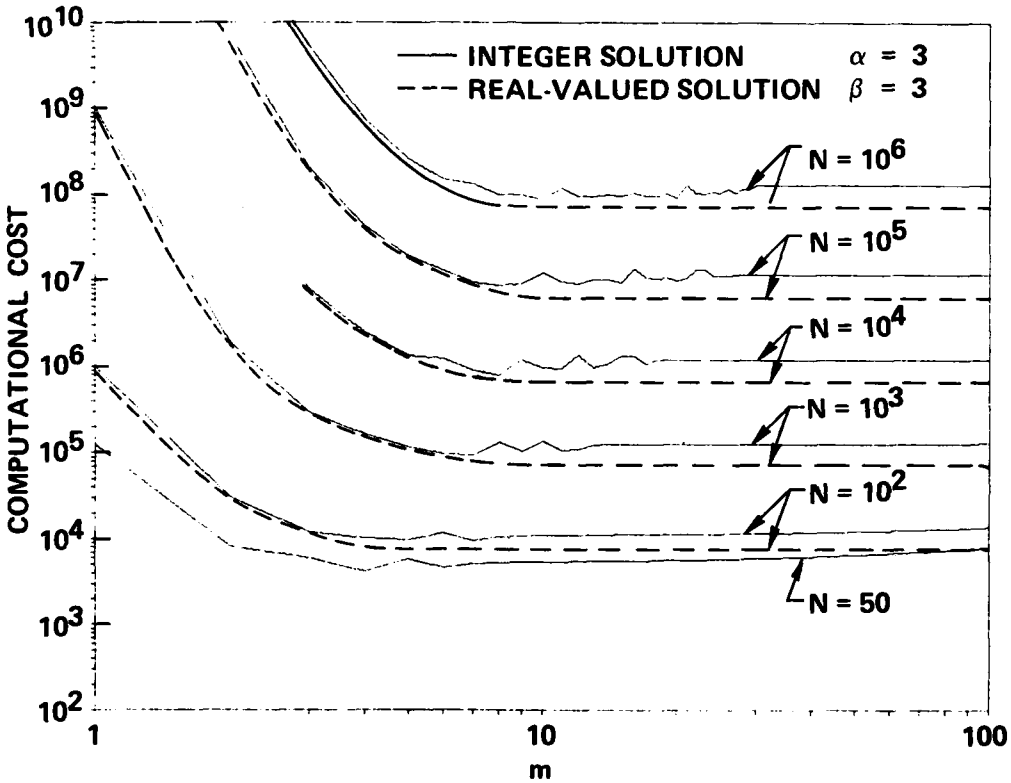


Figure 6.15. Suboptimal Integer Solution, Given m; $\alpha = 3$, $\beta = 3$.

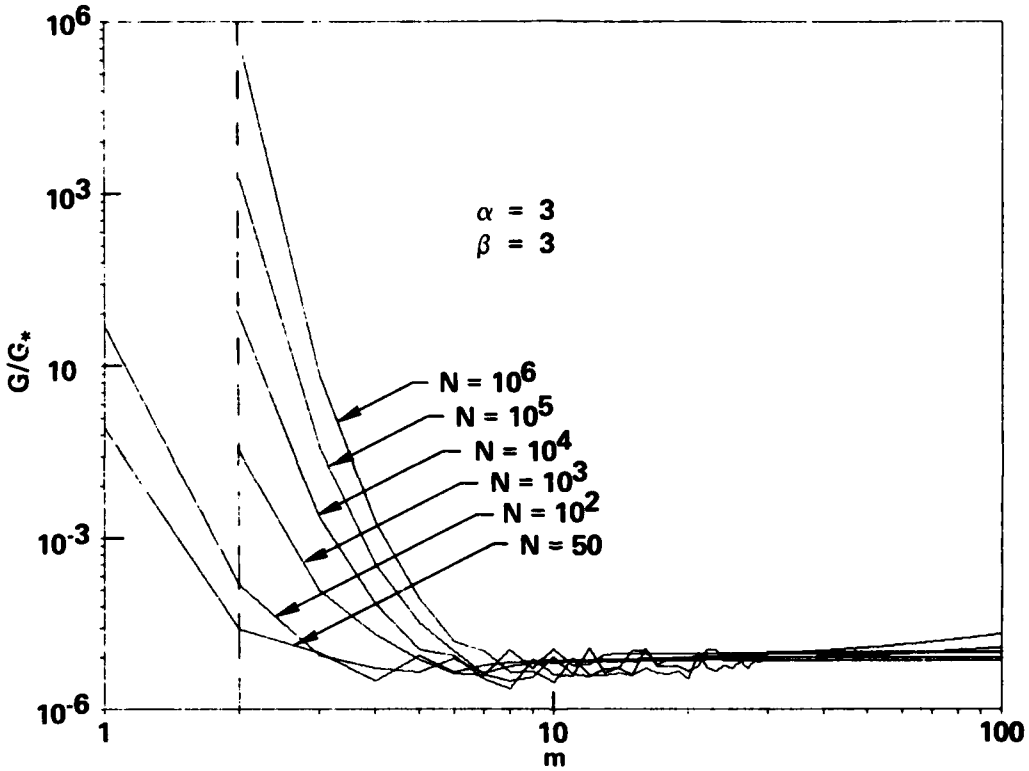


Figure 6.16. Ratio of Suboptimal Integer Solution, Given m, to the Real-Valued Global Optimal Solution; $\alpha = 3$, $\beta = 3$.

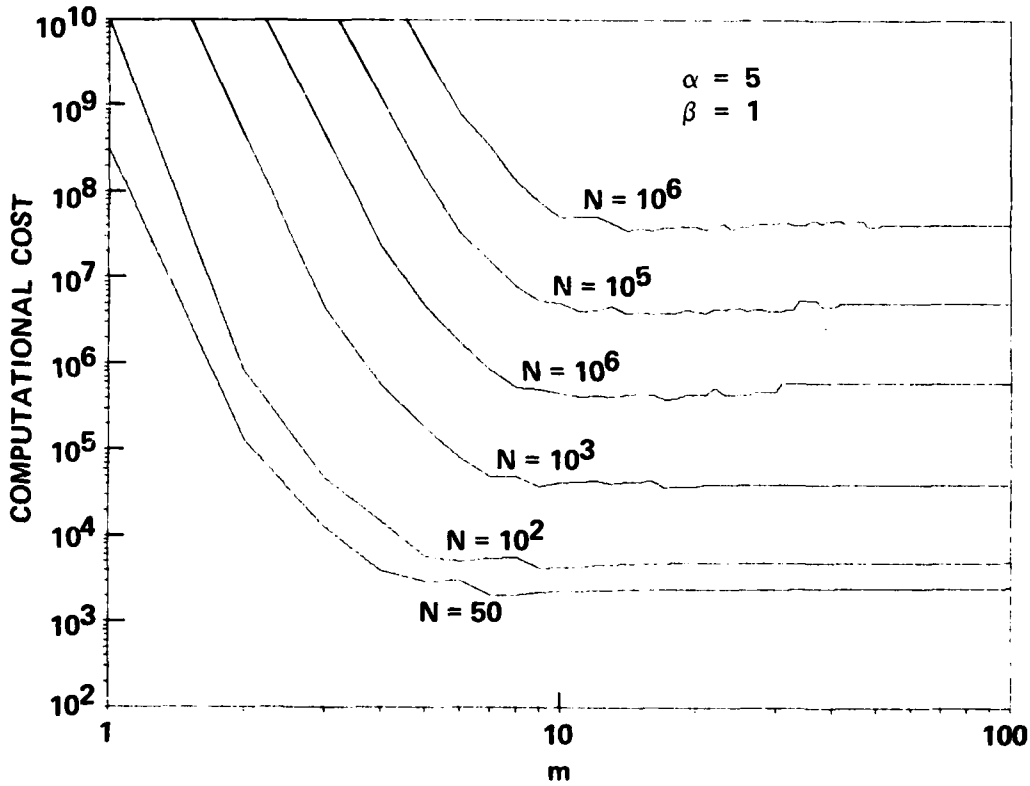


Figure 6.17. Suboptimal Integer Solution, Given m; $\alpha = 5, \beta = 1$.

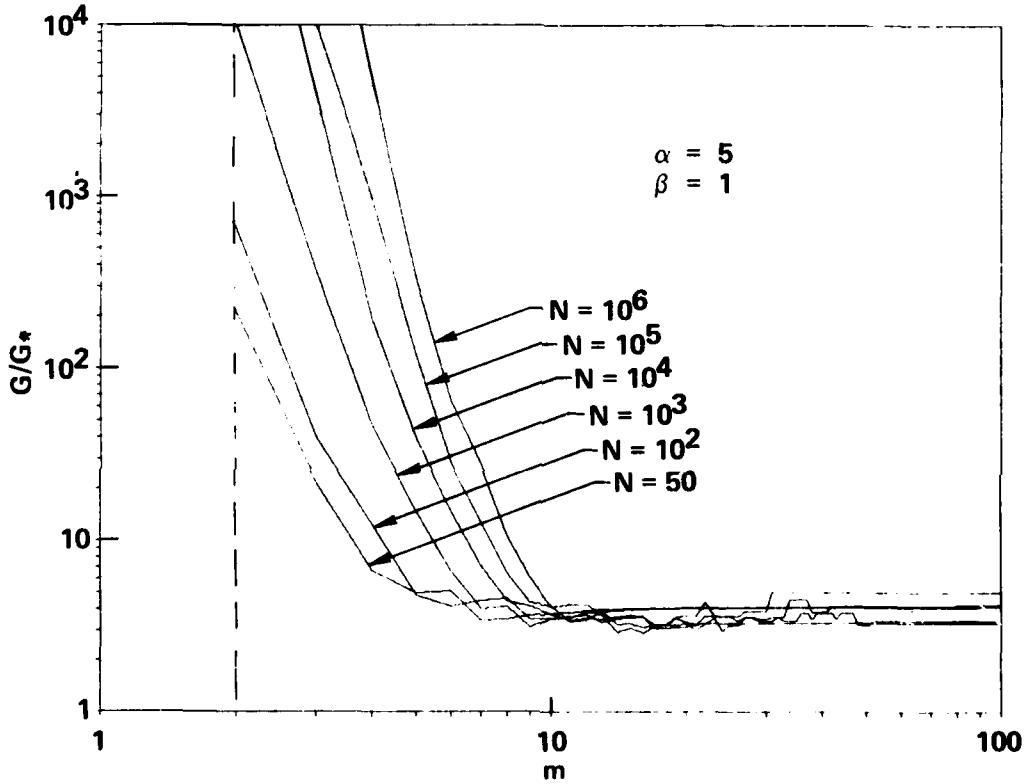


Figure 6.18. Ratio of the Sub-optimal Integer Solution, Given m, to the Real Valued Global Optimum Solution; $\alpha = 5, \beta = 1$.

are also fairly close for small values of m. Moreover, the algorithm seems to work better in the smaller region of α, and it appears that it is not that sensitive to β. The plots, GN/G_*, show the initial fast drop in computational cost followed by a levelling behavior to a value within $2G_*$ for $\alpha = 3$ and $5G_*$ for $\alpha = 5$, and which seems to be fairly independent of N.

6.6 Application to Other Special Cases

This section deals with the following two cases:

(i) uniform design strategy, variable gate assignment, i.e.,

$\alpha_k = \alpha \quad \forall\, k = 1, \ldots, m$ and β_k's variables.

(ii) proportional assignment of gates.

This section is treated in Appendix C. A summary of the results is presented below.

In (i) explicit expressions for the optimal degree vector and computational cost, given m, have been derived. Some partial results related the the global optimality have been found when β_k is of the form $\beta_k = \gamma^{k-2}\beta \quad k \geq 2$.

With respect to (ii), the number of kth level gates to be selected is proportional to the number of $k\text{-}1^{st}$ level gates from which they are selected. The corresponding solution was found to be of no practical interest. The solution is

$$\begin{cases} n_1 = 0, \quad n_m = +\infty \\ n_2, \ldots, n_{m-1} \quad \text{any value different from } 0 \text{ or } \infty, \\ \qquad\qquad \text{such that } \prod_{i=1}^{m} n_i = N. \end{cases}$$

6.7 Delay Expression for Hierarchical Networks

Given a network with an m-level hierarchical structure, the problem is to express the total average delay in the network in terms of the average delays in the layer subnets composing the network. We will assume that all k^{th} layer subnets are of equal size, i.e.,

$$n_k(i_m, i_{m-1}, \ldots, i_{k+1}) = n_k \qquad \forall\, k = 1, \ldots, m$$

and that they will induce the same average delay, T_k, over the traffic. This last assumption will usually appear as a design constraint, and as such, it is a reasonable one.

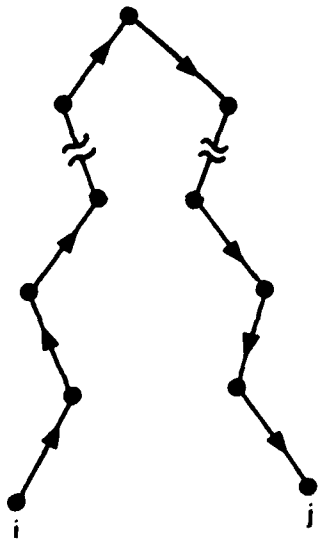
The traffic in hierarchical networks may be divided into m classes. Class k traffic is defined as the traffic between pairs of nodes which belong to the same k^{th} level cluster but not to any lower level clusters. If we define τ_k as the average delay incurred by class k traffic, then obviously

$$T = \sum_{k=1}^{m} \frac{\Gamma_k}{\Gamma} \tau_k \tag{6.66}$$

where Γ_k = total class k traffic

Γ, T are as defined in Eqs. (4.1) and (4.2)

From Flow Assumption 6.1, we know that class k traffic, when going from its origin to its destination, has to go up through k-1 layers to the kth common layer. Then it will go down k-1 layers in a manner described by Fig. 6.19.



LEVEL	DELAY	DEGREE
k	T_k	n_k
k-1	T_{k-1}	n_{k-1}
3	T_3	n_3
2	T_2	n_2
1	T_1	n_1
0		

Figure 6.19. Illustration of the Tree Path of class-k Traffic.

Consequently,

$$\tau_k = 2 \sum_{i=1}^{k-1} T_i + T_k \qquad k \geq 1 \tag{6.67}$$

Γ_k may also be evaluated using the flow assumption but, in general, it will yield too complicated an expression. If we assume a <u>uniform traffic</u> pattern, i.e., $\gamma_{jk} = \gamma \;\; \forall \; (j,k)$, then

$$\Gamma_k = N(n_k - 1)\left(\prod_{i=1}^{k-1} n_i\right)\gamma \qquad k \geq 1 \tag{6.68}$$

<u>Proof:</u>

Let C_k be a kth level cluster, and let $C_{k-1}(i)$ and $C_{k-1}(j)$ be two arbitrary $k\text{-}1^{st}$ level clusters which belong to C_k. The traffic

exchanged between $C_{k-1}(i)$ and $C_{k-1}(j)$ is, according to the previous definition, a class k traffic. The number of nodes in a $k-1^{st}$ level cluster is equal to (see Fig. 2.5) $\prod_{i=1}^{k-1} n_i$. Hence, the volume of traffic from $C_{k-1}(i)$ to $C_{k-1}(j)$ is, for any (i,j), $\left(\prod_{i=1}^{k-1} n_i\right)^2 \gamma$.

The number of pairs of $k-1^{st}$ level clusters in a kth level cluster is $n_k(n_k - 1)$; also, the number of kth level clusters is $\prod_{i=k+1}^{m} n_i$. Consequently, the total amount of class k traffic is

$$\Gamma_k = \left(\prod_{i=k+1}^{m} n_i\right) n_k(n_k - 1)\left(\prod_{i=1}^{k-1} n_i\right)^2 \gamma \tag{6.69}$$

From the size constraint, Eq. (2.1), we know that $\prod_{i=1}^{m} n_i = N$. Substituting this last equation into Eq. (6.69), we find Eq. (6.68).

Notice that Eq. (6.68) checks with the fact that $\sum_{k=1}^{m} \Gamma_k = \Gamma$.

We are now ready to derive an expression of T in terms of the T_k's.

<u>Proposition 6.4</u>

Under the above assumptions, the total average delay in a hierarchical network is

$$T = \frac{1}{N - 1}\left[\left(N - \prod_{k=1}^{m-1} n_k\right)T_m + \sum_{k=1}^{m-1}\left(2N - (n_k + 1)\prod_{i=1}^{k-1} n_i\right)T_k\right] \tag{6.70}$$

<u>Proof</u>:

Substituting Eq. (6.67) into Eq. (6.66), we arrive at

$$T = \frac{1}{\Gamma}\sum_{k=1}^{m}\left(\Gamma_k + 2\sum_{i=k+1}^{m}\Gamma_i\right)T_k \tag{6.71}$$

Notice that Eq. (6.71) is a general expression of T in terms of T_k's which does not use the uniform traffic assumption. From Eq. (6.68),

$$\Gamma_k + 2 \sum_{i=k+1}^{m} \Gamma_i = N\gamma\left(2N - (n_k + 1) \prod_{i=1}^{k-1} n_i\right) \tag{6.72}$$

Substituting Eq. (6.72) into Eq. (6.71), we find Eq. (6.70).

Numerical example[1]:

$$N = 10^3, \qquad n_1 = n_2 = n_3 = 10, \qquad \beta = 1$$

$$T = 0.9T_3 + 1.82T_2 + 1.99T_1 \quad . \tag{6.73}$$

6.8 Conclusion

In this chapter we studied the major aspects related to the hierarchical design of large computer networks. The focus was primarily upon the determination of a certain clustering structure of the set of nodes to be used in the design phase. Optimal clustering structures were determined so as to minimize the computational cost required in the design phase. The general solution (i.e., different design strategies and gate assignment from one level to another) was derived when the number of hierarchical levels m is fixed. The global optimum solution was obtained with the more uniform case whereby the same design strategy and gate assignment are used at all levels. The global optimum solution is such that all degrees at all levels (except the first level) are equal (all layer subnets are of equal size). Such a peculiar property

[1] A similar expression as in Eq. (6.73) is given in [NAC 73]; It is $T = T_3 + 2T_2 + 2T_1$.

was found to have an intuitive explanation. Furthermore, we mention the one-to-one correspondence of the global optimum solution for $\alpha = K/(K - 1)$ with some regular trees of downward degree K.

Finally, we were able to decompose the average message delay in a hierarchical network in terms of the average delays in the layer subnets composing the network.

CHAPTER 7

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

7.1 Conclusions

Faced with the prohibitive cost of a simple extrapolation of present design and routing procedures for large networks, the goal of this dissertation was to evaluate some new techniques to be used in the context of large networks. The techniques studied here represent an extension of present schemes and rely mainly on the natural hierarchical clustering of the network nodes. More specifically we specified, evaluated and discussed the adaptive m-level Hierarchical Routing (MHR) schemes as well as the m-level Hierarchical Topology (MHT) design of large networks. The results obtained are summarized in Section 1.3. Basically, in this research we were able to show that:

1) Under some reasonable cost and performance constraints for a class of large distributed networks, present routing schemes become infeasible, whereas hierarchical routing schemes with optimally chosen table lengths maintain remarkably good network performance for a phenomenal range of network sizes.

2) Some optimal structures were found to minimize the computational cost involved in the hierarchical design of large networks.

3) Some extensions of existing queueing models for networks were introduced in order to accomodate nodal storage requirements and line overhead due to routing updates which become critical in the context of large nets. Furthermore, several buffer sharing schemes were proposed and analyzed in order to optimally utilize the nodal storage.

7.2 Future Research

While there exists an abundant literature on issues related to the design of small and moderate sized networks, the study of large networks is still in its very early stages. Consequently, we do not only inherit unsolved (or poorly solved) problems related to computer networks, but also those adequately solved need to be reevaluated in the context of large nets. Of importance are issues related to the flow control, reliability, security and distributed data bases. We also must not forget the long-standing search for exact solutions of the topology design problem and for more precise analytic network queueing models.

As for research areas which emerge directly from this work, we mention the important issue of clustering, i.e., assignment of nodes to clusters, clusters to superclusters, etc. We briefly addressed that question in Chapter 5, but further work is certainly required. Other clustering techniques (such as those based on a graph theoretic approach) must be investigated and experimented with. The same issue arises in the topology design of large nets where altogether different nearness measures and clustering techniques may be required. Moreover, the hierarchical design procedures should be compared to the non-hierarchical ones on some theoretic grounds, rather than through experimentation, using those algorithms which we here proved to be extremely expensive in the non-hierarchical case. We conjecture that some further limiting results of interest are possible.

Some new routing schemes and design methods should also be investigated.

Finally, it appears that the general methodology and decomposition models developed here for the study of large nets, may be directly applicable or extended to more general large systems where some sort of decomposition must be introduced to alleviate the difficulty in analysis, design and evaluation. The identification of such systems and their study represent a worthwhile investigation.

BIBLIOGRAPHY

ABRA 70 Abramson, N. "The ALOHA System--Another Alternative for Computer Communications," AFIPS Conference Proceedings, 37:281-285, FJCC, Las Vegas, Nevada, 1970.

ANDE 73 Anderberg, M.R. Cluster Analysis for Applications, Academic Press, New York, 1973.

BARA 64 Baran, P. "On Distributed Communications," Rand Corporation, Santa Monica, California, Rand Series Reports, August 1964.

BASK 75 Baskett, F., K. Chandy, R. Muntz, and F. Palacios. "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," Journal of the ACM, 22(2):248-260, April 1975.

BUZE 71 Buzen, J. Queueing Network Models of Multiprogramming, Ph.D. Dissertation, Division of Engineering and Applied Science, Harvard University, Cambridge, Massachusetts, 1971.

CARR 70 Carr, S. S. Crocker, and V. Cerf. "HOST-HOST Communication Protocol in the ARPA Network," AFIPS Conference Proceedings, 36:589-597, SJCC, Atlantic City, New Jersey, 1970.

CHAN 72 Chandy, K.M. "The Analysis and Solutions for General Queueing Networks," Proceedings Sixth Annual Princeton Conference on Information Sciences and Systems, Princeton University, Princeton, New Jersey, March 1972.

CHOW 74 Chow, C.K. "An Optimization of Storage Hierarchies," IBM Journal of Research and Development, 18(3):194-203, May 1974.

CHU 69 Chu, W.W. "A Study of Asynchronous Time Division Multiplexing for Time-Sharing Computer Systems," AFIPS Conference Proceedings, 35:669-678, FJCC, Las Vegas, Nevada, 1969.

CHUR 68 Churchman, C.W. The System Approach, Dell, New York, 1968.

CLOS 72A Closs, F. "Message Delays and Trunk Utilization in Line-Switched and Message-Switched Data Networks," Proceedings of the First USA-Japan Computer Conference, 524-530, 1972.

CLOS 72B Closs, F. "Time Delays and Trunk Capacity Requirements in Line-Switched and Message-Switched Networks," International Switching Symposium Record, Boston, Massachusetts, 1972, pp. 428-433.

COLE 71 Cole, G.C. "Computer Network Measurements: Techniques and Experiments," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7165, 1971.

COVI 74 Coviello, G.J. and R.D. Rosner. "Cost Considerations for a Large Data Network," ICCC Proceedings, 289-294, Stockholm, August 1974.

DANT 63 Dantzig, G.B. Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963.

DAVI 73 Davies, D.W. and D. Barber. Communication Networks for Computers, John Wiley, New York, 1973.

DRUK 75 Drukey, D.L. "Finite Buffers for Purists," TRW Incorporated, Redondo Beach, California, TRW Systems Group Report No. 75.6400-10-97, 1975.

DUDA 73 Duda, R.O. and P.E. Hart. Pattern Classification and Scene Analysis, John Wiley and Sons, New York, 1973.

EURO 73 "Eurodata--A Market Study on Data Communications in Europe, 1972-1985," study sponsored by the European Conference of Postal and Telecommunications Administrations, 1973.

FRAN 70 Frank, H, I.T. Frisch, and W. Chou. "Topological Considerations in the Design of the ARPA Network," AFIPS Conference Proceedings, 36:581-587, SCJJ, Atlantic City, New Jersey, 1970.

FRAN 71 Frank, H. and I.T. Frisch. Communication, Transmission, and Transportation Networks, Addison-Wesley, Reading, Massachusetts, 1971.

FRAN 72 Frank, H. and W. Chou, "Topological Optimization of Computer Networks," Proceedings of the IEEE, 60(11):1385-1397, November 1972.

FRAN 73 Frank, H., M. Gerla, and W. Chou. "Issues in the Design of Large Distributed Computer Communication Networks," Proceedings of the National Telecommunication Conference, 37A-1 to 37A-8, Atlanta, Georgia, November 1973.

FRAT 73 Fratta, L., M. Gerla, and L. Kleinrock. "The Flow Deviation Method--An Approach to Store-and-Forward Communication Network Design," Networks, 3:97-133, 1973.

FULT 72 Fultz, G.L. "Adaptive Routing Techniques for Message Switching Computer-Communication Networks," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7252, July 1972.

GEOF 74 Geoffrion, A.M. and G.W. Graves. "Multicommodity Distribution System Design by Benders Decomposition," Management Science, 20(5):822-844, January 1974.

GERL 73A Gerla, M. "The Design of Store-and-Forward (S/F) Networks for Computer Communications," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7319, January 1973.

GERL 73B Gerla, M., W. Chou, and H. Frank. "Computational Considerations and Routing Problems for Large Computer Communication Networks," Proceedings of the National Telecommunication Conference, 2:2B-1 to 2B-11, Atlanta, Georgia, November 1973.

GERL 73C Gerla, M. "Deterministic and Adaptive Routing Policies in Packet-Switched Computer Networks," Proceedings of the Third Data Communication Symposium, 23-28, St. Petersburg, Florida, November 1973.

GERL 74 Gerla, M. "New Line Tariffs and their Impact on Network Design," AFIPS Conference Proceedings, 43:577-582, NCC, Chicago, 1974.

GORD 67 Gordon, W.J., and G.F. Newell. "Closed Queueing Systems with Exponential Servers," Operations Research, 15:254-265, 1967.

GRAV 72 Graves, G.W. "Water Pollution Control," Western Management Science Institute, University of California, Los Angeles, Reprint No. 102, 1972.

HARA 72 Harary, F. Graph Theory, Addison-Wesley, Reading, Massachusetts, 1972.

HEAR 70 Heart, F., R. Kahn, S. Ornstein, W. Crowther, and D. Walden, "The Interface Message Processor for the ARPA Computer Network," AFIPS Conference Proceedings, 36:551-567, SJCC, Atlantic City, New Jersey, 1970.

HIMM 73 Himmelblau, D.M., ed. Decomposition of Large-Scale Problems, North-Holland, Amsterdam, 1973.

HU 69 Hu, T.C. Integer Programming and Network Flows, Addison-Wesley, Reading, Massachusetts, 1969.

ITOH 73 Itoh, K. and T. Kato. "An Analysis of Traffic Handling Capacity of Packet Switched and Circuit Switched Networks," Proceedings of the Third Data Communication Symposium, 29-37, St. Petersburg, Florida, November 1973.

JACK 57 Jackson, J.R. "Networks of Waiting Lines," Operations Research, 5:518-521, 1957.

JACK 63 Jackson, J.R. "Jobshop-Like Queueing Systems," Management Science, 10(1):131-142, January 1963.

KAHN 71 Kahn, R.E. and W.R. Crowther. "A Study of the ARPA Computer Network Design and Performance," Bolt Beranek and Newman, Inc., Cambridge, Massachusetts, Report No. 2161, August 1971.

KAHN 72 Kahn, R.E. "Resource-Sharing Computer Communications Networks," Proceedings of IEEE, 60(11):1397-1407, November 1972.

KAMO 76 Kamoun, F. and L. Kleinrock. "Analysis of Shared Storage in a Computer Network Environment," Proceedings of the Ninth HICSS, 89-92, Honolulu, Hawaii, January 1976.

KERN 70 Kernighan, B.W. and S. Lin. "An Efficient Heuristic Procedure for Partitioning Graphs," The Bell System Technical Journal, 291-307, February 1970.

KLEI 64 Kleinrock, L. Communication Nets; Stochastic Message Flow and Delay, McGraw-Hill, New York, 1964 (out of print). Reprinted by Dover Publications, 1972.

KLEI 70 Kleinrock, L. "Analytic and Simulation Methods in Computer Network Design," AFIPS Conference Proceedings, 36:569-579, SJCC, Atlantic City, New Jersey, 1970.

KLEI 74 Kleinrock, L. and W.E. Naylor. "On Measured Behavior of the ARPA Network," AFIPS Conference Proceedings, 43:767-780, NCC, Chicago, Illinois, 1974.

KLEI 75 Kleinrock, L. Queueing Systems, Vol. I: Theory, Wiley Inter-Science, New York, 1975.

KLEI 76 Kleinrock, L. Queueing Systems, Vol. II: Computer Applications, Wiley Interscience, New York, 1976.

KNUT 69 Knuth, D.E. The Art of Computer Programming, Vol. 1, Addison-Wesley, Reading, Massachusetts, 1969.

KUHN 73 Kühn, P. "Waiting Time Distributions in Multi-Queue Delay Systems with Gradings," ITC Proceedings, 7:242-1 to 242-9, Stockholm, 1973.

LAM 74 Lam, S. "Packet Switching in a Multi-Access Broadcast Channel with Applications to Satellite Communication in a Computer Network," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7249, March 1974.

LASD 70 Lasdon, L.S. Optimization Theory for Large Systems, Macmillan, New York, 1970.

MCCO 75 McCoy, C. Jr., "Improvements in Routing for Packet-Switched Networks," Naval Research Laboratory, Washington, D.C., NRL Report 7848, 1975.

MCQU 74 McQuillan, J.M. "Adaptive Routing Algorithms for Distributed Computer Networks," Bolt Beranek and Newman, Inc., Cambridge Massachusetts, Report No. 2831, May 1974.

MESA 70 Mesavoric, M., D. Macko, and Y. Takahara. Theory of Hierarchical Multilevel Systems, Academic Press, New York and London, 1970.

MIYA 75 Miyahara, H., T. Hasegawa, and Y. Teshigawara, "A Comparative Analysis of Switching Methods in Computer Communication Networks," Proceedings of the ICC, 6-6 to 6-10, San Francisco, California, June 1975.

MOOR 72 Moore, F.R. "Computational Model of a Closed Queueing Network with Exponential Servers," IBM Journal of Research and Development, 16(6):567-572, November 1972.

NAC 73 Network Analysis Corporation. "The Practical Impact of Recent Computer Advances on the Analysis and Design of Large Scale Networks," First Semiannual Technical Report, Glen Cove, New York, May 1973.

NAYL 75 Naylor, W.E. "A Loop-Free Adaptive Routing Algorithm for Packet Switched Networks," Proceedings of the Fourth Data Communications Symposium, 7-9 to 7-14, Quebec, Canada, October 1975.

PETE 73 Peters, R.A. and K.M. Simpson, "Eurodata: Data Communications in Europe 1972-1985," Datamation, 19(12):76-80, December 1973.

PORT 71 Port, E., and F. Closs. "Comparison of Switched Data Networks on the Basis of Waiting Times," IBM Zurich, Report RZ405, January 1971.

RICH 75 Rich, M.A., and M. Schwartz. "Buffer Sharing in Computer-Communications Network Nodes," Proceedings of the ICC, 3:33-17 to 33-21, San Francisco, California, June 1975.

ROBE 70 Roberts, L.G. and B.D. Wessler, "Computer Network Development to Achieve Resource Sharing," AFIPS Conference Proceedings, 36:543-549, SJCC, Atlantic City, New Jersey, 1970.

ROBE 74 Roberts, L.G. "Data by the Packet," IEEE Spectrum, 11(2):46-51, February 1974.

SCHO 71 Schoderbeck, P.P. Management Systems, John Wiley and Sons, New York, 1971.

SYSK 60 Syski, R. Introduction to Congestion in Telephone Systems, Oliver and Boyd, Edinburgh and London, 1960.

TOBA 74 Tobagi, F.A. "Random Access Techniques for Data Transmission Over Packet Switched Radio Networks," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7499, December 1974.

WILL 74 Williams, A.C. and R.A. Bhandiwad. "A Generating Function Approach to Queueing Network Analysis of Multiprogrammed Computers," Mobil Oil Corp., Princeton, New Jersey, April 1974.

WONG 75 Wong, J.W-N., "Queueing Network Models for Computer Systems," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7579, October 1975.

ZANG 69 Zangwill, W.I. Nonlinear Programming, A Unified Approach, Prentice-Hall, Englewood Cliffs, New Jersey, 1969.

ZEIG 71 Zeigler, J.F. "Nodal Blocking in Large Networks," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7167, 1971.

ZIMM 75 Zimmerman, H. "The Cyclades End to End Protocol," Proceedings of the Fourth Data Communications Symposium, 7-21 to 7-25, Quebec City, Canada, October 1975.

APPENDIX A

PATH LENGTH IN GRID AND TORUS NETWORKS

The purpose of this appendix is to derive closed form expression for the average path length in Grid-type networks. Also the distribution and the corresponding z-transform of path lengths in a Torus network, are defined and determined.

A.1 Definitions

Let (S, A) be a network composed of a set of nodes S and a set of arcs A. Let s, t be any pair of nodes in the network and π_{st} be any path between the two nodes [HARA 72], [HU 69]. Also let a_i represent the length associated with arc i, then the length of a path π_{st} is defined as,

$$\ell(\pi_{st}) = \sum_{i \varepsilon \pi_{st}} a_i \qquad (A.1)$$

The length of the shortest path from s to t, is

$$h_{st} \triangleq \min_{\text{all } \pi_{st}} \{\ell(\pi_{st})\}$$

h_{st} as defined above is a distance function [HARA 72], [FRAN 71]

If the length of all channels is assumed to be equal to 1, then h_{st} represents the minimum number of hops (channels) between s and t.

The average (shortest) path length in a network is defined as [KLEI 64],

$$h = \frac{1}{N(N-1)} \sum_{s,t \in S} h_{st} \tag{A.2}$$

where N is the size of the set S. In some instances it is of interest to consider the weighted average path length, h_w,

$$h_w = \frac{1}{\Gamma} \sum_{s,t \in S} \gamma_{st} h_{st} \tag{A.3}$$

where γ_{st} is the weight on the pair of nodes, s, t, and

$$\Gamma \triangleq \sum_{s,t \in S} \gamma_{s,t} \tag{A.4}$$

Notice that if all the γ_{st}'s are equal then Eq. (A.3) reduces to Eq. (A.2).

The <u>diameter</u>, d, of a network is defined as the longest shortest path in the network, i.e.,

$$d = \max_{s,t \in S} \{h_{st}\} \tag{A.5}$$

In the rest of this appendix we will restrict our considerations to the hop distance.

A.2 Average Path Length of a Grid

Let G be a rectangular $p \times q$ grid as shown in Fig. A.1. Let s of coordinates (x, y) and t of coordinates (u, v) be two arbitrary nodes in G. Then from Fig. A.1 we see that

$$h_{st} = |u - x| + |v - y| \tag{A.6}$$

Let $H_s(x, y)$ denote the distances (length of shortest path) from s to all other nodes in G,

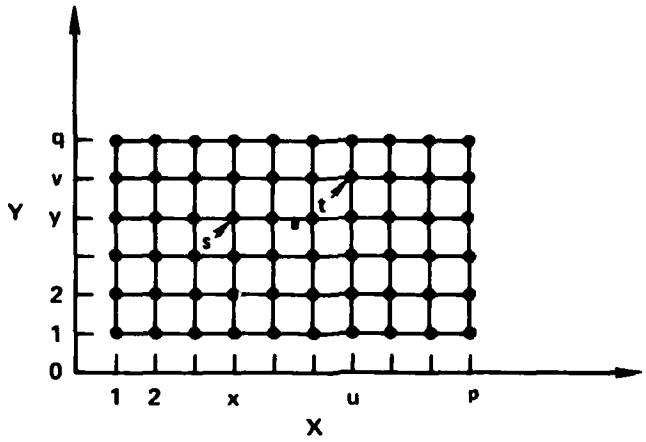


Figure A.1. p.q Grid

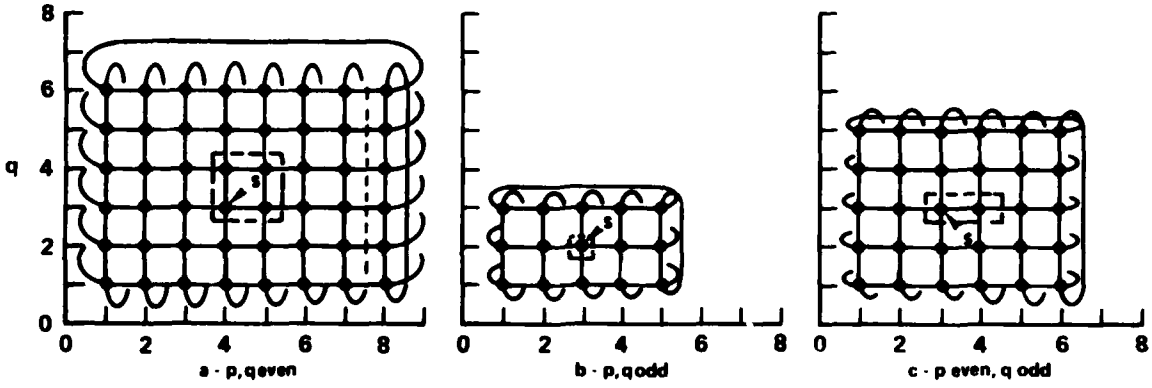


Figure A.2. Torus Nets.

$$H_s(x, y) \triangleq \sum_{t \varepsilon S} h_{st} \tag{A.7}$$

Substituting Eq. (A.6) in the above equation, we find

$$H_s(x, y) = \sum_{u=1}^{p} \sum_{v=1}^{q} (|u - x| + |v - y|) \tag{A.8}$$

Also $$H_s(x, y) = q \sum_{u=1}^{p} |u - x| + p \sum_{v=1}^{q} (v - y)$$

and

$$H_s(x, y) = q\left[x^2 - (p + 1)x + \frac{p(p + 1)}{2}\right] + p\left[y^2 - (q + 1)y + \frac{q(q + 1)}{2}\right] \tag{A.9}$$

Eq. (A.2) may be rewritten as,

$$h = \frac{1}{N(N - 1)} \sum_{s \varepsilon S} H_s(x, y) = \frac{1}{N(N - 1)} \sum_{x=1}^{p} \sum_{y=1}^{q} H_s(x, y) \tag{A.10}$$

Substituting Eq. (A.9) into the above summation, and after some algebra, we arrive at

$$\sum_{x=1}^{p} \sum_{y=1}^{q} H_s(x, y) = \frac{1}{3} q^2 p(p + 1)(p - 1) + \frac{1}{3} p^2 q(q + 1)(q - 1)$$

Also $$N(N - 1) = pq(pq - 1) \tag{A.11}$$

Hence $$h = \frac{p + q}{3} \tag{A.12}$$

From Fig. A.1, we can immediately see that the diameter of G is

$$d = p + q - 2 \tag{A.13}$$

For a <u>square grid</u> $p = q = \sqrt{N}$, hence

$$\begin{cases} h = \frac{2}{3}\sqrt{N} \\ d = 2\sqrt{N} - 2 \end{cases} \tag{A.14}$$

A.3 Average Path Length of a Torus

A torus is a grid where the outside nodes are connected together in a ring-like fashion, as shown in Fig. A.2. The purpose of the external rings is to introduce topological symmetry with respect to the nodes. In other words, all nodes are topologically equivalent. As a consequence, the distribution of path lengths from a given node to any other node is the same for all nodes. Hence the average path length simply becomes,

$$h = \frac{1}{N-1}\sum_{t\varepsilon S} h_{st} = \frac{1}{N-1} H_s(x, y) \tag{A.15}$$

where s is an arbitrary node of coordinate (x, y). In order to simplify the calculations we choose s as shown in Fig. A.2. This figure shows all such nodes s, for all cases except for p odd and q even, which is equivalent to p even and q odd. The addition (to a grid) of the external links does not affect the computation of $H_s(x, y)$ for such a choice of s; hence the result in Eq. (A.9) is true for those nodes. Also the diameter of the torus is equal to the distance from s to the upper right corner node. Several cases to consider are:

i. p, q even

$$h = \frac{1}{N-1} H_s[p/2, q/2]$$

Since $N = pq$ and H_s is given by Eq. (A.9), we get

$$\begin{cases} h = \dfrac{qp^2 + pq^2}{4(pq - 1)} \\ d = \dfrac{p + q}{2} \end{cases} \tag{A.16}$$

ii. p, q odd

$$h = \frac{1}{N - 1} H_s \left(\frac{p + 1}{2}, \frac{q + 1}{2}\right)$$

$$\begin{cases} h = \dfrac{q(p + 1)(p - 1) + p(q + 1)(q - 1)}{4(pq - 1)} = \dfrac{p + q}{4} \\ d = \dfrac{p + q - 2}{2} \end{cases} \tag{A.17}$$

iii. p even, q odd

$$h = \frac{1}{N - 1} H_s \left(\frac{p}{2}, \frac{q + 1}{2}\right)$$

$$\begin{cases} h = \dfrac{qp^2 + p(q + 1)(q - 1)}{4(pq - 1)} \\ d = \dfrac{p + q - 1}{2} \end{cases} \tag{A.18}$$

iv. p odd, q even

$$h = \frac{1}{N - 1} H_s \left(\frac{p + 1}{2}, \frac{q}{2}\right)$$

$$\begin{cases} h = \dfrac{q(p + 1)(p - 1) + pq^2}{4(pq - 1)} \\ d = \dfrac{p + q - 1}{2} \end{cases} \tag{A.19}$$

<u>Square Torus</u>

For a square torus $p = q = \sqrt{N}$

i. p even

$$\begin{cases} h = \dfrac{p^3}{2(p^2 - 1)} = \dfrac{N^{3/2}}{2(N-1)} \\ d = p = \sqrt{N} \end{cases} \tag{A.20}$$

ii. p odd

$$\begin{cases} h = \dfrac{p}{2} = \dfrac{\sqrt{N}}{2} \\ d = p - 1 = \sqrt{N} - 1 \end{cases} \tag{A.21}$$

Notice that for a large N Eqs. (A.20) and (A.21) are equivalent. Hence in the study of large tori, we will not differentiate between the two cases above.

A. 4 Distribution of Path Lengths in a Square Torus

Let h_s be a discrete random variable which represents the distance (length of shortest path) from an arbitrary node s to any other node in the network.

The probability distribution of $\underset{\sim}{h}_s$ is defined as,

$$P_r[\underset{\sim}{h}_s = k] = \frac{\text{number of nodes at a distance k from s}}{N - 1} \tag{A.22}$$

The corresponding z-transform is defined as,

$$H_s(z) = \sum_k z^k P_r[\underset{\sim}{h}_s = k] \tag{A.23}$$

Similarly, let $\underset{\sim}{h}$ be the random variable which represents distances in the network. The probability distribution of $\underset{\sim}{h}$ is defined as

$$P_r[\underset{\sim}{h} = k] = \frac{\text{number of pairs of nodes separated by a distance equal to } k}{N(N-1)} \tag{A.24}$$

Also the corresponding z transform is

$$H(z) = \sum_k z^k P_r[\underset{\sim}{h} = k] \tag{A.25}$$

Evaluation of H(z) for a square torus

Since a torus is symmetrical with respect to any node, then,

$$\underset{\sim}{h}_s = \underset{\sim}{h} \Rightarrow H_s(z) = H(z) \qquad \forall\, s \in S. \tag{A.26}$$

Consequently we will evaluate the transform at a particular node s. When $p = \sqrt{N}$ is odd, the evaluation of $H_s(z)$ is straightforward at our previous node s. Fig. A.3 shows node s and contours of nodes at equal distance from s.

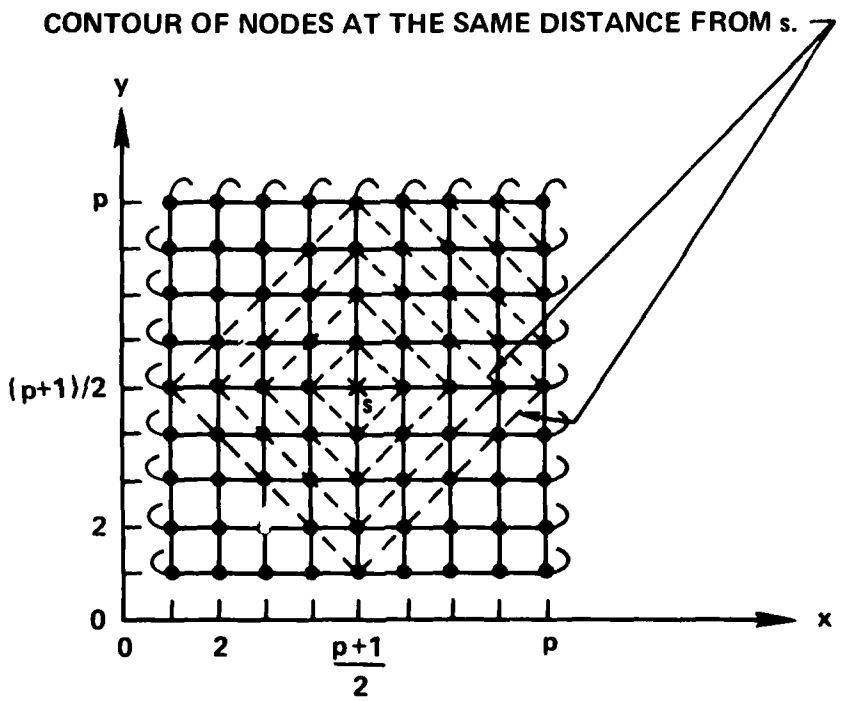


Figure A.3. Square Torus.

If we define N_k as the number of nodes at distance k from s, then

$$N_k = \begin{cases} 4k & 1 \leq k \leq \frac{p-1}{2} \\ 4(p-k) & \frac{p+1}{2} \leq k \leq p-1 \end{cases} \tag{A.27}$$

Note that
$$\sum_{k=1}^{p-1} N_k = p^2 - 1 = N - 1 \tag{A.28}$$

From the definition of the distribution of path length, Eq. (A.22)

$$P_r[\underset{\sim}{h} = k] = \begin{cases} \frac{N_k}{N-1} & 1 \leq k \leq p-1 \\ 0 & \text{otherwise} \end{cases} \tag{A.29}$$

Substituting Eq. (A.29) into Eq. (A.25), we arrive at

$$H(z) = \sum_{k=1}^{p-1} \frac{N_k}{N-1} z^k \tag{A.30}$$

and from Eq. (A.27),

$$H(z) = \frac{4}{N-1}\left[\sum_{k=1}^{(p-1)/2} kz^k + \sum_{k=(p+1)/2}^{p-1} (p-k)z^k\right]$$

To evaluate the above summations we need the identity below

$$\sum_{k=1}^{n} kz^k = \frac{z}{(1-z)^2}(1 + nz^{n+1} - (n+1)z^n)$$

Then after some algebra, we find

$$H(z) = \frac{4z\left(1 - z^{(p-1)/2}\right)\left(1 - z^{(p+1)/2}\right)}{(N-1)(1-z)^2} \tag{A.31}$$

Remarks

i. From the definition of the z-transform (Eq. (A.23)) $H(z)$ must be such that $H(1) = 1$. This fact can be easily checked using l'Hopital's rule.

ii. $H'(1)$ represents the average of the random variable $\underset{\sim}{h}$. It corresponds to the actual average path length (in hops) in the network. Hence, from Eq. (A.21), $H(z)$ must be such that

$$H'(1) = h = p/2 \tag{A.32}$$

Differentiating Eq. (A.31) with respect to z, we find

$$H'(z) = \frac{4}{(N-1)(1-z)^3}\left[1 + z - (p-1)z^{p+1} + (p+1)z^p - 3z^{(p+1)/2} + \frac{p-1}{2}z^{(p+3)/2} - \frac{p+1}{2}z^{(p-1)/2}\right]$$

The application of l'Hopital's rule to $H'(z)$ at $z = 1$, shows that effectively Eq. (A.32) holds true.

iii. For p even, Eq. (A.31) is no longer true. However, for a large N (as is the case in Chapter 4) we may confidently use Eq. (A.31) in both cases.

APPENDIX B

ANALYSIS OF SHARED STORAGE IN A COMPUTER NETWORK ENVIRONMENT

Our earlier considerations (Chapter 4) lead us to model the store-and-forward (S/F) function of a node, as R M|M|1 queueing systems which share a finite waiting room, under some scheme (see Fig. 4.12). The purpose of this appendix is to analyze and compare a few existing and/or intuitive schemes as well as others motivated by the limitations (or deficiencies) of the former.

The first (and simplest) scheme is the Complete Partitioning (CP) scheme where actually no sharing is provided, but where the entire storage (waiting room) is partitioned among the R servers (see Fig. B.1.a). At the other extreme is the second scheme, Complete Sharing (CS), which is such that an arriving customer is accepted if space is available, regardless of its class, i.e., independently of the server to which it is directed (see Fig. B.1.b). CS succeeds in achieving a better performance than CP (smaller probability of blocking) under normal traffic conditions and for fairly balanced input systems. However, for quite asymetrical input rates[1] (λ_i $i = 1, \ldots, R$), CS tends to heavily favor servers with higher input rates, even though they may be close to saturation (input rate equal or almost equal to service rate). The failure to recognize servers at or near saturation results in most of the space being occupied by customers waiting for those

[1]This remark assumes that all servers have equal service rates.

servers, at the detriment of the others. Moreover, even with perfectly balanced arrival rates (i.e., $\lambda_i = \lambda \quad i = 1, \ldots, R$), under heavy traffic conditions ($\lambda >> \mu C$), CS fails (where CP succeeds) in securing a full utilization of all the R servers.

The above considerations intuitively indicate that contention for space must be limited in some ways. In order to avoid the possible utilization of the entire space by any particular type(s) of customers, we impose a limit on the number of buffers to be allocated at any time, to any server. This idea is incorporated in our third scheme: Sharing with Maximum Queues (SMXQ), Fig. B.1.c. Of course, the sum of those maxima must be greater than the total space if some sharing is to be provided. The SMXQ, however, does not guarantee a full utilization of the servers under heavy traffic conditions. This deficiency motivates the fourth scheme: Sharing with a Minimum Allocation (SMA) scheme. With SMA, a minimum number of buffers is always reserved for each server and, in addition, a common pool of buffers is to be shared among all the servers, with no constraints on the queue size (see Fig. 3.1.d).

The shared area is again prone to be, as mentioned earlier, unfairly utilized, and hence the fifth and final scheme: Sharing with a Maximum Queue and a Minimum Allocation (SMQMA)(see Fig. B.1.e).

Rich and Schwartz [RICH 75] studied a scheme very similar to SMA except that the <u>entire</u> common storage is dynamically allocated to one server at a time. Moreover, problems of this sort are frequently encountered in telephony and are referred to as graded systems [KUHN 73]. [SYSK 60]. Their main interest, however, is in sharing (extra) lines as opposed to storage.

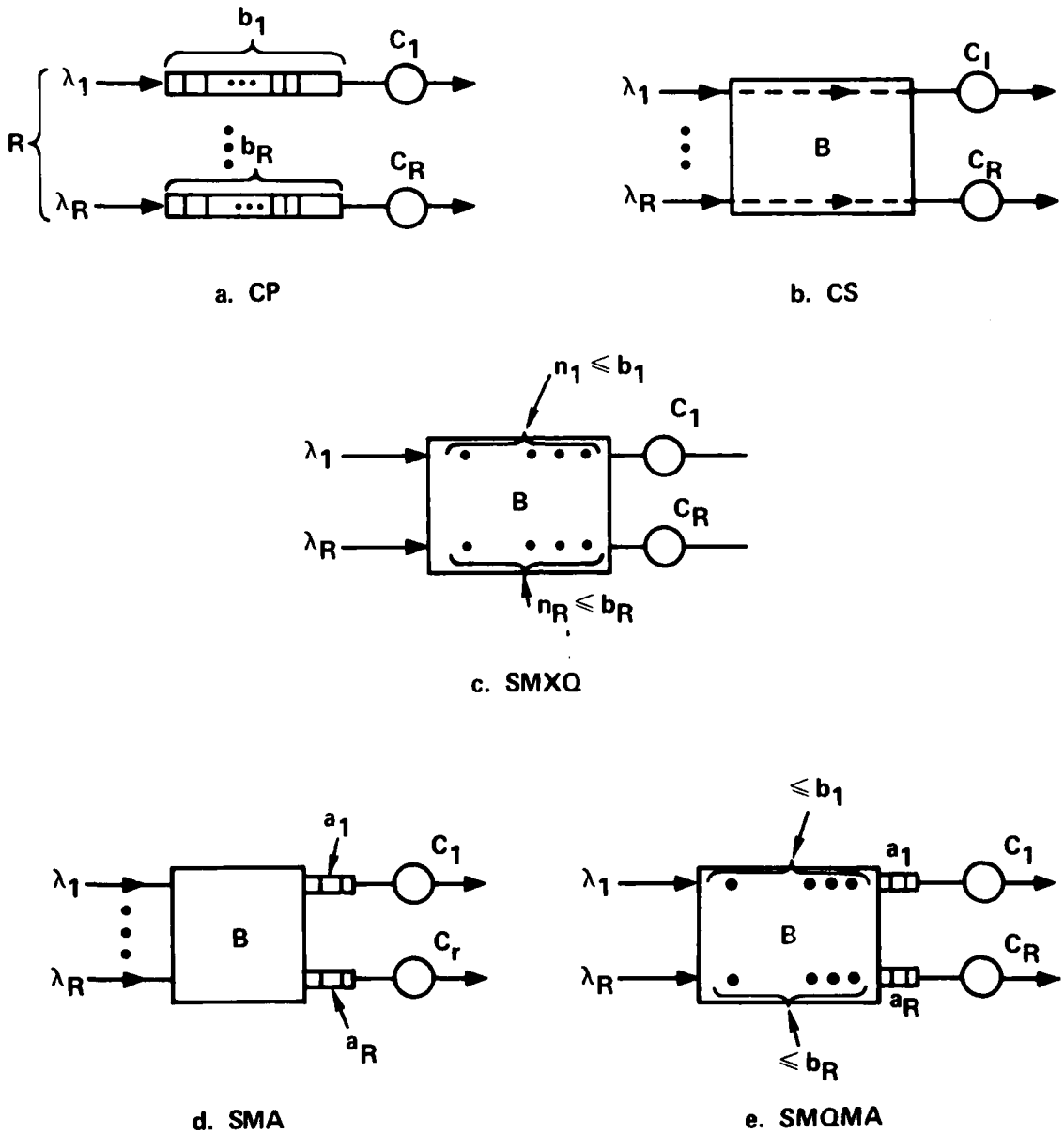


Figure B.1. Storage Sharing Schemes.

In this appendix, we intend to characterize the five schemes under steady state conditions; namely, we derive expressions for the probabilities of blocking, the joint and marginal distributions, the average time in system and the throughput for different types of customers. The key to the analysis lies in the fact that, in steady state, the joint probability distribution obeys the well known product form solution for networks of queues [JACK 57, 63], [GORD 67], [BASK 75], [BUZE 71], [WONG 75], [KLEI 75, 76] (and the bibliographies therein).

A comparison of the sharing schemes is also provided.

B.1 Complete Partitioning (CP)

With CP, we are in the presence of R separate and independent M|M|1 queueing systems with finite waiting rooms (Fig. B.1.a). Queueing system i (i = 1, ..., R) is characterized by a Poisson input stream of rate λ_i, an exponential service time of mean $1/\mu C_i$ and a finite storage of size b_i. Customers destined to server i are referred to as type or class i customers. Arriving customers who find no space in their respective queues depart without service. Accepted (non rejected) customers are served on a First-Come-First-Serve (FCFS) basis.

The basic equations describing the behavior of any of the queues (say subsystem i) are well known (see for example, [KLEI 75]). Let n_i be the number of customers in queueing system i (i.e., type-i customers) and $Pr[n_i = k]$ be the probability, in steady state, of having k type-i customers (in queue and server), then

$$P_r[n_i = k] = \begin{cases} \dfrac{1 - \rho_i}{1 - \rho_i^{b_i+1}} \rho_i^k & 0 \le k \le b_i \\ 0 & \text{otherwise} \end{cases} \tag{B.1}$$

Where $\rho_i = \lambda_i/\mu C_i$.

Also, let PB_i denote the probability of blocking for type-i customers, then [KLEI 75]

$$PB_i = \frac{1 - \rho_i}{1 - \rho_i^{b_i+1}} \rho_i^{b_i} \tag{B.2}$$

Let $\bar{n}_i$ be the average number of type-i customers; then

$$\bar{n}_i = \sum_{k=1}^{b_i} k \; P_r[n_i = k] = \frac{\rho_i}{1 - \rho_i} \; \frac{1 - (1 + b_i)\rho_i^{b_i} + b_i\rho_i^{1+b_i}}{1 - \rho_i^{1+b_i}} \tag{B.3}$$

Let λ_i' be the average rate of non-rejected type-i customers, i.e., the throughput of server i; then

$$\lambda_i' = (1 - PB_i)\lambda_i \tag{B.4}$$

Also if t_i denotes the average time in system (queue and server) of non-blocked type-i customer, then from Little's result

$$t_i = \bar{n}_i/\lambda_i' \tag{B.5}$$

Thus,

$$t_i = \frac{1/\mu C_i}{1 - \rho_i} \; \frac{1 - (1 + b_i)\rho_i^{b_i} + b_i\rho_i^{1+b_i}}{1 - \rho_i^{b_i}} \tag{B.6}$$

The state of the entire system can be described by the vector $\underset{\sim}{n} = (n_1, n_2, \ldots, n_R)$; since all the subsystems are independent, the joint probability distribution in steady state (to be denoted by $P(\underset{\sim}{n})$ or $P(n_1, n_2, \ldots, n_R)$) is

$$P(n_1, n_2, \ldots, n_R) = \prod_{i=1}^{R} P_{n_i} = \left(\prod_{i=1}^{R} \frac{1 - \rho_i}{1 - \rho_i^{1+b_i}} \right) \prod_{i=1}^{R} \rho_i^{n_i} \qquad \forall\, n \in F_a$$

$$= 0 \qquad \text{otherwise} \tag{B.7}$$

Where the set F_a is the set of feasible vectors,

$$F_a = \{\underset{\sim}{n} \text{ integer valued vector, s.t. } 0 \leq n_i \leq b_i,\ i = 1, \ldots, R\} \tag{B.8}$$

The subscript "a" is used here to remind us of scheme (a), i.e., CP.

Notice that the constant term in Eq. (B.7) is, of course, the probability of the entire system being empty, i.e., $P(0, \ldots, 0)$. That term will be denoted in the sequel as P_0.

This terminates the analysis of CP in its general form. A few more remarks are presented below.

i. $b_i \to \infty$ and $\rho_i < 1$; we obtain the usual $M|M|1$ result

$$P_r[n_i = k] = (1 - \rho_i)\rho_i^k \tag{B.9}$$

ii. $\rho_i \to \infty$; from Eqs. (B.1) and (B.2), we get

$$\begin{cases} PB_i \to 1 \\ \overline{n_i} \to b_i \\ \lambda_i' \to \mu C_i \\ t_i \to b_i \mu C_i \end{cases} \tag{B.10}$$

iii. $\rho_i = 1$; all states $n_i = 0, 1, \ldots, b_i$ are equally probable and

$$PB_i = \frac{1}{1 + b_i}$$
$$\bar{n}_i = \frac{b_i}{2} \tag{B.11}$$

We now proceed with the analysis of CS. The definitions and notation introduced in this section are valid throughout this appendix.

B.2 Complete Sharing (CS)

We now combine all the individual waiting rooms in a global one (see Fig. B.1.b) whose size will be denoted by B. Empty space is allocated on a FCFS basis regardless of the type of arriving customer. Drukey [DRUK 75] analyzed this system with the assumption that all the ρ_i's ($\rho_i = \lambda_i/\mu C_i$) are equal; for the general case of different ρ_i's, he restricts his study to two classes of customer (R = 2). In what follows we intend to analyze the more general case of arbitrary ρ_i's $i = 1, \ldots, R$, and then we apply our results to the special case of equal ρ_i's. The forms of the expressions in common with Drukey's differ because of the network of queues approach used here.

B.2.1 General Case

In this section, the ρ_i's are not necessarily equal. The sharing of space introduces dependencies among the R queueing systems. The entire system is a birth-death process whose state can be simply described by the vector $\underset{\sim}{n} = (n_1, \ldots, n_R)$. From the state-transition-rate diagram of Fig. B.2, we can write the set of balance equations below, describing the behavior of the system in steady state (steady

state is always reached, even for $\rho_i > 1$, because of the finite number of states in our ergodic Markov chain).

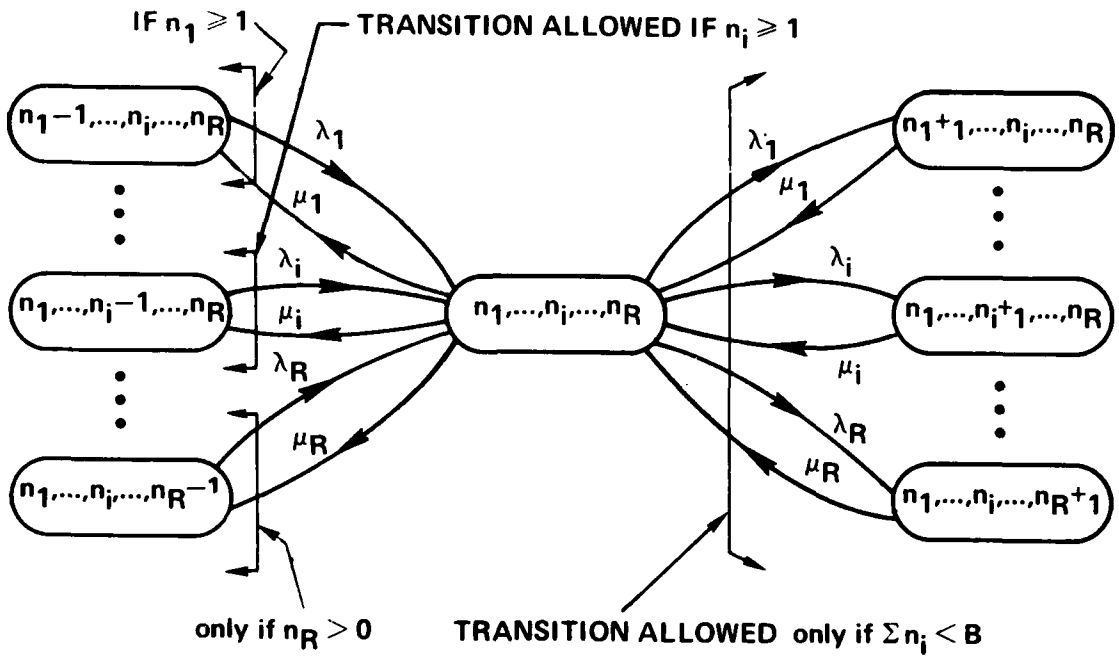


Figure B.2. State-Transition-Rate Diagram.

Let $J = \{j \text{ such that } n_j = 0\}$; two sets of balance equations are presented below which depend on whether or not the total number of customers is less than or equal to B.

i. For all $\underset{\sim}{n}$'s, such that $\Sigma_i n_i < B$, we have

$$\sum_{\substack{i=1 \\ i \notin J}}^{R} \lambda_i P(n_1, \ldots, n_i-1, \ldots, n_R) + \sum_{i=1}^{R} \mu_i P(n_1, \ldots, n_i+1, \ldots, n_R)$$

$$= \left(\sum_{i=1}^{R} \lambda_i + \sum_{\substack{i=1 \\ i \notin J}}^{R} \mu_i\right) P(n_1, \ldots, n_i, \ldots, n_R)$$

ii. For all $\underset{\sim}{n}$'s, such that $\Sigma_i n_i = B$,

$$\sum_{\substack{i=1 \\ i \notin J}}^{R} \lambda_i P(n_1, \ldots, n_i-1, \ldots, n_R) = \left(\sum_{\substack{i=1 \\ i \notin J}} \mu_i\right) P(n_1, \ldots, n_i, \ldots, n_R)$$

Let F_b be the state of feasible states, i.e.,

$$F_b = \{\underset{\sim}{n}, \text{ s.t. } 0 \leq n_i \leq B \quad , \quad 0 \leq \sum_{i=1}^{R} n_i \leq B\}$$

Then, we claim that the steady state joint queue length (queue + server) distribution is such that

$$P(\underset{\sim}{n}) = \begin{cases} P_0 \rho_1^{n_1} \rho_2^{n_2} \ldots \rho_R^{n_R} & \forall \underset{\sim}{n} \in F_b \\ 0 & \text{otherwise} \end{cases} \tag{B.12}$$

where P_0 is a constant; more precisely, it is the probability of an empty system $P(0, 0, \ldots, 0)$.

To prove this claim it is sufficient to show that Eq. (B.12) satisfies the above balance equations; this is easily shown to be the case. Notice also that the product form solution satisfies the local balance equations. In the present situation, the local balance equation [CHAN 72], [KLEI 75] is the one which equates the rate of flow out of server i, due to the departure of customer (type-i), to the rate of flow in, due to the arrival of a customer, to server i. This approach will be used to analyze the other schemes whose global balance equations are more difficult to write.

We are now left with the evaluation of P_0, for that we take advantage of the fact that all probabilities must sum to one, hence

$$P_0^{-1} = \sum_{\underset{\sim}{n} \varepsilon F_b} \rho_1^{n_1} \rho_2^{n_2} \ldots \rho_R^{n_R} \tag{B.13}$$

Let us define G(K) as

$$G(K) = \sum_{\substack{a \le n_i \le K \\ \Sigma_i n_i = K}} \rho_1^{n_1} \ldots \rho_R^{n_R} \tag{B.14}$$

Then, obviously

$$P_0^{-1} = \sum_{K=0}^{B} G(K) \tag{B.15}$$

Several efficient altorithms exist to compute G(K), [BUZE 71], [WILL 74] [MOOR 72]. Moreover, if all the ρ_i's are different, we can use the generating function approach to derive a closed form expression for G(K), [MOOR 72]. Briefly, we let

$$g(t) = \prod_{i=1}^{R} \frac{1}{(1 - \rho_i t)} = \prod_{i=1}^{R} (1 + \rho_i t + \rho_i^2 t^2 + \cdots) \tag{B.16}$$

By expanding the second product we recognize that G(K) is the coefficient of t^K, i.e.

$$g(t) = 1 + G(1)t + G(2)t^2 + \cdots + G(K)t^K + \cdots \tag{B.17}$$

Also, if the ρ_i's are different[1] ($\rho_i \neq \rho_j$), then the partial fraction expression of the first product in Eq. (B.16) gives

$$\begin{cases} \displaystyle\prod_{i=1}^{R} \frac{1}{(1 - \rho_i t)} = \sum_{i=1}^{R} \frac{A_i}{1 - \rho_i t} \\ \text{where} \quad \displaystyle A_i = \prod_{\substack{k=1 \\ k \neq i}}^{R} \frac{1}{(1 - \rho_k/\rho_i)} \end{cases} \tag{B.18}$$

[1]This condition will be assumed to be true throughout the rest of this section.

Hence

$$G(K) = \sum_{i=1}^{R} A_i \rho_i^K \qquad K = 0, 1, 2, \ldots \tag{B.19}$$

Then, from Eqs. (B.13), (B.14) and (B.19)

$$P_0^{-1} = \sum_{K=0}^{B} \sum_{i=1}^{R} A_i \rho_i^K = \sum_{i=1}^{R} A_i \sum_{K=0}^{B} \rho_i^K$$

and

$$P_0^{-1} = \sum_{i=1}^{R} A_i \frac{1 - \rho_i^{B+1}}{1 - \rho_i} \tag{B.20}$$

Eqs. (B.12) and (B.20) completely characterize our system in the steady state. Notice that for $R = 1$, $A_1 = 1$ and we find Eq. (B.1) for $B = b_i$. We now proceed to derive the distributions in steady state or expressions of other variables of interest.

<u>Distribution of the total number in system; Probability of Blocking</u>

Let n be that number and $P_n = P_r[\sum n_i = n]$ be the corresponding distribution; then obviously

$$\begin{cases} P_n = P_0 G(n) = \dfrac{G(n)}{\sum_{k=0}^{n} G(k)} & \text{for } 0 \leq n \leq B \\ = 0 & \text{otherwise} \end{cases} \tag{B.21}$$

Since we have Poisson arrivals, then the probability of blocking, PB, is simply the probability that $n = B$. Notice also, that PB is independent of the customer's type ($PB_i = PB \quad \forall \quad i = 1, \ldots, R$)

$$PB = P_0 G(B) \tag{B.22}$$

Marginal Distributions and Averages

Let us first derive the probability that there are more than j type i customers in the system, $P_r[n_i \geq j]$. That probability is equal to the sum of $P(\underset{\sim}{n})$ for $\underset{\sim}{n} \in F_b$ and such that $n_i \geq j$; after some algebra we find (for $0 \leq j \leq B$),

$$P_r[n_i \geq j] = P_0 \rho_i^j \sum_{n=0}^{B-j} G(n) \tag{B.23}$$

The average number of type i customers is

$$\bar{n}_i = \sum_{j=1}^{B} P_r[n_i \geq j] = P_0 \sum_{j=1}^{B} \rho_i^j \sum_{n=0}^{B-j} G(n)$$

Interchanging the summations above, we arrive at

$$\bar{n}_i = P_0 \sum_{n=0}^{B-1} G(n) \sum_{j=1}^{B-n} \rho_i^j = P_0 \sum_{n=0}^{B-1} \frac{\rho_i - \rho_i^{B+1-n}}{1 - \rho_i} G(n)$$

hence

$$\overline{n_i} = \frac{\rho_i}{1 - \rho_i} \frac{\sum_{n=0}^{B-1} \left(1 - \rho_i^{B-n}\right) G(n)}{\sum_{n=0}^{B} G(n)} \tag{B.24}$$

As for the expression of the average total number in system, it is simply $\bar{n} = \Sigma_i \bar{n}_i$. Another expression can be derived from Eq. (B.21),

$$\bar{n} = P_0 \sum_{k=1}^{B} kG(k)$$

From Eq. (B.19)

$$\bar{n} = P_0 \sum_{k=0}^{B} k \sum_{i=1}^{R} A_i \rho_i^k = P_0 \sum_{i=1}^{R} A_i \sum_{k=0}^{B} k\rho_i^k$$

Thus

$$\bar{n} = P_0 \sum_{i=1}^{R} A_i \frac{1 + B\rho_i^{1+B} - (B + 1)\rho_i^B}{(1 - \rho_i)^2} \tag{B.25}$$

where P_0 is given by Eq. (B.20).

Notice that if $R = 1$, then $\bar{n} = \bar{n}_1$, as found in Eq. (B.3). Eqs. (B.4) and (B.5), for the throughput λ_i and the average delay t_i, hold true here.

This terminates the characterization of CS in its more general case. Of interest is the study of its behavior under some special limiting conditions of storage and traffic.

B.2.2 Limiting Behavior

We consider the two cases; first, when B goes to infinity, and second, when all arrival rates increase uniformly toward infinity.

Infinite waiting room $B \to \infty$

For the existence of a steady state, it is necessary that $\rho_i < 1$ $(i = 1, \ldots, R)$. With this condition, we expect that at $B = \infty$ the system becomes equivalent to R independent $M|M|1$ queues. From Eq. (B.20) and since $\rho_i < 1$,

$$\lim_{B \to \infty} P_0^{-1} = \sum_{i=1}^{R} \frac{A_i}{1 - \rho_i}$$

which is the value of $g(t)$ (Eqs. (B.16) and (B.18)) at $t = 1$, hence,

$$P_0 = \prod_{i=1}^{R} (1 - \rho_i)$$

Thus from Eq. (B.12) $\quad P(\underset{\sim}{n}) = \prod_{i=1}^{R} (1 - \rho_i)\rho_i^{n_i}$

Moreover, from Eq. (B.23) we can see that at the limit

$$P_r(n_i \geq j) = \rho_i^j \Rightarrow P_r(n_i = j) = (1 - \rho_i)\rho_i^j$$

The last three equations prove the fact that the system is equivalent to R independent M|M|1 queues.

Note also, that from Eq. (B.19) and (B.22)

$$\lim_{B \to \infty} G(B) = 0 \quad \Rightarrow \quad \lim_{B \to \infty} PB = 0$$

Uniform increase of input rates

We now intend to let all input rates increase proportionally until eventually reaching infinity. At this point we show that our system becomes equivalent to a closed network of R queues and B customers. Let

$$\lambda_i = \eta\lambda_i^0 \qquad i = 1, \ldots, R \tag{B.26}$$

where η is a positive real variable and λ_i^0 is a constant. The service rates (μC_i) are maintained constant. Eq. (B.26) is also equivalent to saying that $\rho_i = \eta\rho_i^0$ with ρ_i^0 constant.

From the above definition and Eq. (B.15),

$$G(K) = \eta^k G^0(K) \tag{B.27}$$

where $G^0(K) = \sum_{i=1}^{R} A_i(\rho_i^0)^K$.

Notice that A_i is invariant; hence, $G^0(K)$ is a constant. As a result, and from Eq. (B.15)

$$P_0^{-1} = \sum_{K=0}^{B} \eta^K G^0(K) \quad \Rightarrow \quad \lim_{\eta \to \infty} P_0 = 0$$

Furthermore, from Eq. (B.21)

$$\lim_{\eta \to \infty} P_n = \begin{cases} 0 & n \neq B \\ 1 & n = B \end{cases}$$

In other words, as expected, the system is full with Probability 1 (probability of blocking is consequently equal to 1).

With regard to the joint queue length distribution, Eq. (B.12) becomes,

$$P(\underset{\sim}{n}) = \frac{\eta^{\sum_{i=1}^{R} n_i}}{\sum_{k=0}^{B} \eta^k G^0(k)} \prod_{i=1}^{R} \left(\rho_i^0\right)^{n_i} \qquad \forall\, \underset{\sim}{n} \in F_b$$

Thus

$$\lim_{\eta \to \infty} P(\underset{\sim}{n}) = \begin{cases} \dfrac{\prod_{i=1}^{R} \left(\rho_i^0\right)^{n_i}}{G^0(B)} & \text{for } \underset{\sim}{n} \in F_b \text{ and s.t. } \Sigma_i n_i = B \\ 0 & \text{otherwise} \end{cases} \tag{B.28}$$

With respect to the marginal distributions from Eq. (B.24),

$$\lim_{\eta \to \infty} P_r[n_i \geq j] = \frac{G^0(B - j)}{G^0(B)} \left(\rho_i^0\right)^j \tag{B.29}$$

Furthermore, the limiting throughput of type-i customers is

$$\lim_{\eta \to \infty} (1 - PB)\eta\lambda_i^0 = \frac{G^0(B - 1)}{G^0(B)} \lambda_i^0 \tag{B.30}$$

The above results are typical of a closed network of queues with R service centers and B customers, such as Buzen's central server model [BUZE 71]. In fact, if we consider the buffers as customers and if we let $\lambda_i/\Sigma\lambda_i$, i.e., $\lambda_i^0/\Sigma\lambda_i^0$, be the relative rate of arrivals to server i (i.e., rate of allocations of freed buffers to server i), and n_i be the number of buffers allocated to server i, then the distribution of buffers is given by the above expressions (Eqs. (B.28) and (B.29)).

Moreover, Eq. (B.29) shows that there is a non-zero probability that server i is idle $\left(P_r[n_i = o] = 1 - G^0(B - 1)\rho_i^0/G^0(B)\right)$. Therefore, even with infinite input rates, server i is not fully utilized, which was not the case with CP (see Eq. (B.10)).

The numerical example below illustrates the general and limiting behavior of this system with respect to η. In this example we assume that $R = 4$, $B = 20$, $\rho_1^0 = 0.1$, $\rho_2^0 = 0.4$, $\rho_3^0 = 0.6$, $\rho_4^0 = 0.9$ and we let $\rho_i = \eta\rho_i^0$.

The utilization of server i is $\rho_i' = (1 - PB)\rho_i = (1 - PB)\eta\rho_i^0$. The limiting value of $\eta(1 - PB) = 1.111$; hence, $\eta \to \infty \Rightarrow \rho_1' = 0.1111$ $\rho_2' = 0.4444$, $\rho_3' = 0.6666$, $\rho_4' = 0.9999$. Notice that server 4 reached saturation whereas the others are very far from it. The average total limiting utilization is $\bar{\rho} = \frac{1}{R}\sum \rho_i' = 0.555$ instead of 1 which could be obtained with CP.

Furthermore, Fig. B.3 shows the behavior of the average number of type-i customers in the system $(i = 1, \ldots, 4)$ with respect to η. Also represented, is the average total number in system, $\bar{n}$. The limiting values for the averages are (obtained at $\eta = 20$)

$\bar{n}_1 \to 0.125$, $\bar{n}_2 \to 0.800$, $\bar{n}_3 \to 1.99$, $\bar{n}_4 \to 17.02$ and $\bar{n} \to 19.94 \approx B$.

Notice that for large η, most of the buffers are, on the average, used by type 4 customers. Also, a sharp increase of $\bar{n}_4$ (from 4.8 to 14) occurs when η varies from 0.95 to 1.5. The value of $\eta = 1.111$ would correspond to the saturation of server 4 (i.e., $\rho_4 \approx 1$) if there were no limitation in buffer storage, and at that point the queue size becomes infinite. This explains the sharp increase in $\bar{n}_4$, mentioned above.

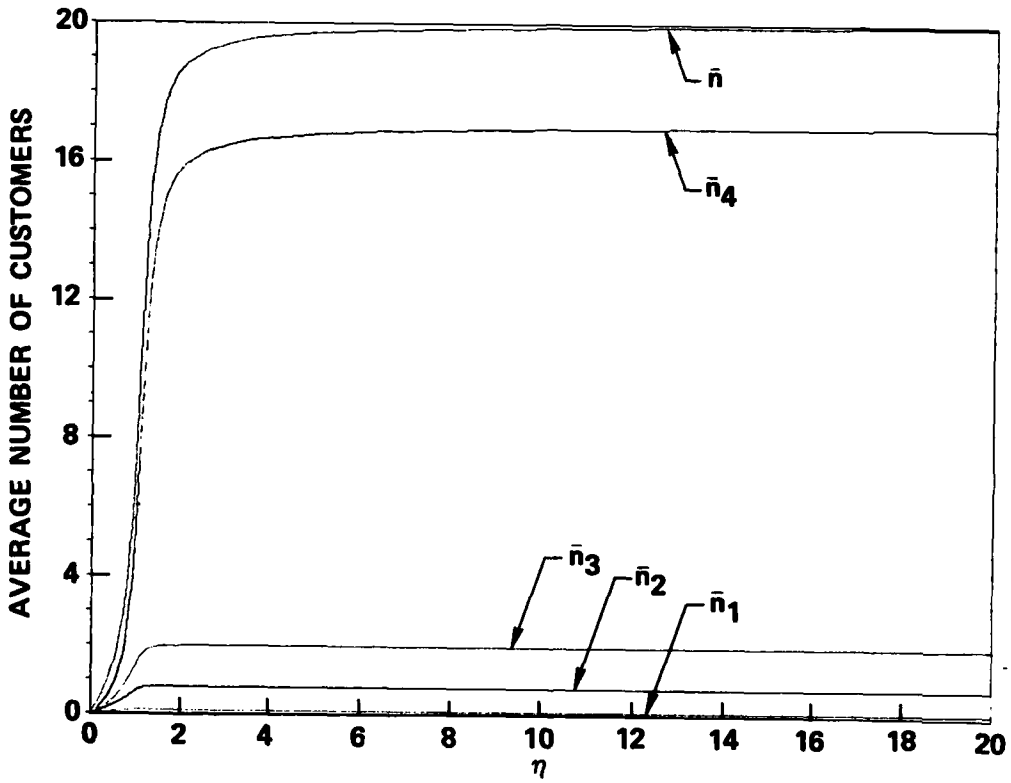


Figure B.3. Average Number of Customers in the System, CS Scheme with Asymetric Input Rates.

In summary, we conclude that with quite asymmetrical utilizations $\{\rho_i^0\}$, CS tends to favor the server with the highest utilization even though it has reached saturation. Furthermore, the other servers are left with very little space to share. These considerations motivate the schemes studied in the rest of this appendix. Before we proceed, let us apply the general results obtained in this section to the case where all ρ_i's are equal.

B.2.3 Special Case: Equal ρ_i's

This section deals with the case where all the ρ_i's are equal and we let ρ be the common value. As a result, a simpler expression is obtained for G(K), and thus for the other variables and distributions.

G(K) is the well known expression obtained for networks of queues [KLEI 75],

$$G(K) = \binom{K + R - 1}{R - 1}\rho^K \tag{B.31}$$

Thus, from Eq. (B.15)

$$P_0^{-1} = \sum_{K=0}^{B} \binom{K + R - 1}{R - 1}\rho^K \tag{B.32}$$

And from Eq. (B.22)

$$PB = \frac{\binom{B + R - 1}{R - 1}\rho^B}{\sum_{K=0}^{B} \binom{K + R - 1}{R - 1}\rho^K} \tag{B.33}$$

Also

$$P(\underset{\sim}{n}) = P_0 \rho^{\Sigma_i n_i} \tag{B.34}$$

With respect to the average number of customers of any type (say i), the numerator of Eq. (B.24) becomes

$$\sum_{n=0}^{B-1} \left(1 - \rho_i^{B-n}\right) G(n) = \sum_{n=0}^{B-1} G(n) - \rho^B \sum_{n=0}^{B-1} \binom{n + R - 1}{R - 1}$$

$$= \sum_{n=0}^{B-1} G(n) - \binom{B + R - 1}{R} \rho^B$$

hence,

$$\bar{n}_i = \frac{\rho}{1 - \rho} \frac{\sum_{K=0}^{B-1} G(K) - \binom{B + R - 1}{R} \rho^B}{\sum_{K=0}^{B} G(K)} \qquad i = 1, \ldots, R \qquad (B.35)$$

In the above proof, we used the identity

$$\sum_{K=0}^{m} \binom{r + k}{k} = \binom{r + m + 1}{m} \qquad (B.36)$$

Similarly, the average time in system of the non-rejected type-i customers is

$$t_i = \frac{1/\mu C_i}{1 - \rho} \frac{\sum_{K=0}^{B-1} \binom{K + R - 1}{R - 1} \rho^K - \binom{B + R - 1}{R} \rho^B}{\sum_{K=0}^{B-1} \binom{K + R - 1}{R - 1} \rho^K} \qquad i = 1, \ldots, R \qquad (B.37)$$

Of interest are the two cases when $\rho = 1$ and $\rho \to \infty$

<u>$\rho = 1$</u>:

$$\left\{ \begin{array}{ll} P_0^{-1} = \binom{B + R}{R} & \lambda_i' = \frac{B}{B + R} \mu C_i \\ P(n_1, \ldots, n_R) = P_0 \quad \forall \underset{\sim}{n} \in F_b & \\ PB = \frac{R}{R + B} & t_i = \frac{B + R}{R + 1} \frac{1}{\mu C_i} \\ \bar{n}_i = \frac{B}{R + 1} \quad i = 1, \ldots, R & \bar{n} = \frac{RB}{R + 1} \end{array} \right. \qquad (B.38)$$

We frequently used Eq. (B.36) in the derivations of the above equations. Note that all states $\underset{\sim}{n}$ are of equal probability P_0. Furthermore, if $R = 1$ we find the results obtained in Section B.1 (Eq. B.11).

The expression of PB may be rewritten as $1/(1 + B/R)$, which is exactly the same as for a single $M|M|1$ queue with B/R buffers (Eq. B.11). This means that for $\rho = 1$, CS and CP lead to the same probability of blocking. This fact will be illustrated in the figures below.

<u>$\rho \to \infty$</u>

The service rates (μC_i) are assumed to be constant. The limits are

$$\begin{cases} P_0 \to 0 \quad , \quad PB \to 1 \\ n_i \to B/R \quad , \quad \bar{n} \to B \\ \lambda_i' \to \dfrac{B}{B + R - 1}\,\mu C_i \qquad i = 1, \ldots, R \\ t_i \to \dfrac{B + R - 1}{R}\,\dfrac{1}{\mu C_i} \qquad i = 1, \ldots, R \end{cases} \tag{B.39}$$

As noticed earlier, infinite input rates do not lead to full utilization of the servers (except for $R = 1$), but only to a fraction $B/(B + R - 1)$ of the capacity.

The behavior of the probability of blocking, PB, the throughput normalized with respect to the capacity, $\lambda_i'/\mu C_i$, and the delay normalized with respect to the average service time, $\mu C_i t_i$, are illustrated respectively in Figs. B.4 - B.6.

This concludes the analysis of the complete sharing (CS) scheme. Let us now compare it with the complete partitioning (CP) scheme.

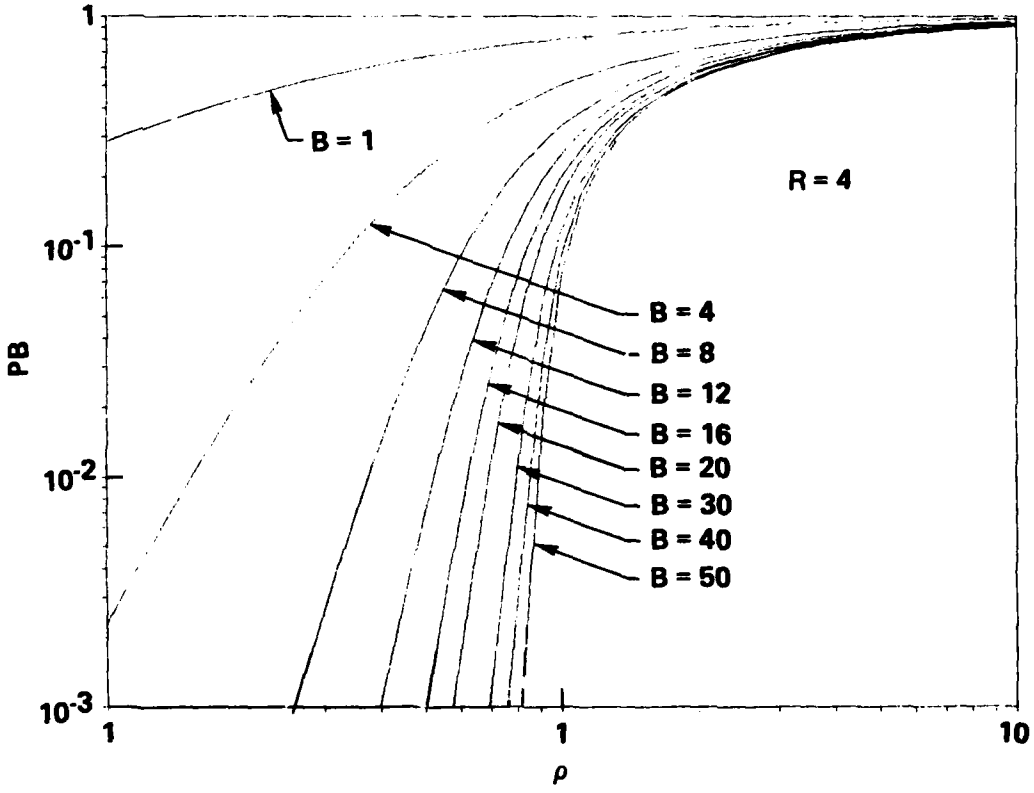


Figure B.4. Probability of Blocking; CS Scheme.

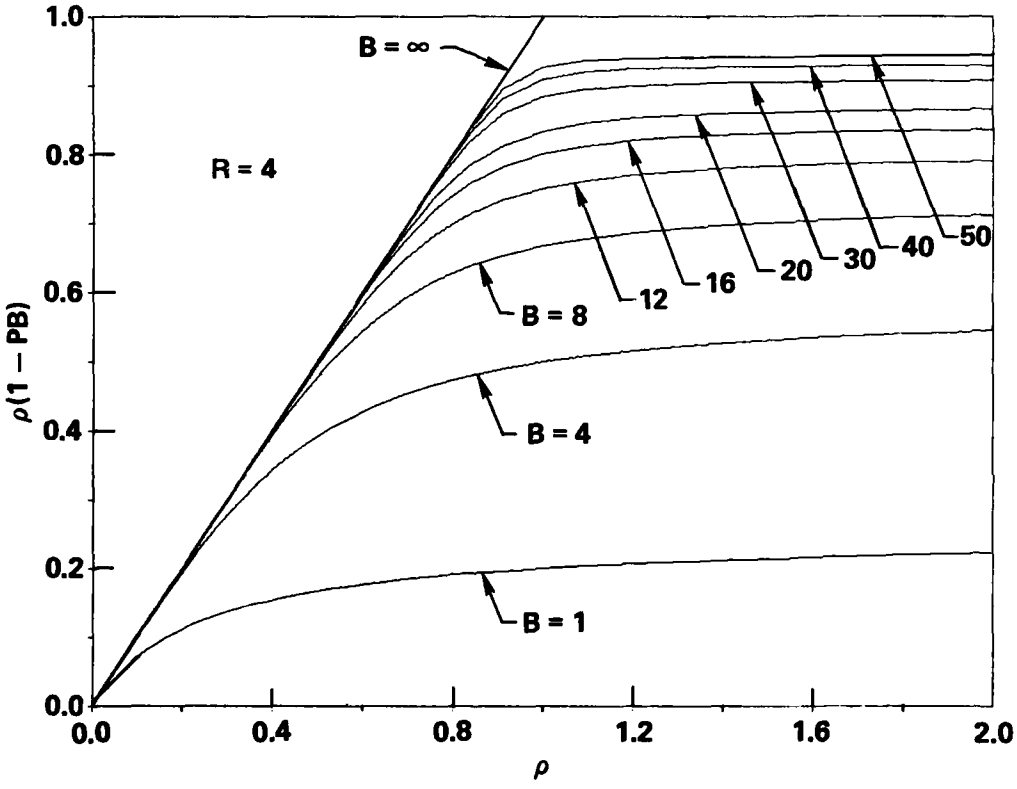


Figure B.5. Normalized Throughput, CS Scheme.

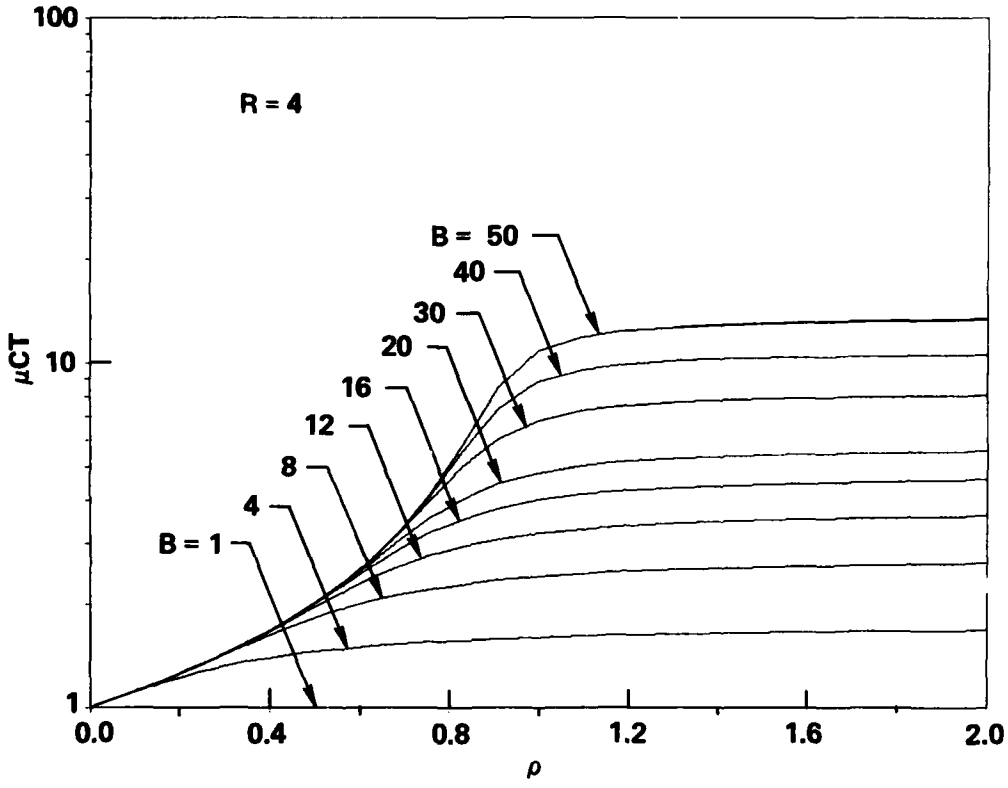


Figure B.6. Normalized Delay; CS Scheme.

B.2.4. Comparison of CP and CS

We assume that all ρ_i's are equal (to ρ) and that each server "contributed" B_0 buffers, i.e., $b_i = B_0 \quad i = 1, \ldots, R$ (see Fig. B.1.a), therefore, $B = RB_0$.

With the above conditions the behavior of CP (any queue) is identical to CS with $R = 1$.

Fig. B.7 illustrates the behavior of the probability of blocking, PB, with respect to ρ and for a set of values of R, $(R = 1, \ldots, 4)$. $R = 1$ corresponds to CP; $R = 2, 3, 4$ corresponds to the merging of 2, 3, 4 single queues. Note that all the curves meet at $\rho = 1$ where, from Eq. (B.38), $PB = 1/(1 + B_0)$. Note also that for $0 \leq \rho \leq 1$, CS leads to a smaller PB, hence a better performance than CP. This improvement is quite considerable for small values of B_0 and increases with R. However, for $\rho \geq 1$, CP shows a slightly better performance (smaller PB) than CS, namely for small values of B_0.

Fig. B.8 shows the respective channel utilizations $\rho(1 - PB)$ (normalized throughputs $\lambda'/\mu C$). Note the loss in limiting throughput $(\rho \to \infty)$ with CS for small values of B_0.

Finally, Fig. B.9 shows the respective average delays. We note that the average message delay increases as more buffers are provided, i.e., as R increases.

The slightly better performance of CP for $\rho > 1$ intuitively indicates that some buffers should be permanently allocated to each server. This idea is incorporated in Scheme 4, SMA. Moreover, we observed earlier that very unbalanced input rates lead to uneven (on the average) usage of the storage space. This remark motivates the next scheme, SMXQ.

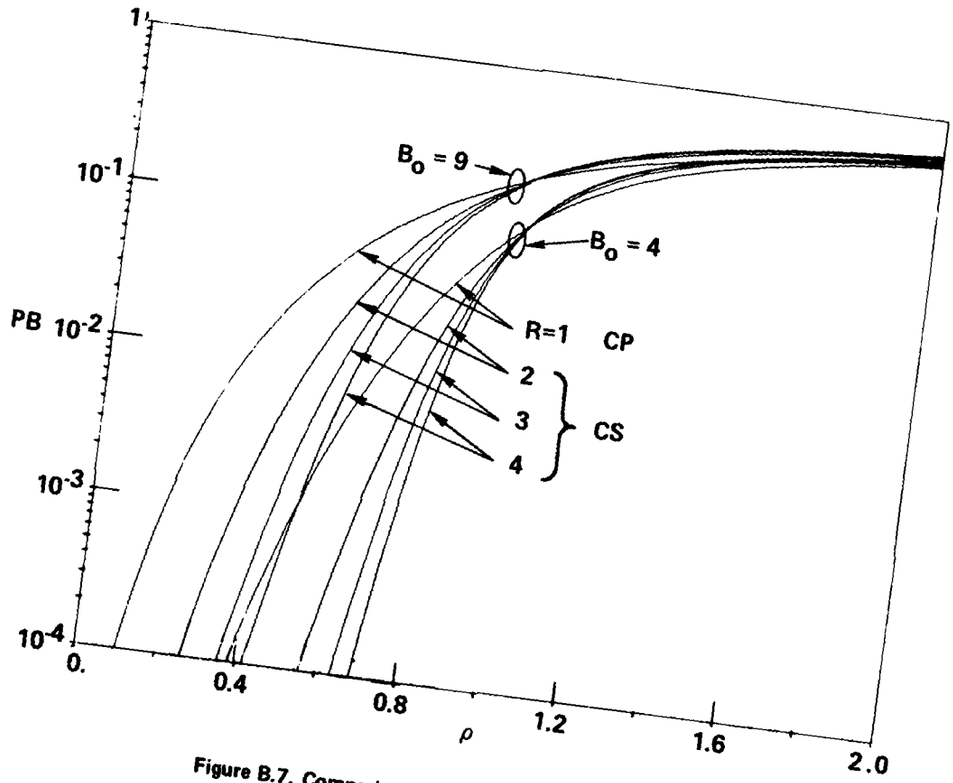


Figure B.7. Comparison of CP and CS : Blocking.

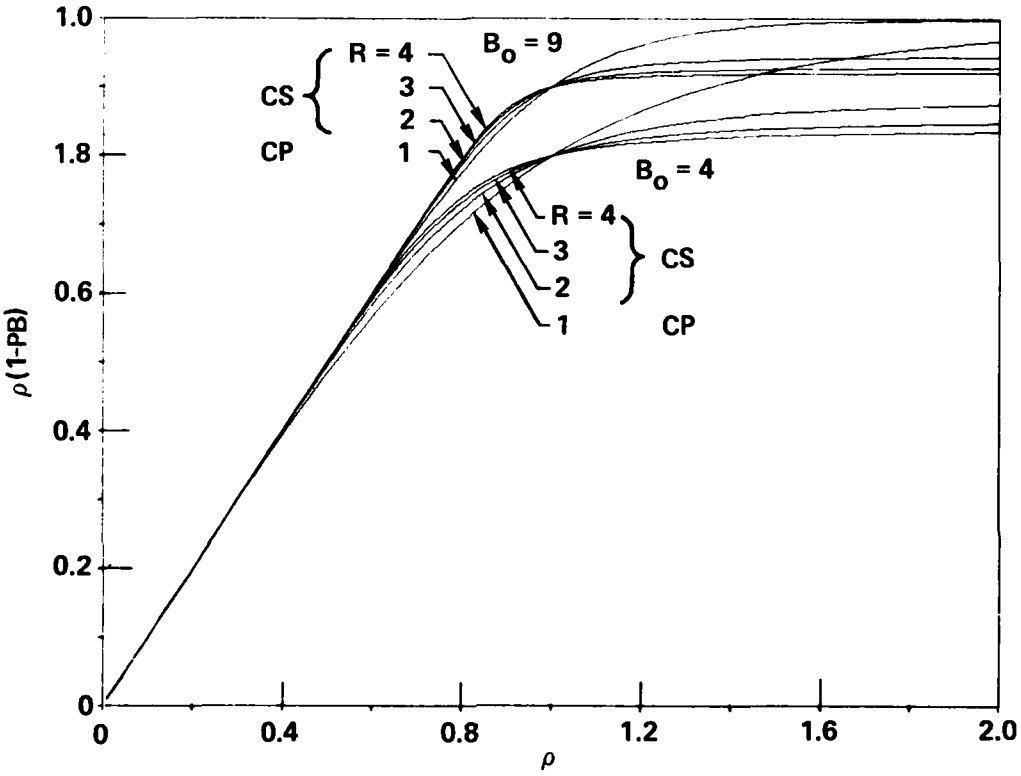


Figure B.8. Comparison of CP and CS : Utilization.

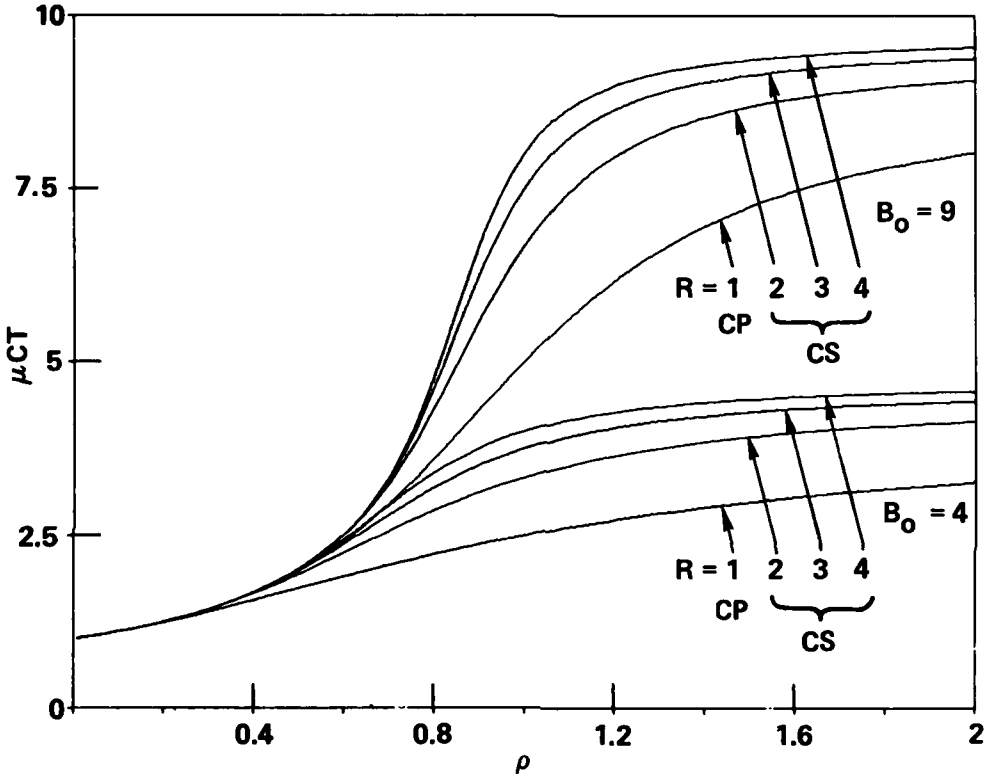


Figure B.9. Comparison of CP and CS : Delay.

B.3 Sharing with Maximum Queue Lengths (SMXQ)

Like CS, SMXQ allows the sharing of a pool of B buffers with a further constraint imposed on the number of buffers to be allocated to any server, and at any time. Let b_i be the maximum number of buffers that can be used by type-i customers; the set of feasible states becomes

$$F_c = \left\{ \underset{\sim}{n} \;\middle|\; 0 \le \sum_{i=1}^{R} n_i \le B \quad , \quad 0 \le n_i \le b_i \qquad i = 1, \ldots, R \right\}$$

Taking advantage of a previous remark, we directly write the set of local balance equations which describe the behavior of the system of queues in steady state. We consider below the possible transitions for type-i customers,

1.
$$\lambda_i P(n_1, \ldots, n_i - 1, \ldots, n_R) + \mu_i P(n_1, \ldots, n_i + 1, \ldots, n_R)$$
$$= (\lambda_i + \mu_i) P(n_1, \ldots, n_i, \ldots, n_R)$$

$$\text{for all } \underset{\sim}{n} \text{ s.t. } \begin{cases} 0 < n_i < b_i \\ 0 \le \sum_{j=1}^{R} n_j + 1 \le B \\ 0 \le n_j \le b_j \qquad j \ne i \end{cases}$$

2.
$$\lambda_i P(n_1, \ldots, n_i - 1, \ldots, n_R) = \mu_i P(n_1, \ldots, n_i, \ldots, n_R)$$

$$\text{for all } \underset{\sim}{n} \text{ s.t. } \begin{cases} n_i = 1 \\ \text{or} \\ n_i = b_i \end{cases} \text{ and } \begin{cases} 0 \le \Sigma n_j \le B \\ 0 \le n_j \le b_j \end{cases}$$

and

$$n_i \ge 1 \text{ and } \begin{cases} \Sigma_j n_j = B \\ 0 \le n_j \le b_j \end{cases}$$

Again, it is easy to check that the joint probability distribution satisfies Eqs. (B.12) and (B.13) where F_b is replaced by F_c.

The evaluation of P_0 is much more complicated here, because of the added constraint on n_i. In what follows, we again consider the two cases of different and equal ρ_i's.

B.3.1 General Case

In this section the ρ_i's are not necessarily equal. We first evaluate P_0. From the above considerations

$$P_0^{-1} = \sum_{K=0}^{B} Q(K) \tag{B.40}$$

where

$$Q(K) = \sum_{\substack{\sum_{i=1}^{R} n_i = K \\ 0 \le n_i \le b_i}} \rho_1^{n_1} \cdots \rho_R^{n_R} \tag{B.41}$$

Note that the difference between Q(K) and G(K), Eq. (B.14), comes from the added constraint, $n_i \le b_i$.

In order to find Q(K), we use a method similar to the generating function approach. Let f(t) be defined as

$$f(t) = \prod_{i=1}^{R} \left(1 + \rho_i t + \rho_i^2 t^2 + \ldots + \rho_i^{b_i} t^{b_i}\right) \tag{B.42}$$

By expanding the above product, we recognize that Q(K) is the factor of t^K, i.e.,

$$f(t) = Q(0) + Q(1)t + \ldots + Q(K)t^K + \ldots + Q\left(\sum_{i=1}^{R} b_i\right) t^{\Sigma b_i}$$

Note that Q(0) = 1 and that the highest degree is equal to $\sum_{i=1}^{R} b_i$.

The function f(t) may be rewritten as,

$$f(t) = \prod_{i=1}^{R} \frac{1 - \rho_i^{1+b_i} t^{1+b_i}}{1 - \rho_i t} \tag{B.43}$$

Let $$h(t) = \prod_{i=1}^{R} \frac{1}{1 - \rho_i^{1+b_i} t^{1+b_i}}$$

$$= \prod_{i=1}^{R} \left(1 + \rho_i^{1+b_i} t^{1+b_i} + \ldots + \rho_i^{k(1+b_i)} t^{k(1+b_i)} \ldots\right) \tag{B.44}$$

Then from Eqs. (B.16), (B.42) - (B.44), we have

$$g(t) = f(t)h(t) \tag{B.45}$$

The above relation will allow us to express G in terms of Q and C_i. The term C_i results from the partial fraction expansion of h(t). Assuming that all ρ_i's are different, we find

$$h(t) = \sum_{i=1}^{R} \frac{C_i}{1 - \rho_i^{1+b_i} t^{1+b_i}}$$

$$= \sum_{i=1}^{R} C_i \left(1 + (\rho_i t)^{1+b_i} + \ldots + (\rho_i t)^{k(b_i+1)} \ldots\right) \tag{B.46}$$

where $$C_i = \prod_{\substack{j=1 \\ j \neq i}}^{R} \frac{1}{1 - (\rho_j/\rho_i)^{1+b_j}} \tag{B.47}$$

Finally,

$$\sum_{K=0}^{\infty} G(K) t^K = \left(\sum_{K=0}^{\Sigma b_i} Q(K)\ t^K\right)\left[\sum_{i=1}^{R} C_i\left(1 + (\rho_i t)^{1+b_i} + \ldots + (\rho_i t)^{k(1+b_i)} + \ldots\right)\right] \tag{B.48}$$

Then equating the terms of equal degrees in t, we arrive at a relation between G, Q and C_i's. This relation is quite complicated and requires the ordering of the b_i's and their multiples. However, if we restrict our considerations either to the case where $b_i \geq \frac{B}{2}$, or when $b_i = b$ for all i, then we obtain the simple relations below.

i. We assume that each queue is allowed to occupy more than half of the entire space, i.e.,

$$b_i \geq B/2 \qquad \forall\, i = 1, \ldots, R \tag{B.49}$$

then $2b_i + 2 > B$ and $b_i + b_j + 2 > B$. Therefore, in order to find $Q(K)$, $K = 0, 1, \ldots, B$, we only need to consider the terms in $h(t)$, of degree 0 or $1 + b_i$. Those terms, obtained directly from Eq. (B.44) (i.e., without the assumption of different ρ_i's), are

$$1 + \sum_{i=1}^{R} (\rho_i t)^{1+b_i}$$

Substituting the above expression into Eq. (B.48), we arrive at

$$G(K) = Q(K) + \sum_{\substack{i=1 \\ i \text{ s.t. } b_i < K}}^{R} \rho_i^{1+b_i} Q(K - b_i - 1) \tag{B.50}$$

ii. $\underline{b_i = b}$ $\quad i = 1, \ldots, R$

Similar to G(K), let us define

$$L(K) = \sum_{\Sigma n_i = K} \rho_1^{(1+b)n_1} \ldots \rho_R^{(1+b)n_R} \tag{B.51}$$

Then, from Eqs. (B.45) - (B.48),

$$L(K) = \sum_{i=1}^{R} C_i \rho_i^{(1+b)K} \tag{B.52}$$

and

$$h(t) = \sum_{K \geq 0} L(K) t^{(1+b)K} \tag{B.53}$$

Substituting the above into Eq. (B.48), we arrive at

$$G(\alpha(b + 1)+k) = \sum_{i=0}^{\alpha} Q((\alpha - i)(b + 1) + k) L(i) \tag{B.54}$$

where $0 \leq k \leq b$

and $0 \leq \alpha(b + 1) + k \leq B$

Note that $G(0) = Q(0) = L(0) = 1$, and that Eq. (B.54) allows the sequential computation of the sequence $Q(K)$, for K varying from 1 to B.

Note that if $b \geq B/2$, then Eq. (B.54) becomes

$$\begin{cases} G(k) = Q(k) & \text{for } 0 \leq k \leq b \\ G(b + 1 + k) = Q(b + 1 + k) + Q(k)L(1) & 0 \leq k \leq B - b - 1 \end{cases}$$

and from Eq. (B.51), $L(1) = \sum_{i=1}^{R} \rho_i^{(1+b)}$, thus the combination of the last three equations gives Eq. (B.50) with $b_i = b$.

In what follows, we restrict our considerations to the case where $b_i \geq B/2$, unless specified otherwise. As a result, and from Eqs. (B.40) and (B.50), we arrive at

$$P_0^{-1} = \sum_{K=0}^{B} G(K) - \sum_{i=1}^{R} \rho_i^{1-b_i} \sum_{K=0}^{B-b_i-1} G(K) \tag{B.55}$$

Note that if $b_i = B$ for all i, then SMXQ becomes CS, and the above equation reduces[1] to Eq. (B.15).

[1] By convention we set $\sum_{a}^{a-1} \triangleq 0 \quad \forall$ a integer

Similar to Section B.2, we now assume that all the ρ_i's are different ($\rho_i \neq \rho_j \quad \forall\, i \neq j$); then using Eqs. (B.19) and (B.20), we arrive at

$$P_0^{-1} = \sum_{i=1}^{R} A_i \frac{1 - \rho_i^{B+1}}{1 - \rho_i} - \sum_{\substack{i=1 \\ 0 \le b_i < B}}^{R} \rho_i^{1+b_i} \sum_{j=1}^{R} A_j \frac{1 - \rho_j^{B-b_i}}{1 - \rho_j} \tag{B.56}$$

Eq. (B.12) with F_b replaced by F_c, and Eq. (B.55) or (B.56) completely characterize the queueing system under SMXQ and the condition of Eq. (B.49). Similarly to Section B.2, we proceed with the derivation of distributions and average quantities of interest.

<u>Distribution of the total number in system: Probability of Blocking</u>

Let n be the total number in system; then, for $0 \le n \le B$, $P_n \triangleq P_r[\Sigma_i n_i = n] = P_0 Q(n)$.

$$P_n = P_0\left[G(n) - \sum_{\substack{i=1 \\ b_i < n}}^{R} \rho_i^{1+b_i} G(n - b_i - 1)\right] \tag{B.57}$$

Let us now derive the probability of blocking of type-i customers, PB_i. Recall that type-i customers are blocked if upon arrival the entire space is full, or if the number of type-i customers is equal to b_i. Since arrivals are Poisson, then

$$PB_i = P_r[\Sigma_j n_j = B \quad \text{or} \quad n_i = b_i] \tag{B.58}$$

which is also

$$PB_i = P_r[\Sigma_j n_j = B] + P_r[n_i = b_i \quad \text{and} \quad \Sigma_j n_j < B]$$

Moreover,

$$PB_i = P_B + \sum_{K=0}^{B-b_i-1} P_r[n_i = b_i \quad \text{and} \quad \sum_j n_j = b_i + K]$$

Because of Eq. (B.49),

$$n_i = b_i \geq B/2 \quad \text{and} \quad \Sigma_j n_j = b_i + K \Rightarrow n_j \leq b_j \qquad \forall\, j \neq i$$

Hence

$$P_r[n_i = b_i \quad \text{and} \quad \sum_j n_j = b_i + K] = P_0 \rho_i^{b_i} G_i(K)$$

where

$$G_i(K) = \sum_{\substack{\Sigma\, n_j = K \\ j \neq i}} \rho_1^{n_1} \cdots \rho_{i-1}^{n_{i-1}} \rho_{i+1}^{n_{i+1}} \cdots \rho_R^{n_R} \tag{B.59}$$

Finally

$$PB_i = P_B + P_0 \rho_i^{b_i} \sum_{K=0}^{B-b_i-1} G_i(K) \tag{B.60}$$

where P_B is given by Eq. (B.57).

Note that $G_i(K)$ is similar to $G(K)$, except that we deleted ρ_i (i.e., type-i customers).

We now proceed with the derivation of the marginal distribution and average number and delay of type-i customers.

<u>Marginal distribution and averages</u>

Let $k \leq b_i$, then

$$P_r[n_i = k] = P_0 \sum_{\substack{\underset{\sim}{n} \in F_c \\ n_i = k}} \rho_1^{n_1} \cdots \rho_R^{n_R} = P_0 \rho_i^k \sum_{\substack{0 \leq \sum_{j \neq i} n_j \leq B-k \\ 0 \leq n_j \leq b_j}} \left(\prod_{\substack{j=1 \\ j \neq i}}^{R} \rho_j^{n_j} \right)$$

The summation above is similar to that of P_0^{-1} except that B is now B - k and the component n_i is deleted. Also note that $b_j \geq \frac{B - k}{2}$ $\forall j \neq i$, hence the above summation is given by Eq. (B.55) where B is replaced by B - k and G(K) by $G_i(K)$ (as defined in Eq. (B.59)). Thus,

$$P_r[n_i = k] = P_0 \rho_i^k \left[\sum_{K=0}^{B-k} G_i(K) - \sum_{\substack{j=1 \\ j \neq i}}^{R} \rho_j^{1+b_j} \sum_{K=0}^{B-k-b_j-1} G_i(K) \right] \quad (B.61)$$

The above summations can be further reduced to expressions similar to the one in Eq. (B.56).

As for the average number of type-i customers,

$$\bar{n}_i = \sum_{k=1}^{b_i} k P_r[n_i = k] \quad (B.62)$$

With respect to the throughput λ_i' and the delay t_i, the expressions $\lambda_i' = (1 - PB_i)\lambda_i$ and $t_i = \bar{n}_i/\lambda_i'$ still hold true.

This terminates the characterization of the system as operated with SMXQ and with the assumption of $b_i \geq B/2$ and different ρ_i's. Next we study the case of equal ρ_i's; we leave numerical applications to Section B.6.

B.3.2 Special Case: Equal ρ_i's

Similar to Section B.2.3, let $\rho_i = \rho$ $\forall$ i; then, G(K) is given by Eq. (B.31). Also, we assume that all the b_i's are equal to b and that $b \geq B/2$. Therefore, from Eq. (B.55) P_0^{-1} becomes

$$P_0^{-1} = \sum_{K=0}^{B} \binom{K + R - 1}{R - 1} \rho^K - R\rho^{b+1} \sum_{K=0}^{B-b-1} \binom{K + R - 1}{R - 1} \rho^K \quad (B.63)$$

As for the distribution of the total number of customers, from Eq. (B.57) we obtain (for $n \leq B$)

$$P_n = \begin{cases} P_0 \binom{n + R - 1}{R - 1} \rho^n & 0 \leq n \leq b \\ P_0 \left[\binom{n + R - 1}{R - 1} - R \binom{n - b + R - 2}{R - 1} \right] \rho^n & b < n \leq B \end{cases} \tag{B.64}$$

In order to find the probability of overflow PB (the same for any type of customers), we need to evaluate $G_i(K)$ (Eq. (B.59)); similarly to $G(K)$, we have

$$G_i(K) = \binom{K + R - 2}{R - 2} \rho^K \tag{B.65}$$

Substituting Eq. (B.64) for $n = B$, and Eq. (B.65) into Eq. (B.60), we arrive at

$$PB_i = PB = P_0 \rho^B \left[\binom{B + R - 1}{R - 1} - R \binom{B - b + R - 2}{R - 1} \right] + P_0 \rho^b \sum_{K=0}^{B-b-1} \binom{K + R - 2}{R - 2} \rho^K \tag{B.66}$$

With regard to the marginal distribution of any type of customer, say i, from Eqs. (B.61) and (B.65), we find

$$P_r[n_i = k] = P_0 \rho^k \left[\sum_{K=0}^{B-k} \binom{K + R - 2}{R - 2} \rho^K - (R-1) \rho^{b+1} \sum_{K=0}^{B-b-k-1} \binom{K + R - 2}{R - 2} \rho^K \right] \quad \text{for } k \leq b$$

$$= 0 \qquad \text{otherwise} \tag{B.67}$$

The rest of the expressions, $\bar{n}_i$, λ_i', t_i, follow in the same way as before. Of interest is the case where $\rho = 1$ and $\rho \to \infty$

<u>$\rho = 1$</u>:

Using the identity in Eq. (B.36), we arrive at

$$\begin{cases} P_0^{-1} = \binom{B+R}{R} - R\binom{B-b-1+R}{R} \\ PB = P_0\left[\binom{B+R-1}{R-1} - (R-1)\binom{B-b+R-2}{R-1}\right] \end{cases} \tag{B.68}$$

Note that <u>if R = 1</u>, then $PB = P_0 = 1/(b + 1)$, which is exactly the probability of blocking (and P_0) for a single $M|M|1$ queue with b buffers; because of the constraint, $n_i \leq b$, the rest of the space $B - b$ is unutilized.

As for <u>R = 2</u>, then

$$PB = \frac{b+1}{(B+2)(B+1)/2 - (B-b+1)(B-b)} \tag{B.69}$$

Note that for the two limiting cases where $b = B$, i.e., CS and $b = B/2$, i.e., CP, we have $PB = 2/(B + 2)$; which checks with the previous results (Sections B.1 and B.2). Moreover, a question arises as to choice of b. This question and numerical examples are presented in Section B.6.

Let us now consider the case when ρ goes to infinity.

<u>$\rho \to \infty$</u>

Of interest is the limiting utilization. Let ρ' be the utilization of any server; then $\rho' = (1 - PB)\rho$. Before we proceed, note that $P_0 \to 0$, $PB \to 1$. As for ρ'

$$\rho' = (1 - PB)\rho \to \frac{\binom{B+R-2}{R-1} - R\binom{B-b+R-3}{R-1} - \binom{B-b+R-3}{R-2}}{\binom{B+R-1}{R-1} - R\binom{B-b+R-2}{R-1}} \tag{B.70}$$

Note that <u>if R = 1</u> then $\rho' = 1$ (provided that $B \geq 1$); whereas, for <u>R = 2</u> ($B \geq 1$), two cases are possible:

i. $B/2 \leq b < B \Rightarrow \rho' = 1$

ii. $b = B \Rightarrow \rho' = B/(B + 1)$

Both results are to be expected. $b < B$ implies that no one of the two types can utilize the whole space, and hence, at $\rho = \infty$ the two types are always present. $b = B$ implies CS; hence we are back to the results of Eq. (B.39) for $\lambda_i'/\mu C_i$ and $R = 2$.

B.4 Sharing with Minimum Allocations (SMA)

Similar to CS, SMA allows the sharing of a pool of B buffers, and in addition, a_i buffers are permanently allocated to type-i customers, $i = 1, \ldots, R$ (see Fig. B.1). As a result, the set of feasible states becomes

$$F_d = \left\{ \underset{\sim}{n} \;\middle|\; \sum_{i=1}^{R} \sup\{0, n_i - a_i\} \leq B \quad , \quad 0 \leq n_i \leq B + a_i \quad i = 1, \ldots, R \right\}$$

Here also, we can write the local balance equations and verify that the product form solution satisfies those equations. Following the same steps as earlier, we first consider the general case of different ρ_i's.

B.4.1 General Case

In order to evaluate P_0, we partition the set F_d into disjoint subsets which lead to known summations. Let $\mathscr{R}$ be the set of customer types,

$$\mathscr{R} = \{1, 2, \ldots, R\}$$

and let $\mathscr{X}$ be the set of all subsets of $\mathscr{R}$,

$$\mathscr{X} = \{X_m \mid X_m \subset \mathscr{R} \quad , \qquad 1 \leq m \leq 2^R\}$$

The set $\mathscr{X}$ contains 2^R elements; among them are the set $\mathscr{R}$ itself and the empty set. We then associate with each subset, X_m, a subset of F_d, S_m defined as

$$S_m = \left\{ \underset{\sim}{n} \in F_d \;\middle|\; n_i \begin{cases} \geq a_i & i \in X_m \\ < a_i & \text{otherwise} \end{cases} \right\}$$

Equivalently,

$$S_m = \left\{ \underset{\sim}{n} \;\middle|\; \begin{array}{l} \sum\limits_{i \in X_m} (n_i - a_i) \leq B, \quad a_i \leq n_i \leq B + a_i \text{ for } i \in X_m \\ \qquad\qquad\qquad\qquad\qquad\qquad n_i < a_i \text{ for } i \notin X_m \end{array} \right\}$$

Obviously,

$$S_m \cap S_n = \phi \qquad m \neq n \qquad \text{and} \qquad F_d = \bigcup_{m=1}^{2^R} S_m \,.$$

Therefore,

$$P_0^{-1} = \sum_{\underset{\sim}{n} \in F_d} \left(\prod_{i=1}^{R} \rho_i^{n_i} \right) = \sum_{m=1}^{2^R} \sum_{\underset{\sim}{n} \in S_m} \left(\prod_{i=1}^{R} \rho_i^{n_i} \right) \tag{B.71}$$

Let H_m be the summation over all states in S_m, then

$$H_m \triangleq \sum_{\underset{\sim}{n} \in S_m} \left(\prod_{i=1}^{R} \rho_i^{n_i} \right) = \prod_{i \notin X_m} \left(\sum_{0 \leq n_i < a_i} \rho_i^{n_i} \right) \sum_{\substack{0 \leq \sum\limits_{i \in X_m} (n_i - a_i) \leq B \\ a_i \leq n_i \leq B + a_i}} \prod_{i \in X_m} \rho_i^{n_i}$$

Hence,

$$H_m = \prod_{i \notin X_m} \frac{1 - \rho_i^{a_i}}{1 - \rho_i} \left(\prod_{i \in X_m} \rho_i^{a_i} \right) \sum_{\substack{0 \leq \sum\limits_{i \in X_m} n_i \leq B \\ 0 \leq n_i \leq B}} \prod_{i \in X_m} \rho_i^{n_i}$$

Let us define the generating function C_m, such that

$$C_m(K) = \sum_{\substack{\sum_{i\in X_m} n_i = K \\ 0 \le n_i \le K}} \left(\prod_{i\in X_m} \rho_i^{n_i} \right) \tag{B.72}$$

$C_m(K)$ is similar to $G(K)$ given in Eq. (B.14), thus it can be computed in the same way. Finally, let $\underset{\sim}{a} = (a_1, \ldots, a_R)$. Then

$$\begin{cases} P_0^{-1}[\underset{\sim}{a}, B] \triangleq P_0^{-1} = \sum_{m=1}^{2^R} H_m \\ H_m[\underset{\sim}{a}, B] \triangleq H_m = \prod_{i\notin X_m} \dfrac{1-\rho_i^{a_i}}{1-\rho_i} \prod_{i\in X_m} \rho_i^{a_i} \sum_{K=0}^{B} C_m(K) \end{cases} \tag{B.73}$$

Note that if $a_i = 0$ for all i, then $\mathscr{X} = \{\mathscr{R}\}$, $C_m(K) = G(K)$, and the above equation reduces to the description of CS. Also, the summation of $C_m(K)$ is set to 1 if $X_m = \phi$.

If we now assume that all ρ_i's are different, then from Eqs. (B.18) and (B.19),

$$C_m(K) = \sum_{i\in X_m} A_{i,m}\, \rho_i^K \tag{B.74}$$

where

$$A_{i,m} = \prod_{\substack{k\in X_m \\ k\neq i}} \frac{1}{1-\rho_k/\rho_i} \tag{B.75}$$

Using a similar summation as in Eq. (B.20), we arrive at

$$H_m = \prod_{i\notin X_m} \frac{1-\rho_i^{a_i}}{1-\rho_i} \prod_{i\in X_m} \rho_i^{a_i} \sum_{i\in X_m} A_{i,m} \frac{1-\rho_i^{B+1}}{1-\rho_i} \tag{B.76}$$

Eq. (B.12) with F_b replaced by F_d, and Eq. (B.73) or (B.76) completely characterize our system. We now proceed with the derivation of distributions and averages of the variables of interest.

Distribution of total number of customers in shared area; Probability of Blocking

Let n_s be the total number of customers in the shared area, i.e.,

$$n_s \triangleq \sum_{i=1}^{R} \sup \{0, n_i - a_i\} \tag{B.77}$$

Then the distribution of n_s is, for $k \leq B$, equal to the sum of probabilities of states $\underset{\sim}{n}$ which satisfy Eq. (B.77) for $n_s = k$. In order the evaluate that summation, we use the same methodology as for the determination of P_0. Let $F_d(k) \subset F_d$ and $S_m(k) \subset S_m$ be defined as

$$F_d(k) = \left\{ \underset{\sim}{n} \;\middle|\; \sum_{i=1}^{R} \sup \{0, n_i - a_i\} = k, \qquad 0 \leq n_i \leq B + a_i \right\}$$

$$S_m(k) = \left\{ \underset{\sim}{n} \;\middle|\; \begin{array}{lll} a_i \leq n_i \leq a_i + k & i \in X_m & , \quad \sum_{i \in X_m} (n_i - a_i) = k \\ n_i < a_i & i \notin X_m & \end{array} \right\}$$

It is obvious that $F_d(k) = \bigcup_m S_m(k)$, and consequently

$$P_r[n_s = k] = P_0 \sum_{m=1}^{2^R} \sum_{\underset{\sim}{n} \in S_m(k)} \left(\prod_{i=1}^{R} \rho_i^{n_i} \right) \tag{B.78}$$

Let $h_m(k)$ be the summation over all states in $S_m(k)$, then similar to the above

$$\begin{cases} P_r[n_s = k] = P_0 \sum_{m=1}^{2^R} h_m(k) \\ h_m(k) = \prod_{i \notin X_m} \dfrac{1 - \rho_i^{a_i}}{1 - \rho_i} \left(\prod_{i \in X_m} \rho_i^{a_i} \right) C_m(k) \end{cases} \tag{B.79}$$

Note that if $X_m = \phi$, then $h_m(k) \triangleq \prod_{i=1}^{R} \frac{1 - \rho_i^{a_i}}{1 - \rho_i}$ and that

$$P_r[n_s = 0] = P_0 \sum_{0 \le n_i \le a_i} \rho_1^{n_1} \cdots \rho_R^{n_R} = P_0 \prod_{i=1}^{R} \frac{1 - \rho_i^{1+a_i}}{1 - \rho_i} \qquad (B.80)$$

With respect to the probability of blocking type-r customers, PB_r, it is

$$PB_r = P_r[n_s = B \quad \text{and} \quad n_r \geq a_r] \qquad (B.81)$$

PB_r can be computed similarly to $P_r[n_s = B]$ with a restriction on the choice of the subsets X_m, which must contain r, i.e.,

$$PB_r = P_0 \sum_{m \mid r \in X_m} h_m(B) \qquad (B.82)$$

There are 2^{R-1} such sets which can be obtained as follows. Let $\mathscr{R}' = \mathscr{R} - \{r\}$ and $\mathscr{X}' = \{X_n' \quad n = 1, \ldots, 2^{R-1}\}$ be the set of subsets of $\mathscr{R}'$, then $X_n = X_n' \cup \{r\}$ is such a subset of $\mathscr{R}$ which contains r.

Marginal distribution and average number and time in system

We now proceed with the derivation of the marginal distribution of the number of type-i customers; more precisely, we intend to find $P_r[n_r \geq j]$. We consider the two cases which depend on whether j is less than a_r or not.

i. $0 \leq j < a_r$

$$P_r[n_r \geq j] = P_0 \rho_r^j \sum_{\substack{\underset{\sim}{n} \in F_d \\ n_r \geq j}} \rho_1^{n_1} \cdots \rho_r^{n_r - j} \cdots \rho_r^{n_R}$$

Let

$$a_i' = \begin{cases} a_i & i \neq r \\ a_r - j \end{cases} \tag{B.83}$$

be the new minimum allocations, and F_d' be such that

$$F_d' = \left\{ \underset{\sim}{n} \;\middle|\; \sum_{i=1}^{R} \sup\{0, n_i - a_i'\} \leq B, \quad 0 \leq n_i \leq B + a_i' \right\}$$

Therefore

$$P_r[n_r \geq j] = P_0 \rho^j \sum_{\underset{\sim}{n} \in F_d'} \rho_1^{n_1} \cdots \rho_r^{n_r} \cdots \rho_R^{n_R}$$

The summation above is similar to the one of P_0^{-1}, Eq. (B.71), where F_d is replaced by F_d'. Let us define the vector $\underset{\sim}{a}' = (a_1', a_2', \ldots, a_R')$; then, replacing $\underset{\sim}{a}$ by $\underset{\sim}{a}'$ in Eq. (B.73), we find

$$P_r[n_r \geq j] = P_0 \rho_r^j P_0^{-1}[\underset{\sim}{a}', B] = P_0 \rho_r^j \sum_{m=1}^{2^R} H_m[\underset{\sim}{a}', B] \tag{B.84}$$

$$\forall\, j < a_r$$

ii. $\underline{a_r \leq j \leq B}$

$$P_r[n_r \geq j] = P_0 \rho_r^{j-a_r} \sum_{\substack{\underset{\sim}{n} \in Fd \\ \tilde{n}_r \geq j}} \rho_1^{n_1} \cdots \rho_r^{n_r - j + a_r} \cdots \rho_R^{n_R}$$

To evaluate the above summation, let us define the set

$$F_d'' = \left\{ \underset{\sim}{n} \;\middle|\; \begin{array}{l} \sum_{i=1}^{R} \sup\{0, n_i - a_i\} \leq B - j + a_r \\ 0 \leq n_i \leq B - j + a_r + a_i \end{array} \right\}$$

Then, replacing $n_r - j + a_r$ by n_r, in the above equation, we arrive at

$$P_r[n_r \geq j] = P_0 \rho_r^{j-a_r} \sum_{\substack{\underset{\sim}{n} \varepsilon F_d'' \\ n_r \geq a_r}} \rho_1^{n_1} \cdots \rho_r^{n_r} \cdots \rho_R^{n_R}$$

The set F_d'' is similar to F_d except for the condition $n_r \geq a_r$, and the change from B to $B - j + a_r$; then similar to the derivation of PB_r,

$$P_r[n_r \geq j] = P_0 \rho_r^{j-a_r} \sum_{\substack{m \\ m \mid r \varepsilon X_m}} H_m(\underset{\sim}{a}, B - j + a_r) \quad \text{for} \quad a_r \leq j \leq B \tag{B.85}$$

Eqs. (B.84) and (B.85) give the marginal distribution of the number of type-r customers. As a result, we can derive $\bar{n}_r$, λ_r' and t_r.

Let us now apply the above results for the special case of uniform utilization and allocation.

B.4.2 <u>Special case</u>: $\rho_i = \rho$, $a_i = a$

The assumption of equal ρ_i's and a_i's, leads to much simpler expressions of the variables above. First, if p is the size of the subset $X_m (p = |X_m|)$, then from Eq. (B.76)

$$C_m(K) = \binom{K + p - 1}{p - 1} \rho^K \tag{B.86}$$

Note that if p = 0 then

$$C_m(K) = \begin{cases} 0 & K > 0 \\ 1 & K = 0 \end{cases} \tag{B.87}$$

Also, from Eq. (B.77)

$$H_m = \left(\frac{1 - \rho^a}{1 - \rho}\right)^{R-p} \rho^{pa} \sum_{K=0}^{B} \binom{K + p - 1}{p - 1} \rho^K \tag{B.88}$$

Note that $C_m(K)$ and $H_m(k)$ depend only on the size p of the set X_m. The number of sets X_m of size p is equal to $\binom{R}{p}$, thus from Eqs. (B.73) and (B.88),

$$P_0^{-1} = \sum_{p=0}^{R} \binom{R}{p} \left(\frac{1 - \rho^a}{1 - \rho}\right)^{R-p} \rho^{pa} \sum_{K=0}^{B} \binom{K + p - 1}{p - 1} \rho^K \qquad (B.89)$$

Similarly, we derive the expression for PB_r (Eq. (B.82). Recall that we only account for sets X_m which contain r, hence $p \geq 1$.

$$PB_r = P_0 \sum_{p=1}^{R} \binom{R - 1}{p - 1} \left(\frac{1 - \rho^a}{1 - \rho}\right)^{R-p} \rho^{pa} \binom{B + p - 1}{p - 1} \rho^B \qquad (B.90)$$

$$r = 1, \ldots, R .$$

Note that in the above expressions, a was assumed to be greater than zero; <u>if a = 0</u>, then all subsets X_m are empty except one: $X_m = \mathscr{R}$, whose size is equal to R. Moreover, with a = 0, SMA reduces to CS and Eqs. (B.89) and (B.90) for a = 0 check with Eqs. (B.32) and (B.33). Now, <u>if B = 0</u> then SMA reduces to CP, and in fact, Eqs. (B.89) and (B.90) become

$$P_0^{-1} = \left(\frac{1 - \rho^{a+1}}{1 - \rho}\right)^R \quad , \quad PB_r = \frac{1 - \rho}{1 - \rho^{a+1}} \rho^a$$

Let us now derive the marginal distribution of the number of type-r customers. Eq. (B.84) provides $P_r[n_i \geq j]$ for $j < a_r$; the terms $H_m[\underset{\sim}{a}', B]$ can be evaluated as in (B.89) except that $a_r' = a_r - j$. Therefore, we must distinguish the sets X_m which contain r from those which do not. If $p = |X_m|$, then $\binom{R - 1}{p - 1}$ such sets contain r and $\binom{R - 1}{p}$ do not. As a consequence

$$P_r[n_r \geq j] = P_0 \sum_{p=1}^{R} \binom{R-1}{p-1} \left(\frac{1-\rho^a}{1-\rho}\right)^{R-p} \rho^{pa} \sum_{K=0}^{B} \binom{K+p-1}{p-1} \rho^K$$

$$+ P_0 \sum_{p=0}^{R-1} \binom{R-1}{p} \left(\frac{1-\rho^a}{1-\rho}\right)^{R-p-1}$$

$$\times \frac{1-\rho^{a-j}}{1-\rho} \rho^{pa+j} \sum_{K=0}^{B} \binom{K+p-1}{p-1} \rho^K \qquad \text{for } j < a_r \tag{B.91}$$

For the case where $j \geq a_r$, then from Eq. (B.85) and using the same procedure as above, we find

$$P_r[n_r \geq j] = P_0 \sum_{p=1}^{R} \binom{R-1}{p-1} \left(\frac{1-\rho^a}{1-\rho}\right)^{R-p} \rho^{(p-1)a+j} \sum_{K=0}^{B-j+a} \binom{K+p-1}{p-1} \rho^K \tag{B.92}$$

With respect to $\bar{n}_r$, λ_r', t_r , they follow from the above considerations.

Of further interest is the limiting behavior when ρ goes to infinity.

<u>$\rho \to \infty$</u>

We expect, with the minimum allocation of a buffers per server, to obtain a full utilization of all servers. The terms of highest degree in P_0^{-1}, PB_r and $\rho(1 - PB_r)$ are

$$\binom{B+R-1}{R-1} \rho^{B+Ra} \qquad \text{for } P_0^{-1} \text{ and } PB_r$$

$$\left[\binom{B+R-2}{R-1} + \binom{B+R-2}{R-2}\right] \rho^{B+Ra} \qquad \text{for } \rho(1 - PB_r)$$

Therefore, $\rho \to \infty \implies P_0 \to 0$, $PB_r \to 1$, $\rho(1 - PB_r) \to 1$.

This concludes the analysis of CSMA, the numerical applications and questions related to the choice of $\underset{\sim}{a}$ are treated in Section B.6.

B.5 Sharing with Maximum Queue Length and Minimum Allocations: SMQMA

In addition to SMA, SMQMA (or scheme e) imposes a constraint on the number of buffers from the shared pool to be allocated to any server, and at any time (see Fig. B.1). Let b_i be that constraint with respect to server i. As a result, the set of feasible states becomes

$$F_e = \{n \in F_d \mid 0 \leq \sup\{0, n_i - a_i\} \leq b_i \qquad i = 1, \ldots, R\}$$

Equivalently,

$$F_e = \left\{\underset{\sim}{n} \;\middle|\; 0 \leq \sum_{i=1}^{R} \sup\{0, n_i - a_i\} \leq B, \qquad 0 \leq n_i \leq a_i + b_i\right\}$$

Here also, we can check that the product form solution satisfies the local balance equations. Thus we may proceed as earlier with the evaluation of P_0 and subsequently determine the distributions and quantities of interest. First we consider the general case of different ρ_i's.

General case

The same procedure as in Section B.4 can be utilized to partition the set F_e into disjoint subsets which then lead to known summations. Those subsets are

$$S_m^e = \left\{\underset{\sim}{n} \;\middle|\; \sum_{i \in X_m} (n_i - a_i) \leq B, \qquad \begin{array}{ll} a_i \leq n_i \leq b_i + a_i & i \in X_m \\ n_i < a_i & i \notin X_m \end{array}\right\}$$

Consequently,

$$P_0^{-1} = \sum_{m=1}^{2^R} H_m^e \tag{B.93}$$

with

$$H_m^e = \left(\prod_{i \notin X_m} \frac{1 - \rho_i^{a_i}}{1 - \rho_i} \right) \left(\prod_{i \in X_m} \rho_i^{a_i} \right) \sum_{K=0}^{B} Q(K) \qquad (B.94)$$

where Q(K) is as defined in Eq. (B.41)

As a consequence, the computation of P_0^{-1} follows as in Sections B.3 (for Q(K)) and B.4. This remark holds true for the computation of the summations which may appear in the analysis of this scheme. As a result, we need not carry the study of this scheme any further.

B.6 Further Numerical Results and Comparisons

In this section we intend to compare our first four sharing schemes: CP, CS, SMXQ, SMA, under the assumption of equal ρ_i's. Before we proceed, let us recall that if B is the total number of buffers ($B = RB_0$) and b is the maximum queue size (for any queue) when using an SMXQ scheme, then

1. if $b = B$ SMXQ is equivalent to CS
2. if $b = B_0$ SMXQ is equivalent to CP
3. if $R = 2$ then SMXQ is equivalent to SMA with a minimum allocation per queue equal to $B - b$.

As a result of the above, the study of SMXQ with R = 2 and a variable b will allow us to cover the four sharing schemes to be considered here.

In the numerical example below we assume that R = 2, B = 6 and that b satisfies $B/2 \leq b \leq B$ (see Eq.(B.49), i.e., b = 3,4,5,6. From our previous considerations we know that b = 3 leads to CP, b = 4 and b = 5 lead to non-degenerate SMXQ, SMA, and b = 6 leads to CS.

Figs. B.10, B.11 and B.12 show respectively the probability of blocking PB, the channel utilization $\rho(1 - PB)$, the normalized average message delay μCT, obtained with the four schemes. With respect to blocking and utilization, the optimal b (i.e., the optimal scheme) is a function of ρ. We note that for small values of ρ, b = 6 (i.e., CS) is optimal; as ρ increases b = 5, then b = 4 (i.e., (SMXQ, SMA) become optimal, and finally, for a larger ρ, b = 3 (i.e., CP) becomes optimal. With respect to the average delay, it is an increasing function of b. There is, consequently, a tradeoff between the probability of blocking and the system delay. Therefore, the selection of a particular scheme should account for these two variables as well as the load on the system.

B.7 Conclusion

In this study, we considered various schemes for sharing a pool of buffers among a set of communication channels in a communication network environment. Five sharing schemes are examined and the results of the analysis are presented and displayed in a fashion which permits one to establish the tradeoffs among blocking probability, utilization, throughput and delay.

We found that no one scheme is always optimal; one should select a scheme to fit the particular operational environment. This study shows that, in general, sharing with some restrictions on the contention of space is certainly more advantageous than no-sharing, especially when little storage is available.

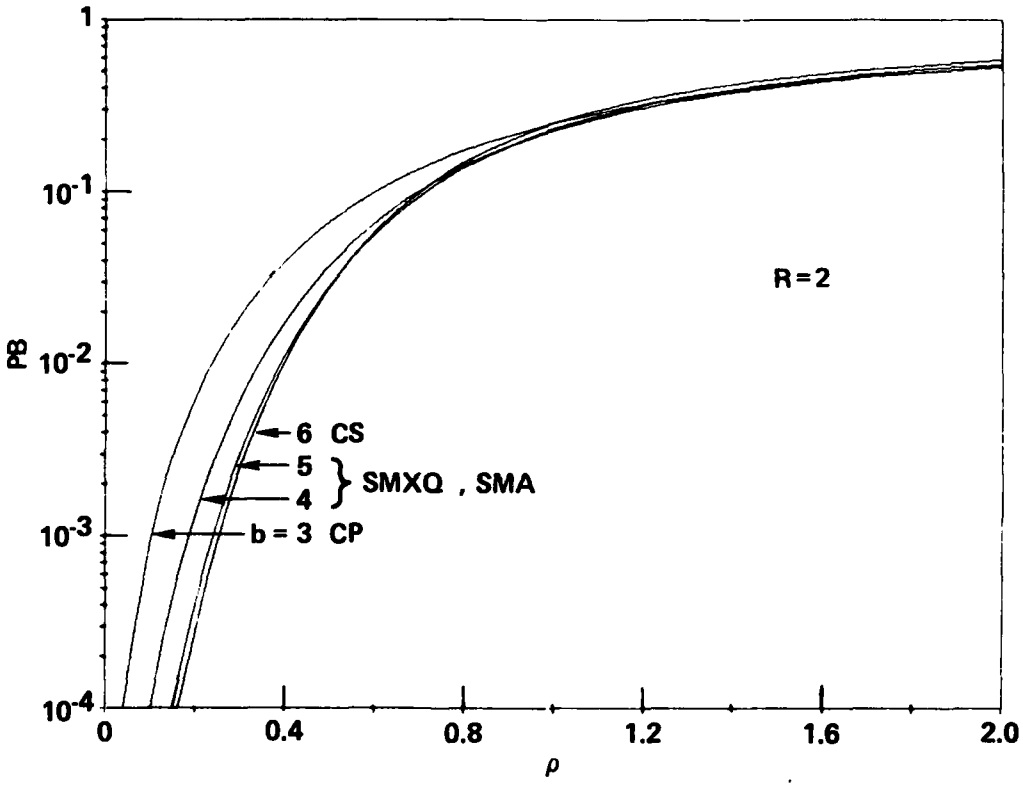


Figure B.10. Comparison of the Four Schemes : Blocking.

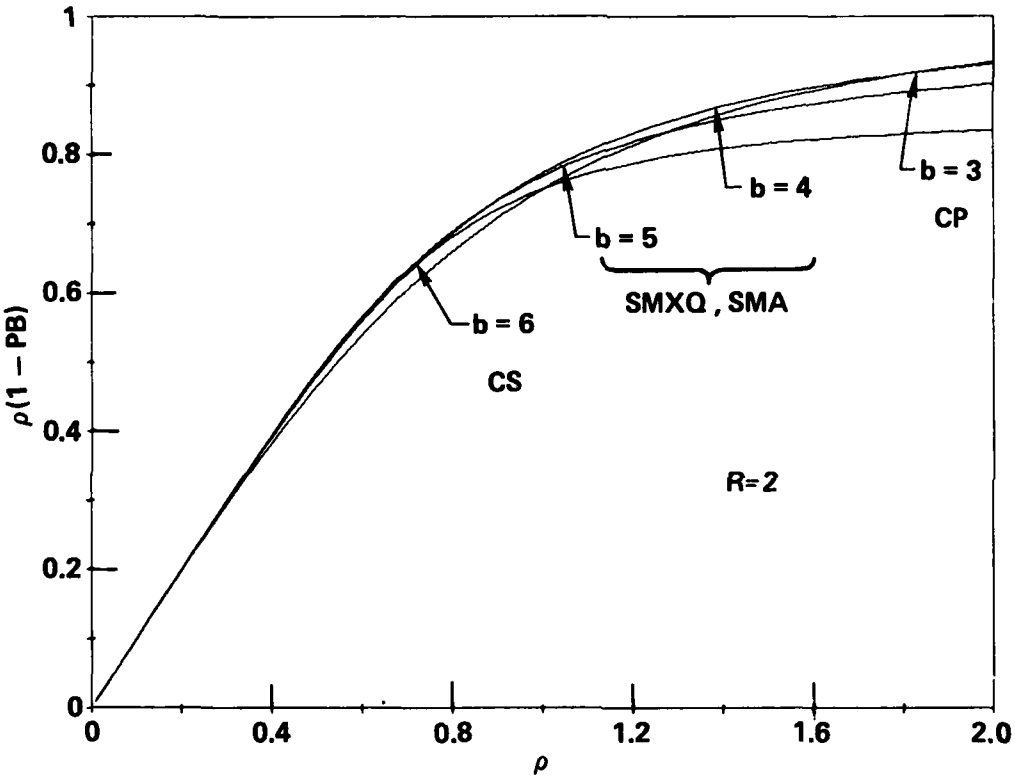


Figure B.11. Comparison of the Four Schemes : Utilization.

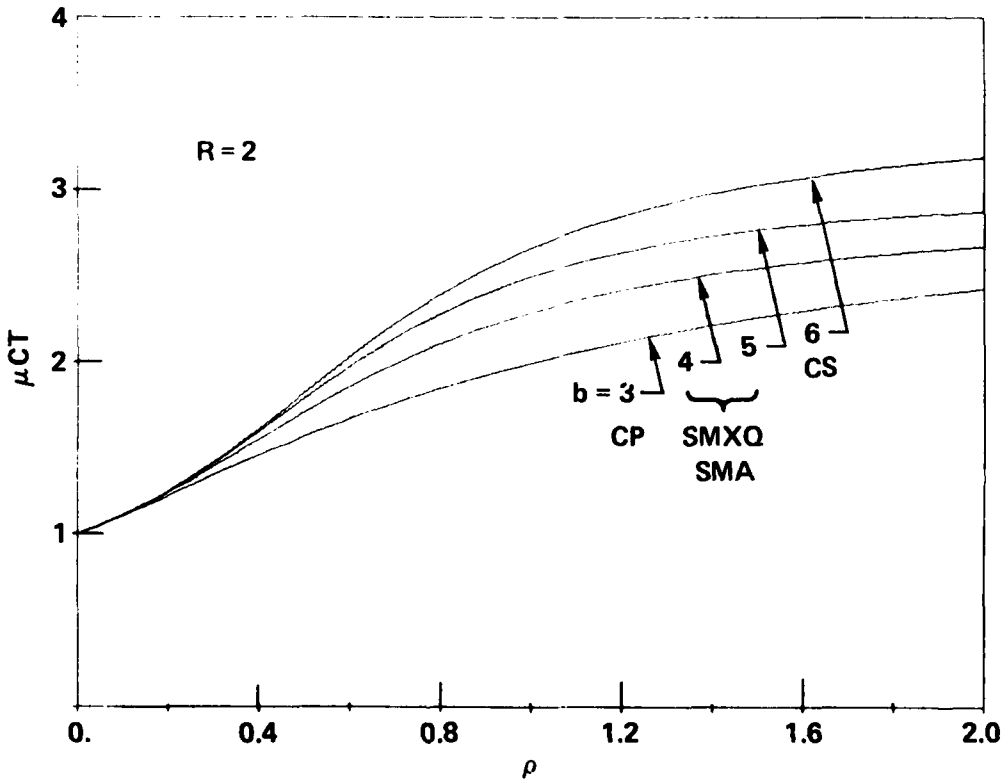


Figure B.12. Comparison of the Four Schemes : Delay.

APPENDIX C

PROPOSITION 6.1 AND ITS PROOF:

APPLICATION TO SOME SPECIAL CASES

C.1 Optimal Clustering Structure

Proposition 6.1

Given m, the number of levels in the hierarchy, and assuming that $\alpha_i > 1$ for all $i = 1, \ldots, m$, then the solution of the optimization problem 6.11, i.e., the optimal clustering structure, is such that:

(a) All clusters at the same level, $k = 1, \ldots, m$, are composed of an equal number of lower level clusters, i.e., all nodes at the same level in the tree representation are of equal degree. The optimal degree vector reduces to an m-dimensional vector, $\underset{\sim}{n} = (n_1, n_2, \ldots, n_m)$, whose components are the solution of the following set of difference equations:

$$n_k(i_m, i_{m-1}, \ldots, i_{k+1}) = n_k \qquad \text{for all } k = 1, \ldots, m \text{ and } (i_m, \ldots, i_{k+1})$$

$$\begin{cases} n_1 = \dfrac{N}{n_2 n_3 \cdots n_m} \\[2ex] n_k = \left[\dfrac{\prod_{i=1}^{k-1} (\alpha_i - 1)}{\alpha_k (\beta_k)^{\alpha_k}} \; \dfrac{B_k}{D_k} \left[\prod_{i=k+1}^{m} n_i \right]^{-\frac{\prod_{i=1}^{k-1} \alpha_i}{D_k}} \right]^{\frac{D_k}{D_{k+1}}} \end{cases} \qquad (6.12)$$

$$k = 2, 3, \ldots, m$$

where by convention $\prod_{i=m+1}^{m} n_i = 1$

and D_k is the solution of

$$\begin{cases} D_2 = 1 \\ D_k = \alpha_{k-1} D_{k-1} + \prod_{i=1}^{k-2} (\alpha_i - 1) \qquad k \geqq 3 \end{cases} \tag{6.13}$$

Also, B_k is the solution of

$$\begin{cases} B_2 = N^{\alpha_1} \\ \left[\frac{B_k}{D_k}\right]^{D_k} = \alpha_{k-1}^{-\alpha_{k-1} D_{k-1}} \left[\frac{\prod_{i=1}^{k-2} (\alpha_1 - 1)}{(\beta_{k-1})^{\alpha_{k-1}}}\right]^{-\prod_{i=1}^{k-2} (\alpha_i - 1)} \left[\frac{B_{k-1}}{D_{k-1}}\right]^{\alpha_{k-1} D_{k-1}} \end{cases} \tag{6.14}$$

$$k \geqq 3$$

(b) With this optimum solution, the minimum computational cost is:

$$\overline{G}(m, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = B_{m+1} \tag{6.15}$$

During the course of the proof, we will first fix a set of variables and carry the optimization step over the subset of non-fixed variables. Then we will replace the optimal values of the non-fixed variables in the objective function and repeat the procedure until we exhaust all of the variables. Note that the optimal values of the non-fixed variables will be expressed in terms of the fixed variables. Step-by-step, the above strategy will proceed as follows:

1. $k \leftarrow 1$
2. Fix all variables $n_j(i_m, \ldots, i_{j+1}) \quad \forall\, j = k+1, \ldots, m$
   Solve the optimization problem with respect to the variables $n_k(i_m, \ldots, i_{k+1})$

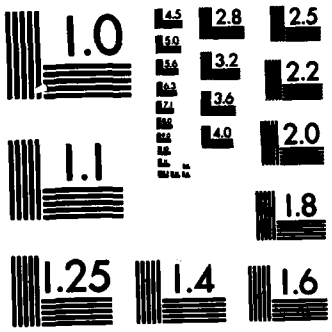
AD-A122 995 DESIGN CONSIDERATIONS FOR LARGE COMPUTER COMMUNICATION NETWORKS(U) CALIFORNIA UNIV LOS ANGELES SCHOOL OF ENGINEERING AND APPLIED SCIENCE F KAMOUN APR 76 5/5

UNCLASSIFIED UCLA-ENG-7642 DAHC15-73-C-0368 F/G 9/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

3. Replace the $n_k(\ )$'s by their optimal values in the objective function.

4. If $k = m$ Stop

   Else $k \leftarrow k + 1$ Go to Step 2

Note that when $k = m$, there are no fixed variables, and the optimization is carried out with respect to n_m.

To introduce some of the ideas incorporated in the general proof, we shall first solve for $m = 2$.

<u>Optimality for m = 2</u>

The optimization problem becomes:

$$\begin{cases} \text{min:} \quad G(2, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = \sum_{i_2=1}^{n_2} [n_1(i_2)]^{\alpha_1} + [\beta_2 n_2]^{\alpha_2} \\ \text{over:} \quad \underset{\sim}{n} \\ \text{constraint:} \quad \sum_{i_2=1}^{n_2} n_1(i_2) = N \ ; \qquad \underset{\sim}{n} \geq 0 \end{cases}$$

<u>Claim C.1</u>

According to Proposition 6.1, the solution of the above problem must be such that:

$$\begin{cases} n_1(i_2) = n_1 = \dfrac{N}{n_2} \qquad \forall \ i_2 = 1, \ldots, n_2 \\ n_2 = \left[\dfrac{\alpha_1 - 1}{\alpha_2 (\beta_2)^{\alpha_2}} N^{\alpha_1} \right]^{\frac{1}{\alpha_1 + \alpha_2 - 1}} \end{cases}$$

Hence,

$$\alpha_1 [n_1(i_2)]^{\alpha_1 - 1} - \lambda = 0 \Rightarrow n_1(i_2) = \left[\frac{\lambda}{\alpha_1}\right]^{\frac{1}{\alpha_1 - 1}}$$

$$\forall\, i_2 = 1, \ldots, n_2$$

which means that all $n_1(i_2)$'s are equal; therefore, from the size constraint:

$$n_1(i_2) = \frac{N}{n_2} \qquad \forall\, i_2 = 1, \ldots, n_2 \qquad \text{(notice } n_1(i_2) > 0\text{)}$$

<u>Step 2</u>

n_2 is now a variable, and the problem becomes:

$$\begin{cases} \min G = N^{\alpha_1}(n_2)^{1-\alpha_1} + (\beta_2 n_2)^{\alpha_2} \\ \text{over:} \quad n_2 \geq 0 \end{cases}$$

Since α_1, α_2 are strictly greater than one, the objective is a convex function in the region $(n_2 > 0)$. Then, if n_2 solution of $\frac{dG}{dn_2} = 0$ is such that $n_2 > 0$, it must be the optimal solution.

$$\frac{dG}{dn_2} = 0 \Rightarrow (1 - \alpha_1)(N)^{\alpha_1}(n_2)^{-\alpha_1} + \alpha_2(\beta_2)^{\alpha_2}(n_2)^{\alpha_2 - 1} = 0$$

Hence,

$$n_2 = \left[\frac{\alpha_1 - 1}{\alpha_2(\beta_2)^{\alpha_2}} N^{\alpha_1}\right]^{\frac{1}{\alpha_1 + \alpha_2 - 1}}$$

Notice $n_2 > 0$.

By replacing n_2 in the objective function,

$$\overline{G} = (n_2)^{1-\alpha_1}\left[(N)^{\alpha_1} + (\beta_2)^{\alpha_2}(n_2)^{\alpha_1 + \alpha_2 - 1}\right]$$

$$\min G(2, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = (\alpha_1 + \alpha_2 - 1)$$

$$\times \left[\alpha_2^{-\alpha_2} \left[\frac{\alpha_1 - 1}{(\beta_2)^{\alpha_2}} \right]^{-(\alpha_1 - 1)} \left[N^{\alpha_1} \right]^{\alpha_2} \right]^{\frac{1}{\alpha_1 + \alpha_2 - 1}}$$

<u>Proof</u>:

Following the strategy outlined above, the proof will proceed in two steps:

<u>Step 1</u>

n_2 is fixed. Solve the problem with respect to $n_1(i_2)$, $i_2 = 1, 2, \ldots, n_2$.

The objective function, being a sum of power functions $[n_1(i_2)]^{\alpha_1}$, which are convex in the region $\{n_1(i_2) > 0$, for all i's$\}$, is a convex function in that region (recall $\alpha_1 > 1$). Taking the Langrangian:

$$L(\underset{\sim}{n}_1, \lambda) = \sum_{i_2=1}^{n_2} [n_1(i_2)]^{\alpha_1} - \lambda \left[\sum_{i_2=1}^{n_2} n_1(i_2) - N \right]$$

where $\underset{\sim}{n}_1 \triangleq \{n_1(i_2)\}$ and $\lambda \geq 0$.

Notice that we discarded the constant term $(\beta_2 n_2)^{\alpha_2}$ and the positivity constraint, which we will check a posteriori.

The Langrangian is also a convex function (for $n_1(i_2) > 0$), and at optimality $\underset{\sim}{n}_1$ must be such that:

$$\frac{\partial L}{\partial n_1(i_2)} = 0 \qquad \forall\, i_2 = 1, \ldots, n_2$$

$$\overline{G} = (n_2)^{1-\alpha_1}\left(\frac{\alpha_2 + \alpha_1 - 1}{\alpha_2}\right)(N)^{\alpha_1}$$

$$= \left[\frac{\alpha_1 - 1}{\alpha_2(\beta_2)^{\alpha_2}}(N)^{\alpha_1}\right]^{\frac{1-\alpha_1}{\alpha_1+\alpha_2-1}} \frac{\alpha_1 + \alpha_2 - 1}{\alpha_2}(N)^{\alpha_1}$$

which, after grouping the terms in α_2 at the denominator, results in the expression proposed by Claim C.1.

General Case

In accordance with the step-by-step strategy, we will first prove a lemma that provides the solution for the optimization problem obtained after k steps.

Lemma C.1

The optimal solution of the optimization problem obtained after fixing the variables $n_j(i_m, \ldots, i_{j+1})$ for $j = k + 1, \ldots, m$ is such that

$$n_\ell(i_m, \ldots, i_{\ell+1}) = n_\ell \qquad \forall\, (i_m, \ldots, i_\ell);\ \ell = 1, \ldots, k$$

$$\begin{cases} n_\ell = \left[\dfrac{\prod_{i=1}^{\ell-1}(\alpha_i - 1)}{\alpha_\ell(\beta_\ell)^{\alpha_\ell}} \dfrac{B_\ell}{D_\ell}\left[\sum_{i_m=1}^{n_m}\cdots\sum_{i_{\ell+2}=1}^{n_{\ell+2}(\cdot)} n_{\ell+1}(i_m, \ldots, i_{\ell+2})\right]^{-\prod_{i=1}^{\ell-1}\alpha_i}\right]^{\frac{D_\ell}{D_{\ell+1}}} \\ \qquad\qquad \forall\, \ell = 2, 3, \ldots, k \\ n_1 = N\left[n_2 n_3 \cdots n_k \sum_{i_m=1}^{n_m}\cdots\sum_{i_{k+2}=1}^{n_{k+2}(\cdot)} n_{k+1}(i_m, \ldots, i_{k+2})\right]^{-1} \end{cases} \tag{C.1}$$

and the minimum G is expressed by:

$$\overline{G}(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = B_{k+1}\left[\sum_{i_m} \cdots \sum_{i_{k+2}} n_{k+1}(i_m, \ldots, i_{k+2})\right]^{-\frac{\prod_{i=1}^{k}(\alpha_i - 1)}{D_{k+1}}}$$

$$+ \sum_{i_m=1}^{n_m} \cdots \sum_{i_{K+2}} [\beta_{k+1} n_{k+1}(i_m, \ldots, i_{k+2})]^{\alpha_{k+1}}$$

$$+ \sum_{j=k+2}^{m} G_k(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) \qquad \text{(C.2)}$$

<u>Proof</u>:

The proof will proceed by induction. Let us prove that Lemma C.1 is true for k = 1.

(i) <u>k = 1</u>: All variables, $n_j(\ldots)$ $j = 2, \ldots, m$, are fixed. Thus the optimization problem (6.11) reduces to:

$$\min G_1 = \sum_{i_m=1}^{n_m} \cdots \sum_{i_2=1}^{n_2(i_m, \ldots, i_3)} [n_1(i_m, \ldots, i_2)]^{\alpha_1}$$

over: $n_1(i_m, \ldots, i_2) > 0$

s.t.: size constraint

Since $\alpha_1 > 1$, the above objective is convex in the region n_1's > 0. Taking the Lagrangian, we get

$$L(\underset{\sim}{n}_1, \lambda) = G_1 - \lambda \left(\sum_{i_m} \cdots \sum_{i_2} n_1(i_{m+1}, \ldots, i_2) - N \right)$$

In the region, $\{\underset{\sim}{n}_1 > 0\}$, the Langrangian is a convex function, and if the solution of $\nabla L_1 = 0$, where ∇ denotes the gradient operator [ZAN 69], is such that $\underset{\sim}{n}_1 > 0$, then it must be optimal.

$$\nabla L_1 = 0 \implies \frac{\partial L}{\partial n_1(i_m, \ldots, i_2)} = 0 \qquad \forall\, (i_m, \ldots, i_2)$$

By equating the partial derivatives to zero:

$$\alpha_1 [n_1(i_m, \ldots, i_2]^{\alpha_1 - 1} = \lambda \qquad \forall\, (i_m, \ldots, i_2)$$

This implies that all n_1's must be <u>equal</u>. Hence, from the size constraint,

$$n_1(i_m, \ldots, i_2) = n_1 = \frac{N}{\sum_{i_m} \cdots \sum_{i_3} n_2(i_m, \ldots, i_3)}$$

By replacing n_1 in the objective function,

$$\overline{G} = N^{\alpha_1} \Big[\sum_{i_m} \cdots \sum_{i_3} n_2(i_m, \ldots, i_3) \Big]^{1-\alpha_1} + \sum_{j=2}^{m} G_k(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta})$$

Since $B_2 = N^{\alpha_1}$ and $D_2 = 1$, the above expressions satisfy Lemma C.1.

(ii) <u>General Case</u>

Assuming that Equations (C.1) and (C.2) are true for $\ell = 1, 2, \ldots, k$, with $k < m$, let us prove that they are still true for $\ell = k + 1$.

According to the step-by-step strategy, we propose to solve for step $k + 1$. Let all $n_j(i_m, \ldots, i_{j+1})$ be fixed $\forall\, j = k + 2, \ldots, m$ and let us solve for

$$\text{min:} \quad \overline{G}(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) \qquad \text{as given in Eq. (C.2)}$$

$$\text{over:} \quad \{n_{k+1}(i_m, \ldots, i_{k+2})\} \overset{\Delta}{=} \underset{\sim}{n}_{k+1} > 0.$$

Notice that the term, $\sum_{j=k+2}^{m} G_k(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta})$ in Eq. (C.2) is a constant;

therefore, it will be discarded in the optimization step.

Claim C.2

$\overline{G}(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta})$ is convex, with respect to $\underset{\sim}{n}_{k+1}$, in the region $\underset{\sim}{n}_{k+1} > 0$. It is sufficient to prove that Y, the first term of Eq. (C.2), is a convex function. (In what follows, $n_k(i_m, i_{m-1}, \ldots, i_{k+1})$ occasionally will be denoted by $n_k(\cdot)$.)

$$\frac{\partial Y}{\partial n_{k+1}(i_m, \ldots, i_{k+2})} = - \frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}} B_{k+1} \times \left[\sum_{i_m} \cdots \sum_{i_{k+2}} n_{k+1}(\cdot)\right]^{-\frac{\prod_{i=1}^{k}(\alpha_i - 1)}{D_{k+1}} - 1}$$

and

$$\frac{\partial^2 Y}{\partial [n_{k+1}(i_m, \ldots, i_{k+2})]^2} = \frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}} \left[\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}} + 1\right] B_{k+1} \times \left[\sum \cdots \sum n_{k+1}(\cdot)\right]^{-\frac{\prod_{i=1}^{k}(\alpha_i - 1)}{D_{k+1}} - 2}$$

which we define as $\partial^2 Y$.

The same expression is true for

$$\frac{\partial^2 Y}{\partial n_{k+1}(i_m, \ldots, i_{k+2}) \partial n_{k+1}(i'_m, \ldots, i'_{k+2})}$$

Consequently, the Hessian matrix corresponding to Y is:

$$H_Y = [\partial^2 Y] \begin{bmatrix} 1 & 1 & . & . & . & 1 \\ 1 & 1 & . & . & . & 1 \\ . & . & . & . & . & . \\ 1 & 1 & . & . & . & 1 \end{bmatrix}$$

H_Y is equal to $\partial^2 Y$ (defined above) times a matrix of ones's. Let us prove that H_Y is positive definite in the region $\underset{\sim}{n}_{k+1} > 0$.

$$\underset{\sim}{n}_{k+1} H_Y \underset{\sim}{n}^t_{k+1} = \partial^2 Y \left[\sum \cdots \sum n_{k+1}(i_m, \ldots, i_{k+2}) \right]^2 \qquad \forall\, \underset{\sim}{n}_{k+1}$$

where $\underset{\sim}{n}^t_{k+1}$ is the transpose of $\underset{\sim}{n}_{k+1}$. For $\underset{\sim}{n}_{k+1} > 0$, $\partial^2 Y > 0$.

Therefore, $\underset{\sim}{n}_{k+1} H_Y \underset{\sim}{n}^t_{k+1} > 0 \qquad \forall\, \underset{\sim}{n}_{k+1} > 0$.

This proves that Y is a convex function in the region of interest; moreover, since the second term of Eq. (C.2) is a sum of convex functions, then $G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta})$ is a convex function.

As a consequence of Claim C.2, if the solution, $\underset{\sim}{n}_{k+1}$ of $\nabla G = 0$, is such that $\underset{\sim}{n}_{k+1} > 0$, then it must be the optimal solution.

$$\nabla \overline{G} = 0 \implies \frac{\partial \overline{G}}{\partial n_{k+1}(i_m, \ldots, i_{k+2})} = 0 \qquad \forall\, i_m, \ldots, i_{k+2}$$

Hence

$$\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}} B_{k+1} \left[\sum_{i_m} \cdots \sum_{i_{k+2}} n_{k+1}(i_m, \ldots, i_{k+2}) \right]^{-\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}} - 1}$$

$$= \alpha_{k+1} \beta_{k+1}^{\alpha_{k+1}} \left[n_{k+1}(i_m, \ldots, i_{k+2}) \right]^{\alpha_{k+1} - 1} \qquad \forall\, (i_m, \ldots, i_{k+2})$$

Notice that the left-hand side of this equation is the summation of all $n_{k+1}(i_m, \ldots, i_{k+2})$; hence, it will be the same for all $(i_m, \ldots, i_{k+2})$.

Consequently, all n_{k+1}'s are equal. Let $n_{k+1}(i_m, \ldots, i_{k+2}) = n_{k+1}$.

From the above equation

$$\alpha_{k+1}\beta_{k+1}^{\alpha_{k+1}}(n_{k+1})^{\alpha_{k+1}-1} = \prod_{i=1}^{k} (\alpha_i - 1) \frac{B_{k+1}}{D_{k+1}}$$

$$\times \left[n_{k+1} \sum_{i_m} \cdots \sum_{i_{k+3}} n_{k+2}(\cdot) \right]^{-\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}} - 1}$$

Grouping the terms in n_{k+1} to the left-hand side leads to n_{k+1} with the following exponent (to be denoted: exp [ ]).

$$\exp\,[n_{k+1}] = \alpha_{k+1} + \frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}} = \frac{\alpha_{k+1}D_{k+1} + \prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}}$$

Using Eq. (6.2)

$$\exp\,[n_{k+1}] = \frac{D_{k+2}}{D_{k+1}}$$

Also, the exponent of the multiple summation is equal to

$$-\frac{D_{k+1} + \prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}} = -\frac{\prod_{i=1}^{k} \alpha_i}{D_{k+1}}$$

This is due to the fact that:

$$D_{k+1} + \prod_{i=1}^{k} (\alpha_i - 1) = \prod_{i=1}^{k} \alpha_i \qquad \text{(C.3)}$$

The proof of Eq. (C.3) results from Eq. (6.13) as follows:

$$D_{k+1} + \prod_{i=1}^{k} (\alpha_i - 1) = \alpha_k \left[D_k + \prod_{i=1}^{k-1} (\alpha_i - 1) \right]$$

$$= \alpha_k \alpha_{k-1} \left[D_{k-1} + \prod_{i=1}^{k-2} (\alpha_i - 1) \right]$$

$$= \alpha_k \alpha_{k-1} \cdots \alpha_2 (D_2 + \alpha_1 - 1)$$

$$= \prod_{i=1}^{k} \alpha_i \qquad \text{Q.E.D}$$

Finally,

$$\alpha_{k+1} \beta_{k+1}^{\alpha_{k+1}} [n_{k+1}]^{\frac{D_{k+2}}{D_{k+1}}} = \prod_{i=1}^{k} (\alpha_i - 1) \frac{B_{k+1}}{D_{k+1}} \left[\sum_{i_m} \cdots \sum_{i_{k+3}} n_{k+2}(\cdot) \right]^{-\frac{\prod_{i=1}^{k} \alpha_i}{D_{k+1}}}$$

Extracting n_{k+1} would give Eq. (C.1) for $\ell = k + 1$. Notice that $\underset{\sim}{n}_{k+1} > 0$. Now, replacing $n_{k+1}(i_m, \ldots, i_{k+2})$ by n_{k+1} in the objective function, Eq. (C.2), we arrive at

$$\overline{G} = B_{k+1} \left[n_{k+1} \sum_{i_m} \cdots \sum_{i_{k+3}} n_{k+2}(i_m, \ldots, i_{k+3}) \right]^{-\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}}}$$

$$+ \beta_{k+1}^{\alpha_{k+1}} n_{k+1}^{\alpha_{k+1}} \sum_{i_m} \cdots \sum_{i_{k+3}} n_{k+2}(i_m, \ldots, i_{k+3})$$

$$+ \sum_{j=k+2}^{m} G_k(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta})$$

Replacing n_{k+1} above with its expression given in Eq. (C.1), leads to:

$$\overline{G} = B_{k+1}\left[\frac{\prod_{i=1}^{k}(\alpha_i - 1)}{\alpha_{k+1}(\beta_{k+1})^{\alpha_{k+1}}}\ \frac{B_{k+1}}{D_{k+1}}\right]^{-\frac{\prod_{i=1}^{k}(\alpha_i - 1)}{D_{k+2}}}\left[\sum_{i_m}\cdots\sum_{i_{k+3}} n_{k+2}(\cdot)\right]^X$$

$$+\ \beta_{k+1}^{\alpha_{k+1}}\left[\frac{\prod_{i=1}^{k}(\alpha_i - 1)}{\alpha_{k+1}(\beta_{k+1})^{\alpha_{k+1}}}\ \frac{B_{k+1}}{D_{k+1}}\right]^{\alpha_{k+1}\frac{D_{k+1}}{D_{k+2}}}\left[\sum_{i_m}\cdots\sum_{i_{k+3}} n_{k+2}(\cdot)\right]^Y$$

$$+\ \sum_{j=k+2}^{m} G_k(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta})$$

The exponents X, Y are such that

$$X = -\frac{\prod_{i=1}^{k}(\alpha_i - 1)}{D_{k+1}} + \left(1 + \frac{\prod_{i=1}^{k}(\alpha_i - 1)}{D_{k+1}}\right)\frac{\prod_{i=1}^{k}(\alpha_i - 1)}{D_{k+2}}$$

$$= \frac{\prod_{i=1}^{k}(\alpha_i - 1)}{D_{k+1}D_{k+2}}\left[D_{k+1} + \prod_{i=1}^{k}(\alpha_i - 1) - D_{k+2}\right]$$

Also, $$D_{k+2} = \alpha_{k+1}D_{k+1} + \prod_{i=1}^{k}(\alpha_i - 1)$$

Thus, $$X = -\frac{\prod_{i=1}^{k+1}(\alpha_i - 1)}{D_{k+2}}$$

As for Y, we have

$$Y = 1 - \left(1 + \frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+1}}\right)^{\alpha_{k+1}} \frac{D_{k+1}}{D_{k+2}}$$

By performing the same manipulations as for X,

$$Y = - \frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+2}}$$

Consequently, X = Y; this property is the <u>key</u> to the proof. It says that at each optimization step we get one less term in the objective function.

To prove that Eq. (C.2) is satisfied for $\ell = k + 1$, there remains to be shown that the coefficient of

$$\left[\sum \cdots \sum n_{k+2}(i_m, \ldots, i_{k+3})\right]^X$$

is equal to B_{k+2}. Let U be that coefficient, then

$$U = \left[\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{\alpha_{k+1}(\beta_{k+1})^{\alpha_{k+1}}} \frac{B_{k+1}}{D_{k+1}}\right]^{\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+2}}}$$

$$\times \left[B_{k+1} + \beta_{k+1}^{\alpha_{k+1}} \left[\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{\alpha_{k+1}(\beta_{k+1})^{\alpha_{k+1}}} \frac{B_{k+1}}{D_{k+1}}\right]^{Z}\right]$$

where $$Z = \alpha_{k+1} \frac{D_{k+1}}{D_{k+2}} + \frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+2}} = 1$$

Then, the term in the second bracket becomes

$$B_{k+1} + \frac{\prod_{i=1}^{k} (\alpha_i - 1)}{\alpha_{k+1}} \frac{B_{k+1}}{D_{k+1}} = \frac{D_{k+2}}{\alpha_{k+1}} \frac{B_{k+1}}{D_{k+1}}$$

After substitution,

$$U = D_{k+2} \left[\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{(\beta_{k+1})^{\alpha_{k+1}}} \right]^{-\frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+2}}} \left[\frac{B_{k+1}}{\alpha_{k+1} D_{k+1}} \right]^{1 - \frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+2}}}$$

but

$$1 - \frac{\prod_{i=1}^{k} (\alpha_i - 1)}{D_{k+2}} = \frac{\alpha_{k+1} D_{k+1}}{D_{k+2}}$$

Therefore, $U = B_{k+2}$ as given in Eq. (6.14). This terminates the proof of Lemma C.1.

Lemma C.1 implicity assumed that $k \leq m - 1$. Let us show that Eqs. (C.1) and (C.2) are still true for $\ell = k = m$. Observe that for $\ell = k = m - 1$

$$\sum_{i_m=1}^{n_m} \cdots \sum_{i_{\ell+2}=1}^{n_{\ell+2}(i_m, \ldots, i_{\ell+3})} n_{\ell+1}(i_m, \ldots, i_{\ell+2}) = n_m$$

Furthermore, by convention, we set the value of the above sum to be 1 when $\ell = m$. Consequently, Eqs. (C.1) and (C.2) become meaningful for $\ell = m$. There remains to be shown that these equations effectively hold true at that value of $\ell = m$.

For $\ell = k = m - 1$, Eq. (C.2) becomes:

$$\overline{G}(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = B_m [n_m]^{-\frac{\prod_{i=1}^{m-1} (\alpha_i - 1)}{D_m}} + [\beta_m n_m]^{\alpha_m} \qquad (C.4)$$

The optimization will now be performed over the <u>single</u> variable, n_m. $\overline{G}$, as given in Eq. (C.4), is convex in the region $\{n_m > 0\}$. Again, if the solution n_m of $\nabla G = 0$ is such that $n_m > 0$, then it must be the optimal solution.

$$\frac{\partial \overline{G}}{\partial n_m} = 0 \implies \alpha_m (\beta_m)^{\alpha_m} [n_m]^{\alpha_{m-1}} = \left[\prod_{i=1}^{m} (\alpha_i - 1) \right] \frac{B_m}{D_m} [n_m]^{-\frac{\prod_{i-1}^{m} (\alpha_i - 1)}{D_m} - 1}$$

Solving for n_m, $$n_m = \left[\frac{\prod_{i=1}^{m} (\alpha_i - 1)}{\alpha_m (\beta_m)^{\alpha_m}} \frac{B_m}{D_m} \right]^{\frac{D_m}{D_{m+1}}}$$

This result satisfies Eq. (C.1) for $\ell = k = m$, where the summation is set ot one.

Also we can easily check that the objective function becomes

$$\overline{G}(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\beta}) = B_{m+1} \qquad (C.5)$$

since it is the last step in the optimization. The above objective is actually the minimum computational cost. It also checks with the expression in Eq. (C.2) where the summations are set to one.

<u>End of Proof of Proposition 6.1</u>

From Lemma C.1 and the extension of the expresssion in Eq. (C.1) to $\ell = m$, we know that

$$n_\ell(i_m, \ldots, i_{\ell+1}) = n_\ell \qquad \forall\, (i_m, \ldots, i_{\ell+1}) \qquad \forall\, \ell = 1, \ldots, m$$

Thus, the multiple summation in Eq. (C.1) becomes,

$$\sum_{i_m=1}^{n_m} \cdots \sum_{i_{\ell+2}=1}^{n_{\ell+2}} n_{\ell+1}(i_m, \ldots, i_{\ell+2}) = n_m n_{m-1} \cdots n_{\ell+1} = \prod_{i=\ell+1}^{m} n_i$$

The substitution of the above summations in Eq. (C.1) results in Eq. (6.12). Also, Eq. (C.5), giving minimum G, is exactly what we have to prove, i.e., Eq. (6.15).

C.2 Application to Other Special Cases

This section deals with the following two cases:

(i) Uniform design strategy, variable gate assignment, i.e.,

$$\alpha_k = \alpha \qquad \forall\, k = 1, \ldots, m$$

$$\beta_k\text{'s variables} \tag{C.6}$$

(ii) Proportional assignment of gates.

In (i), explicit expressions for the optimal degree vector and computational cost, given m, have been derived. Some partial results related to the global optimality have been found when β_k is of the form, $\beta_k = \gamma^{k-2}$, $k \geq 2$.

With respect to (ii), the number of kth level gates to be selected is proportional to the number of $k\text{-}1^{st}$ level gates from which they are selected. The corresponding solution was found to be of no practical interest. The solution is

$$\begin{cases} n_1 = 0, \qquad n_m = +\infty \\ n_2, \ldots, n_{m-1} \quad \text{any value different from 0 or } \infty, \text{ s.t. } \prod_{i=1}^{m} n_i = N. \end{cases}$$

C.2.1 Uniform Design Strategy, Variable Gate Assignment

The purpose of this section is to model design requirements whereby different reliability constraints are imposed on the design of the layer subnets depending on their levels in the hierarchy. However, the same design procedure is applied at all levels.

C.2.1.1 Explicit Expressions

Corollary C.1

Under the conditions of Proposition 6.1, and assuming that all the α_k's are equal, the optimal solution is such that

$$n_k = \frac{\alpha}{\alpha - 1}\left[\left(\frac{\alpha - 1}{\alpha}\right)^m N\right]^{\frac{(\alpha-1)^{m-k}\alpha^{k-1}}{D_{m+1}}} \times \left[\frac{\prod_{i=k+1}^{m} (\beta_i)^{\alpha^{m+k-i}(\alpha-1)^{i-k-1}} \prod_{i=1}^{k-1} (\beta_i)^{\alpha^{k-i}(\alpha-1)^{m-k+i-1}}}{\beta_k^{\alpha Dm}}\right]^{\frac{1}{D_{m+1}}} \qquad k = 1, 2, \ldots, m \tag{C.7}$$

With this optimal assignment, the minimum computational cost is

$$\overline{G} = D_{m+1}\left[\left(\prod_{i=2}^{m} \beta_i^{\alpha^{m+1-i}(\alpha-1)^{i-1}}\right)\left(\frac{\alpha}{\alpha - 1}\right)^{\alpha(\alpha-1)Dm}\left(\frac{(\alpha-1)^{(\alpha-1)^m}}{\alpha^{\alpha^m}}\right)^{m-1} N^{\alpha^m}\right]^{\frac{1}{D_{m+1}}} \tag{C.8}$$

Proof:

From Corollary 6.1 (Eqs. (6.21) and (6.22) we know that Eqs. (C.7) and (C.8) are true with regard to the terms containing α and N. The term containing β in Eq. (C.7) can be directly obtained from Eq. (6.19). The rest of the proof of Eq. (C.8) proceeds by induction exactly in the same way as in the proof of Eq. (6.21).

Limiting behavior when $\alpha \to \infty$

From Eqs. (6.23) and (C.7),

$$\lim_{\alpha \to +\infty} n_k = \frac{1}{\beta_k} \left[N \prod_{i=1}^{m} \beta_i \right]^{1/m} \qquad k = 1, 2, \ldots, m \tag{C.9}$$

Since $g_k = \beta_k n_k$, Eq. (C.9) implies that at the limit all layer subnets must be of equal size.

C.2.1.2 A geometrically monotonic gate assignment strategy

In order to reach any conclusion with regard to global optimality, it is necessary to specify β_k in terms of k. Let β_k be a geometrically monotonic function of k, i.e.,

$$\beta_k = \gamma^{k-2}\beta \qquad k = 2, 3, \ldots, m \tag{C.10}$$

$\gamma = 1$ corresponds to the uniform gate assignment strategy studied in Section 6.5. With the above gate assignment strategy, Eqs. (C.7) and (C.8) become:

$$\begin{cases} n_1 = \dfrac{\alpha}{\alpha - 1}\,\beta\gamma^{\alpha-1}\left[\left(\dfrac{\alpha - 1}{\alpha\gamma^{\alpha}}\right)^{m}\dfrac{N\gamma}{\beta}\right]^{\dfrac{(\alpha-1)^{m-1}}{D_{m+1}}} \\[2ex] n_k = \dfrac{\alpha}{\alpha - 1}\,\gamma^{\alpha}\left[\left(\dfrac{\alpha - 1}{\alpha\gamma^{\alpha}}\right)^{m}\dfrac{N\gamma}{\beta}\right]^{\dfrac{\alpha^{k-1}(\alpha-1)^{m-k}}{D_{m+1}}} \end{cases} \qquad k = 2, 3, \ldots, m \tag{C.11}$$

and

$$\overline{G} = D_{m+1}\left[\left(\beta\,\frac{\alpha}{\alpha - 1}\right)^{\alpha(\alpha-1)Dm}\left[\frac{(\alpha - 1)^{(\alpha-1)^{m}}}{\alpha^{\alpha^{m}}}\right]^{m-1} \times \gamma^{\alpha(\alpha-1)^{2}\left(D_m-(m-1)(\alpha-1)^{m-2}\right)}\,N^{\alpha^{m}}\right]^{\dfrac{1}{D_{m+1}}} \tag{C.12}$$

<u>Remarks</u>

(i) For

$$m = m_1 \triangleq \frac{\ln\dfrac{N\gamma}{\beta}}{\ln\dfrac{\alpha\gamma^{\alpha}}{\alpha - 1}} \tag{C.13}$$

the degree vector becomes

$$\begin{cases} n_1 = \dfrac{\alpha}{\alpha - 1}\,\beta\gamma^{\alpha-1} \\[2ex] n_k = \dfrac{\alpha}{\alpha - 1}\,\gamma^{\alpha} \qquad k = 2, \ldots, m_1 \end{cases} \tag{C.14}$$

Eq. (C.14) is quite similar to Eq. (6.30), but as we will see later, m_1 is not the optimal number of levels (except when $\gamma = 1$).

(ii) In order to satisfy the gate constraint, Eq. (6.6), the degree vector, must be such that

$$n_1 \geq \beta \quad , \quad n_k \geq \gamma \qquad k = 2, \ldots, m \tag{C.15}$$

The above condition will always be satisfied if $m \leq m_1$ and $\gamma \geq 1$. Other situations may be checked numerically.

Global Optimality

Under the conditions $N\gamma/\beta > 1$ and $\gamma > \left(\frac{\alpha - 1}{\alpha}\right)^{1/\alpha}$, the optimal number of levels, m_*, is the unique positive root of the equation:

$$m\left[\ln \frac{\alpha}{\alpha - 1}\right] \ln \frac{\alpha}{\alpha - 1} \gamma^{\alpha} = \left(\ln \frac{N\gamma}{\beta}\right)\left(\ln \frac{\alpha}{\alpha - 1}\right) + \alpha\left[1 - \left(\frac{\alpha - 1}{\alpha}\right)^{m}\right] \ln \gamma \tag{C.16}$$

The corresponding optimal degree is

$$\begin{cases} n_1 = \beta \dfrac{\alpha}{\alpha - 1} \gamma^{\alpha-1} e^{-\left(\frac{\alpha-1}{\alpha}\right)^{m_*-1} \frac{\ln \gamma}{\ln(\alpha/(\alpha-1))}} \\ n_k = \dfrac{\alpha}{\alpha - 1} \gamma^{\alpha} e^{-\left(\frac{\alpha-1}{\alpha}\right)^{m_*-k} \frac{\ln \gamma}{\ln(\alpha/(\alpha-1))}} \\ \qquad k = 2, \ldots, m_* \end{cases} \tag{C.17}$$

Notice that the expression of n_{m_*} is <u>independent</u> of N.

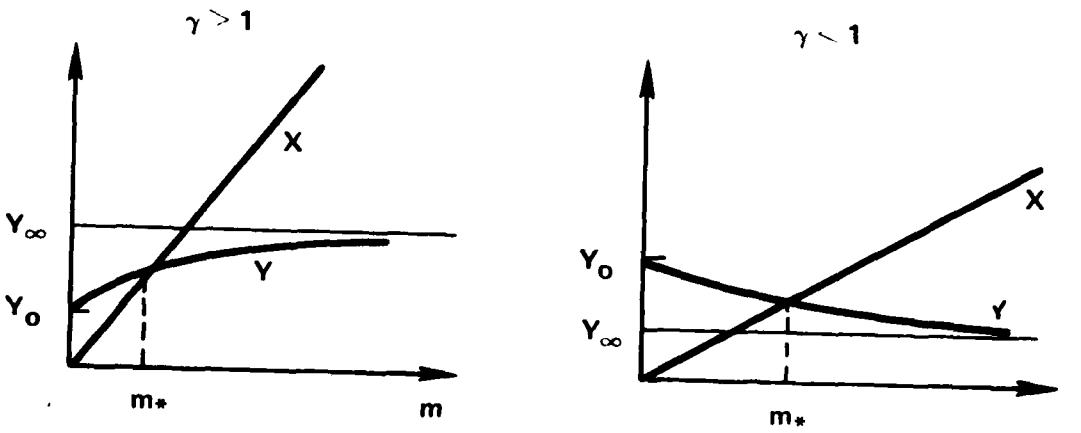
Proof·

Differentiating Eq. (C.12) with respect to m, we arrive at

$$\frac{d\overline{G}}{dm} = \frac{(\alpha - 1)^{m}\overline{G}}{D_{m+1}^{2}} \left[\alpha^{m}\left(m \ln \frac{\alpha\gamma^{\gamma}}{\alpha - 1} - \ln \frac{N\gamma}{\beta}\right) \ln \frac{\alpha}{\alpha - 1} - \alpha D_{m+1} \ln \gamma\right]$$

The roots and the sign of the above equation can be determined from the study of Eq. (C.16)· Fig. C.1 shows a straight line for the left-hand side expression of Eq. (C.16) and a monotonically increasing ($\gamma > 1$) or decreasing ($\gamma < 1$) curve for the right-hand side of Eq. (C.16)

It also shows a unique intersection point m_*. From the relative positions of those curves, we conclude that m_* leads to a minimum value of $\overline{G}$.



X = LEFT HAND SIDE OF EQ. (C.16)
Y = RIGHT HAND SIDE OF EQ. (C.16)
$Y_0 = (\text{Lg } N\gamma/\beta)\,(\text{Log } \alpha/(\alpha - 1))$
$Y_\infty = Y_0 + \alpha \text{ Log } \gamma$

Figure C.1. Comparative Behavior of the Two Sides of Eq. (C.16).

C.2.2 Proportional Gate Assignment

In certain situations, it is important to make the number of gates selected at any level sensitive (e.g., proportional) to the number of nodes of the set from which they are selected. Let $\gamma_k (\gamma_k < 1)$ be the fraction of the kth level gates to be selected as $k+1^{st}$ level gates, and let $\underset{\sim}{\gamma} \triangleq (\gamma_1, \gamma_2, \ldots, \gamma_m)$. From the above definitions, the

degree and the size vectors are such that

$$\begin{cases} g_1(i_m, \ldots i_2) = n_1(i_m, \ldots, i_2) \\ g_k(i_m, \ldots, i_{k+1}) = \sum_{i_k=1}^{n_k(i_m, \ldots, i_{k+1})} \gamma_{k-1} g_{k-1}(i_m, \ldots, i_k) \end{cases} \tag{C.18}$$

$$k = 2, \ldots, m$$

thus,

$$g_k(i_m, \ldots, i_{k+1}) = \sum_{i_k=1}^{n_k(i_m, \ldots, i_{k+1})} \cdots \sum_{i_2=1}^{n_2(i_m, \ldots, i_3)} \left(\prod_{j=1}^{k-1} \gamma_j \right) n_1(i_m, \ldots, i_2) \tag{C.19}$$

The general expression of the computational cost of all k^{th} level layer subnets is

$$G_k(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\gamma}) = \sum_{i_m=1}^{n_m} \cdots \sum_{i_{k+1}=1}^{n_{k+1}(i_m, \ldots, i_{k+2})} [g_k(i_m, \ldots, i_{k+1}]^{\alpha_k} \tag{C.20}$$

Also, Eq. (6.9) still holds true.

Substituting Eq. (C.19) into Eq. (C.20) for k = m, we find

$$G_m = \left[\sum_{i_m=1}^{n_m} \cdots \sum_{i_2=1}^{n_2(i_m, \ldots, i_3)} \left(\prod_{j=1}^{m-1} \gamma_j \right) n_1(i_m, \ldots, i_2) \right]^{\alpha_m}$$

Consequently, if $\underset{\sim}{n}$ satisfies the size constraint, Eq. (2.1)

$$G_m = \left[\left(\prod_{i=1}^{m-1} \gamma_i \right) N \right]^{\alpha_m}$$

Hence $$G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\gamma}) \geq \left[\left(\prod_{i=1}^{m-1} \gamma_i\right) N\right]^{\alpha_m} \qquad \forall\, \underset{\sim}{n} \text{ feasible} \tag{C.21}$$

Therefore, if for a particular feasible solution $\underset{\sim}{n}$, G is equal to its lower bound given above, then that vector $\underset{\sim}{n}$ is optimal.

<u>Proposition C.2</u>

Under the proportional gate assignment and for $\alpha_k > 1$, $(k = 1, \ldots, m)$ and a fixed m, an optimal solution of Problem 6.11 is $\underset{\sim}{n}^*$,

$$\begin{cases} n_1^* = 0 \\ n_m^* = +\infty \\ n_2^*, n_3^*, \ldots, n_{m-1}^* > 0, \neq \infty \qquad \text{s.t.} \prod_{i=1}^{m} n_i^* = N \end{cases} \tag{C.22}$$

With this assignment

$$\underline{G}(m, \underset{\sim}{\alpha}, \underset{\sim}{\gamma}) = \left(\prod_{i=1}^{m-1} \gamma_i\right)^{\alpha_m} N^{\alpha_m}$$

<u>Proof</u>:

Let $\underset{\sim}{n}$ be a feasible vector, and such that

$$\begin{cases} n_k(i_m, \ldots, i_{k+1}) = n_k \qquad \forall\, k = 1, \ldots, m-1 \\ \prod_{k=1}^{m} n_k = N \end{cases} \tag{C.23}$$

From Eqs. (C.19), (C.20) and (C.23), we get

$$G(m, \underset{\sim}{n}, \underset{\sim}{\alpha}, \underset{\sim}{\gamma}) = N(n_1)^{\alpha_1 - 1} + \sum_{k=2}^{m-1} \left(N \prod_{i=1}^{k-1} \gamma_i \right)^{\alpha_k} \left(\prod_{i=k+1}^{m} n_i \right)^{1-\alpha_k} + \left(\prod_{i=1}^{m-1} \gamma_i \right)^{\alpha_m} N^{\alpha_m}$$

For $\underset{\sim}{n}$ equals $\underset{\sim}{n}^*$, (Eq. (C.22))

$$\lim_{n_m \to +\infty} \left(\prod_{i=k+1}^{m} n_i \right)^{1-\alpha_k} = 0 \qquad k = 2, \ldots, m - 1$$

Hence, $$G(m, \underset{\sim}{n}^*, \underset{\sim}{\alpha}, \underset{\sim}{\gamma}) = \left(\prod_{i=1}^{m-1} \gamma_i \right)^{\alpha_m} N^{\alpha_m}$$

Then, because of Eq. (C.21), $\underset{\sim}{n}^*$ is optimal.

For a realistic network, this optimal solution is meaningless, and it is necessary either to keep the integer constraint on $\underset{\sim}{n}$ or to introduce other constraints.

FILMED

2-83

DTIC