1

4 1000 02400 000

1000 02400 000

39038

National Library
of CanadaBibliothèque nationale
du CanadaCANADIAN THESES
ON MICROFICHETHÈSES CANADIENNES
SUR MICROFICHE

NAME OF AUTHOR/NOM DE L'AUTEUR

JOHN G. GRUBER

TITLE OF THESIS/TITRE DE LA THÈSE

TRANSMISSION OF ENCODED VOICE IN A
TRANSPARENT INTELLIGENT NETWORK

UNIVERSITY/UNIVERSITÉ

CARLETON

DEGREE FOR WHICH THESIS WAS PRESENTED/
GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE

Ph.D.

YEAR THIS DEGREE CONFERRED/ANNÉE D'OBTENTION DE CE DEGRÉ

1978

NAME OF SUPERVISOR/NOM DU DIRECTEUR DE THÈSE

M.E. ULUG / B. PAGUREK

Permission is hereby granted to the NATIONAL LIBRARY OF
CANADA to microfilm this thesis and to lend or sell copies
of the film.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈ-
QUE NATIONALE DU CANADA de microfilmer cette thèse et
de prêter ou de vendre des exemplaires du film.

The author reserves other publication rights, and neither the
thesis nor extensive extracts from it may be printed or other-
wise reproduced without the author's written permission.

L'auteur se réserve les autres droits de publication; ni la
thèse ni de longs extraits de celle-ci ne doivent être imprimés
ou autrement reproduits sans l'autorisation écrite de l'auteur.

DATED/DATE

7/9/78

SIGNED/SIGNÉ

John G. Gruber

PERMANENT ADDRESS/RÉSIDENCE FIXE

6355 MATTICE AVE.

ORLEANS, ONTARIO

K1C-2G1

National Library of Canada

Cataloguing Branch
Canadian Theses Division

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada

Direction du catalogage
Division des thèses canadiennes

NOTICE

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

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

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

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

AVIS

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

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

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

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

TRANSMISSION OF ENCODED VOICE IN A
TRANSPARENT INTELLIGENT NETWORK

by

J.G. Gruber

A thesis submitted to the Faculty of
Graduate Studies in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

Faculty of Engineering
Carleton University
Ottawa, Ontario
August 1978

© 1978 John Gerald Gruber

The undersigned recommend to the Faculty of Graduate Studies acceptance of the thesis:

TRANSMISSION OF ENCODED VOICE IN A
TRANSPARENT INTELLIGENT NETWORK

submitted by J. G. Gruber, B.Sc.E., M.Sc.E. in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

L. Allen

Dean of Engineering

J. S. Riordan

Chairman,
Department of Systems Engineering,
and Computing Science

Carleton University

ABSTRACT

The objective of this thesis is to investigate certain aspects of transmitting encoded voice with data in a shared network environment. The investigation is primarily experimental and to some extent analytical, and the shared network, called TI-NET, uses a transparent intelligent approach. The motivation for this approach is to provide efficient sharing of network resources for both voice and data traffic, and at the same time to minimize end-to-end delays, particularly variable delays, to which voice traffic is known to be most susceptible.

The thesis begins by discussing the nature of the problem, which is related to the effects of delay on voice transmission. This is followed by a review of speech analysis/synthesis techniques, and approaches to integrated voice and data networks. In terms of efficiency and delays, the limitations of some of these approaches suggest the suitability of a transparent intelligent network.

The major emphasis in the thesis is in analysing, implementing and evaluating a capability for the transmission of encoded voice statistically multiplexed with data in an experimental network. The focal point for most aspects of the work is network delays - as they apply to voice transmission.

The analytical work first summarizes internal subnet delays, discusses various aspects of parallel servicing, and then makes a beginning in the analysis of peripheral (or "admission") delays [56,57]. For the case of single-hop synchronous traffic (not involving flow control), the solution is provided for the pdf of the number of "bulk units" in an $M/M/\infty$ and in an $M/M/\infty/M$ system with "non-homogeneous" job population, and the results are representative of entrance node operation under light load conditions.

A first step is made in the study of multi-hop synchronous traffic (involving flow control) by presenting an approximate model for such traffic. The model accounts approximately for the various components of admission delays associated with requesting, locating and delivering "reservations" to an entrance node, as well as the queueing delay associated with finding sufficient free servers on the first (entrance) link.

In terms of the experimental work, a capability for statistically multiplexing data and encoded voice in an experimental network is described [47,48]. This capability consists of a software implementation of an external link protocol for voice users, and the implementation of an internal protocol for synchronous voice and data traffic.

A preliminary subjective evaluation of encoded voice transmission subjected to emulated "admission" delays, suggests that fairly large random delays are imperceptible provided they occur predominantly between phrases and sentences rather than (unnaturally) between syllables and words.

The feasibility of 12-14 GHz satellite links for use with "extension" nodes is discussed based on the transmission of data and encoded voice over an experimental link, between Carleton University's Wired City Laboratory and NASA AMES Research Center in California. A detailed link error performance study is presented in terms of packet error free interval lengths (EFIL's) as well as the number of bits in error and bit EFIL's per packet [48,49]. The results for packet EFIL's suggested a simple model for satellite transmission with terrestrial retransmission for error control purposes.

The significance of the work described in this thesis, is that it represents an important step in the development of a transparent and intelligent public network capable of transmitting encoded voice (at 9.6 kb/s and lower rates) and data. Such a network can serve as an integrated system for data and voice, and provide significant cost benefits to its users through savings in bandwidth, by statistically multiplexing data with encoded voice [85,87].

ACKNOWLEDGEMENTS

I would like to express my appreciation to Professor M.E. Ulug for his constant guidance and support throughout this work, and for providing constructive suggestions and criticisms as part of his review of the thesis. I also thank Professor B. Pagurek for taking on the role of supervisor during the final phase of the program, and Dr. P.K. Verma for being my external examiner. Acknowledgement is due to D. Weir who developed the original software for the experimental two-node TI-NET. I would also like to thank other team members, namely D. Guy, J. Lyrette and in particular N. Janakiraman for useful discussions relating to TI-NET flow control and admission delays. Gratitude is also extended to N. Janakiraman and to my wife Lise for their efforts in perceiving the effects of variable delays on voice transmission.

In the earliest phase of this project, hardware and software support in the area of speech processing, was provided by Professor L.R. Morris to whom I am grateful. Somewhat later, support was also received from Carleton's Wired City Laboratory in the form of assistance from personnel, and the use of facilities for satellite experiments. In this regard, acknowledgement is extended to the Department of Communications, the Wired City and NASA AMES Research Center for making the satellite experiments possible. In particular I thank the personnel of the Wired City and NASA AMES for their assistance throughout the experiments. Gratitude is also extended to Professor J.K. Cavers who made available his fast Z-transform program with which certain analytical results in the thesis were verified.

I am indebted to Mr. D.P. Fulghum of E. Systems Inc., Dallas, Texas for his support in making possible the loan of two VADAC II speech processors for nearly the entire duration of this project, and two VADAC V processors for a shorter period, and who on many occasions provided helpful information regarding these units.

I also wish to extend my thanks to CNCP Telecommunications of Montreal, who on one occasion made possible the loan of a high speed modem for satellite experiments, and on another occasion, arranged modem rental on very short notice.

The financial assistance provided by the National Research Council of Canada has made this work possible, and is gratefully acknowledged.

To my wife Lise I wish to express my sincere gratitude for standing by me and encouraging me during this difficult period.

Finally, I am grateful for the conscientious approach taken by Mrs. Ruth Thomas, who so accurately and speedily typed this thesis.

TABLE OF CONTENTS

	PAGE
CHAPTER 1 - INTRODUCTION	1
1.1 General	1
1.2 Nature of the Problem: Effects of Delay on Voice Transmission	2
1.2.1 Fixed Delays in Switched Telecommunication Networks	3
1.2.1.1 Subjective Evaluation of Voice Transmission with Fixed Delays ...	5
1.2.1.2 Effect of Fixed Delays on Conversational Behavior	6
1.2.2 Variable Delays in Conventional Packet Switched Networks	8
1.2.2.1 Introduction	8
1.2.2.2 Previous Work in Packetized Voice Transmission	10
1.2.2.3 Subjective Effects of Anomalies on Packetized Speech	13
1.3 Motivation for Research in Packetized Voice Transmission	15
1.4 Proposed Solution - TI-NET	18
1.4.1 TI-NET Description	18
1.4.2 TI-NET Project History and Organization ...	18
1.5 Thesis Organization	19
1.5.1 Objective and Significance	19
1.5.2 Scope and Focus	20
1.5.3 Contributions	23
1.5.3.1 Analytical	23
1.5.3.2 Experimental	24
1.5.4 Thesis Structure	25
CHAPTER 2 - RELEVANT PAST AND PRESENT RESEARCH	27
2.1 Speech Analysis/Synthesis Techniques	27
2.1.1 General	27
2.1.2 Frequency Domain Techniques	30
2.1.2.1 The Channel Vocoder	30
2.1.2.2 The Formant Vocoder	31
2.1.2.3 The Homomorphic Vocoder	31
2.1.2.4 Voice Excited Vocoder	32

TABLE OF CONTENTS (Continued)

	PAGE
2.1.3 Time Domain Techniques	33
2.1.3.1 Linear Predictive Coder (LPC)	33
2.1.3.2 Adaptive Predictive Coder (APC) ..	34
2.1.3.3 LPC and APC Vocoders with Quantization and Encoding Schemes	35
2.1.4 Vocoder Evaluation	37
2.2 Approaches to Integrated Data and Voice Trans- mission and Their Implications with Respect to Delay	42
2.2.1 Circuit Switching	42
2.2.2 Hybrid Switching	44
2.2.3 Conventional Packet Switching	46
2.2.4 Transparent Intelligent Network Approach ..	46
2.2.4.1 General	46
2.2.4.2 SPIDER Network	49
2.2.4.3 TRAN M3200 System	50
2.2.4.4 CODEX 6000 System	52
2.3 Delay Related Issues in Packetized Voice Trans- mission	53
2.3.1 Overhead Versus Packet Formation Delay	54
2.3.2 Topology and Connectivity	55
2.3.3 Routing, Packet Sequencing and Message Reassembly	55
2.3.4 Network Protocols	57
2.3.4.1 Error Control	57
2.3.4.2 Flow Control	58
CHAPTER 3 - OVERVIEW OF TI-NET SYSTEM DESIGN	63
3.1 Subnet Access and Protocol Transparency	63
3.2 Transparency in Time and Packet Structure	64
3.3 Terrestrial Subnet Architecture	70
3.3.1 Topology and Transportation Mechanisms	70
3.3.2 Error Control	73
3.3.3 Flow Control	76
3.4 Satellite Subnet Architecture	78
3.5 Satellite Extension Nodes	81
3.6 Summary of TI-NET Characteristics	84

TABLE OF CONTENTS (Continued)

	PAGE
CHAPTER 4 - AN ANALYSIS OF TI-NET DELAYS	87
4.1 Summary of TI-NET Delays	87
4.2 Aspects of Parallel Servicing	92
4.2.1 Parallel Servicing Vs a Single Server System	92
4.2.2 Physical Description of TI-NET Entrance and Exit Node Operation	93
4.3 Admission Delays for Single-Hop Synchronous Traffic	101
4.3.1 Non-Homogeneous Job Population in an M/M/ ∞ System	101
4.3.1.1 Motivation and Assumptions	101
4.3.1.2 System Representation Using Parallel Service Facilities	103
4.3.1.3 System Representation Using a Single (Combined) Service Facility	121
4.3.2 Non-Homogeneous Job Population in an M/M/ ∞ /M System	138
4.4 Admission Delays for Multi-Hop Synchronous Traffic	149
4.4.1 An Approximate Model for Admission Delays ..	149
4.4.2 A Simplified Multiple Resource Model for the Time to Locate Reservations in TI-NET ..	153
4.4.3 Formulation of the Admission Delay Model ..	159
CHAPTER 5 - EXPERIMENTAL WORK ON ENCODED VOICE TRANSMISSION IN TI-NET	174
5.1 Background and Preliminaries	174
5.1.1 Voice Terminals	174
5.1.1.1 An LPC Voice Response System	174
5.1.1.2 The VADAC Channel Vocoder	176
5.1.2 External Link Considerations	181
5.1.2.1 Subnet Access	181
5.1.2.2 Protocols	181
5.2 External Link Protocol Implementation	186
5.3 Packetizing and Transmitting Encoded Voice within the Experimental Terrestrial Network	191
5.3.1 Description of the Two-Node Terrestrial Network	191
5.3.2 Internal Transmission Protocol for Synchronous Voice and Data Traffic	194
5.3.3 Experimental Network Delays	199

TABLE OF CONTENTS (Continued)

	PAGE
5.4 Emulation of Admission Delays	203
5.4.1 Admission Delay Model	203
5.4.2 Emulation Implementation	208
5.4.3 Experiment Description and Results	213
5.5 Transmission on an Experimental Satellite Link	228
5.5.1 Preliminaries	228
5.5.2 Experimental Satellite Link Description	230
5.5.3 Significance of the Experiments	235
5.5.4 Description of Experiments	240
5.5.5 Experimental Results	242
5.5.5.1 Packet Error Rates, Uplink Powers and Atmospheric Conditions	242
5.5.5.2 Packet Error Free Interval Lengths	245
5.5.5.3 Number of Bit Errors and Lengths of Bit Error Free Intervals Within Packets	252
5.5.5.4 Satellite Propagation Delays	257
5.5.6 A Model for Estimating Error Performance Improvement for a Satellite Link with Terrestrial Retransmission	259
CHAPTER 6 - SUMMARY, CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK	265
6.1 Summary and Conclusions	265
6.1.1 General	265
6.1.2 Analytical Work	268
6.1.3 Experimental Work	272
6.2 Suggestions for Further Work	279
6.2.1 Simulation and Analysis	279
6.2.2 Experimental	283
REFERENCES	285
APPENDIX A	296
A.1 Summary of Results for an M/M/m System	296
A.2 Algorithm for Determining the Ways of Having k Bulk Units in the System	298
A.3 State and Substate Representation for the Single Combined Service Facility	302
A.4 Standard Deviation for the Number of Messages in an M/M/m/M System	310
A.5 Summary of Results for an M/M/m/M System	312
A.6 Convolution of Four Exponential pdf's with Different Means	315

TABLE OF CONTENTS (Continued)

	PAGE
APPENDIX B	317
B.1 Software Documentation	317
B.2 Hardware Documentation	345
B.3 Concatenation of Messages in Node Buffers	350
B.4 Emulation of Admission Delays: Model and Speech Source	353
B.5 Emulation of Admission Delays: Prepared Tape and Results	362
B.6 Satellite Link Emulation, and Supplementary Results of Satellite Link Error Measurements .	371

LIST OF FIGURES

FIGURE		PAGE
3.1	TI-NET Packet Structure	66
3.2	Terrestrial Subnet Topology: Fully Connected Starnets	71
3.3	Integrated Terrestrial and Satellite Subnet Architectures: Normal Operation	80
3.4	Satellite Extension Node	82
4.1	Example of 3 Hop Traffic in TI-NET	88
4.2	TI-NET Delays Internal to the Subnet	89
4.3	Normalized Mean Time Spent Waiting in the Queue for M/M/m System	94
4.4	Normalized Standard Deviation of Time Spent Waiting in the Queue for M/M/m System	95
4.5	TI-NET Entrance and Exit Node Operation	96
4.6	Example of a TI-NET Multi-User Packet - MP Containing Minipackets - mp's	99
4.7	Non-Homogeneous Jobs in an M/M/ ∞ System: Representation Using 4 Parallel Service Facilities	104
4.8	Rate Transition Diagram for the i th Facility in an M/M/ ∞ System with Non-Homogeneous Population	105
4.9	Illustration for the Convolution $P_1(k) * P_2(k)$	111
4.10(a)	pdf for an M/M/ ∞ System with Non-Homogeneous Population	114
4.10(b)	pdf for an M/M/ ∞ System with Non-Homogeneous Population	115
4.10(c)	pdf for an M/M/ ∞ System with Non-Homogeneous Population	116
4.11	Non-Homogeneous Jobs in an M/M/ ∞ System: Representation Using a Single Service Facility	122
4.12	Rate Transition Diagram for Single Service Facility	129
4.13	Localized Rate Transition Diagram for Type i Jobs	134
4.14	Average Proportional State Dependent Service Rates vs Number of Bulk Units in the System ...	135
4.15	Average Proportional State Dependent Service Rates Evaluated at State $k=r$ vs Mean Number of Bulk Units in the System	137

LIST OF FIGURES (Continued)

FIGURE		PAGE
4.16	Overall Average Proportional Service Rates vs Mean Number of Bulk Units in the System	139
4.17	Non-Homogeneous Jobs in an M/M/ ∞ /M System: Representation Using 4 Parallel Service Facilities	141
4.18	Rate Transition Diagram for the i th Facility in an M/M/ ∞ /M System with Non-Homogeneous Population	143
4.19	pdf's for an M/M/ ∞ /M System with Non-Homogeneous Population	146
4.20	Example of 3 Hop Traffic in TI-NET	151
4.21	Model for Admission Delays in a TI-NET Entrance Node	151
4.22	Simplified Multiple Resource Model for the Time to Locate Reservations in TI-NET	154
4.23	Normalized Mean Time Spent in the Queue for M/M/m/M System	157
4.24	(a) Asymptotic Mean Times to Locate Reservations (b) pdf for the Number of Active Terminals	163
4.25	Service Time pdf's for the TI-NET Admission Delay Model (in Figure 4.21)	168
4.26	Service Time pdf for the TI-NET Admission Delay Model	169
4.27	Service Time pdf for the TI-NET Admission Delay Model	171
5.1	Functional Diagram of Existing Voice Response System	175
5.2	Functional Diagram of VADAC Channel Vocoder ...	178
5.3	Format of 2.4 kb/s Synchronous Output of VADAC Speech Processor	180
5.4	External Link Protocol for Encoded Voice Transmission	183
5.5	Subnet Access for Encoded Voice Transmission ..	185
5.6	Processing of Vocoder Frames before Packetizing	189
5.7	Processing of Frames before Delivery to Vocoder	192
5.8	Block Diagram of the Experimental TI-NET	193

LIST OF FIGURES (Continued)

FIGURE		PAGE
5.9	Formation of mp's in a TI-NET Entrance Node	195
5.10	Illustration of the Coding for Conveying Reservation Information in (not last) Voice and Data mp's and for Padding Partially Filled mp's	198
5.11	Experimental TI-NET Internal Delays	200
5.12	Simplified Model Used for the Emulation of Admission Delays	204
5.13	Example of pdf's for the Emulation of Delays ..	207
5.14(a)	Effect of Emulated Delays: $D_l < \tau$	210
5.14(b)	Effect of Emulated Delays: $D_l > \tau$	210
5.15	Experimental System for Evaluating the Effects of Admission Delays	214
5.16	Effect of Hangover	216
5.17	Mean Time Messages Spent in the Input Buffer Versus Emulated Delay	218
5.18	Perceptible Effect of Admission Delay	223
5.19	Potential Effect of the Central Node Reservation Scheme	227
5.20	Satellite Extension Node Experiment	231
5.21	Details of Carleton Satellite Terminal Equipment	232
5.22	Illustration of NAK Plus Retransmission Delays, and Associated Output Buffering Delays	237
5.23	NAK + Retransmission Delays in the Experimental System	241
5.24	MP Error Rate vs NASA AMES Uplink Power for Various Atmospheric Conditions	243
5.25	Density of MP Error Free Interval Lengths (High MP Error Rate: 1.4×10^{-2})	246
5.26	Density of MP Error Free Interval Lengths (Lower MP Error Rate: 2.0×10^{-3})	248
5.27	MP Error Free Interval Lengths vs MP Error Rate	250
5.28	Performance of Error Control by Terrestrial Retransmission for Satellite Extension Nodes, Based on MP Error Free Interval Data	251

LIST OF FIGURES (Continued)

FIGURE		PAGE
5.29	Density of Number of Bit Errors within MP's (Header + Data Field + 128 Bits)	253
5.30	Density of Bit Error Free Interval Lengths within MP's (Header + Data Field + 128 Bits) ..	254
5.31	Output Buffering Delay vs Terrestrial Link Speed for a 56 kb/s Satellite Link with Error Control by Terrestrial Retransmission	262
5.32	Performance of Error Control by Retransmission for a 56 kb/s Satellite Link Based on Exponential MP EFIL's	264
A.2.1	Breakdown of Number of Ways of Having k Bulk Units in the System in Figure 4.7	301
A.3.1	Illustration of Ways of Having k=6 in Terms of Cartesian Products	303
A.3.2	Rate Transition Diagram for [k=6, B _j (6)]	305
A.3.3	Rate Transition Diagram for Figure A.3.2 with Substate Dependencies Absorbed	307
A.5.1	Normalized Mean Time Spent in the Queue for M/M/m/M System	314
B.1.1	Part of Routine 3) - AMUX: mp Header (Byte Count) Processing	321
B.1.2	Part of Routine 4) - ADEMUX: mp Header (Byte Count) Processing	322
B.1.3	Routine 5) - SMUX: (1st of 3) Message Status and mp Header Bit Processing	323
B.1.4	Routine 5) - SMUX: (2nd of 3) Multiplexing and mp Trailer Processing	324
B.1.5	Routine 5) - SMUX: (last of 3) Message Management and mp Padding Calculations	325
B.1.6	Routine 6) - SDEMUX: (1st of 2) mp Header and Trailer Processing	326
B.1.7	Routine 6) - SDEMUX: (last of 2) Demultiplexing and Message Management	327
B.1.8	Part of Routine 7) - DCB45T: Receive from VADAC (1st of 3); Framing, Frame Checking and Reframing	328
B.1.9	Part of Routine 7) - DCB45T: Receive from VADAC (2nd of 3) Frame Alignment, Initial Sequence Transmission, End of Message Management and Signaling	329

LIST OF FIGURES (Continued)

FIGURE		PAGE
B.1.10	Part of Routine 7) - DCB45T: Receive from VADAC (last of 3); Active Frame Detection, Hangover Implementation, Message Buffering	330
B.1.11	Part of Routine 7) - DCB45T: Transmit to VADAC (1st of 3); Message Status, Active Frame Sequence Identification, Meshing Check, Idle to Active Frame Transfer	331
B.1.12	Part of Routine 7) - DCB45T: Transmit to VADAC (2nd of 3) End of Message Signalling Check, End of Message Frame Sequence Identification, Message Alignment before Output	332
B.1.13	Part of Routine 7) - DCB45T: Transmit to VADAC (last of 3) Active to Idle Frame Transfer, Frame Checking, Continuous Frame Output	333
B.1.14	Routine 10) - ADMCAL: Calling Routine and Computation of Emulated Delays	334
B.1.15	Part of Routine 11) - DEMOA: Setup and Logging of Emulated Delays	335
B.1.16	Part of Routine 12) - SMUXA: Implementation of Emulated Delays	336
B.1.17	Part of Routine 13) - SDMUXA: "Central Node Reservation" Supply	337
B.1.18	Part of Routine 16) - DEMOS: (1st of 2) Transmission Error Analysis and Logging of Error Statistics	338
B.1.19	Part of Routine 16) - DEMOS: (last of 2) Synchronization of Synchronous MP Processing and Reception (Logically Identical for Asynchronous MP's)	339
B.1.20	Part of Routine 17) - DCB45S (1st of 3) Clock Service Routine: Task Timing, Propagation and Message Delay Accumulators, MP Error Free Interval Timing	340
B.1.21	Part of Routine 17) - DCB45S (2nd of 3) MP Transmit Service Routine: MP Start and End Timing, Initiation of Propagation Delay Measurements	341
B.1.22	Part of Routine 17) - DCB45S (last of 3) MP Receive Service Routine: MP Start and End Timing, Termination of Propagation Delay Measurements, Error Detection, Buffering for Synchronization of MP Processing and Reception	342

LIST OF FIGURES (Continued)

FIGURE		PAGE
B.1.23	Part of Routine 20) - DCB45N (1st of 2) NAK Transmit Routine (Retransmit Send Routine in Routine 22) - DCB40R is Nearly Identical)	343
B.1.24	Part of Routine 20) - DCB45N (last of 2) Retransmission Receive Routine (NAK Receive Routine in Routine 22) - DCB40R is Nearly Identical)	344
B.2.1	VADAC Interface to PDP-11	346
B.2.2	Modem Interface and Clock Distribution	347
B.2.3	Audio Frequency Adapters	348
B.2.4	Signal/Noise Combiner	349
B.3.1	Preventing Concatenation of Two Messages in the Entrance Node Buffer	351
B.3.2	Preventing Concatenation of Two Messages in the Exit Node Buffer	352
B.4.1	Frequency of Active Periods having Various Durations for 2½ Minute Taped Session	359
B.4.2	Frequency of Idle Periods having Various Durations for 2½ Minute Taped Session	360
B.4.3	Effect of Hangover on Mean Active and Idle Period Duration	361
B.5.1	Mean Input Buffer Contents vs Emulated Delay ...	368
B.5.2	Peak Input Buffer Contents vs Emulated Delay ...	369
B.5.3	Other Effects of the Central Node Reservation Scheme	370
B.6.1	Emulated Satellite Link	373
B.6.2	Density of MP Error Free Interval Lengths; MP Error Rate: 2.1×10^{-2}	374
B.6.3	Density of MP Error Free Interval Lengths; MP Error Rate: 1.4×10^{-2}	375
B.6.4	Density of MP Error Free Interval Lengths; MP Error Rate: 8.8×10^{-3}	376
B.6.5	Density of MP Error Free Interval Lengths; MP Error Rate: 3.3×10^{-3}	377
B.6.6	Density of MP Error Free Interval Lengths; MP Error Rate: 1.2×10^{-3}	378
B.6.7	Density of MP Error Free Interval Lengths; MP Error Rate: 9.8×10^{-4}	379

LIST OF FIGURES (Continued).

FIGURE		PAGE
B.6.8	Density of Number of Bit Errors within MPs (Header + Data Field + 128 Bits)	380
B.6.9	Density of Number of Bit Errors within MPs (Header + Data Field + 128 Bits)	381
B.6.10	Density of Number of Bit Errors within MPs (Header + Data Field + 128 Bits)	382
B.6.11	Density of Bit Error Free Interval Lengths within MPs (Header + Data Field + 128 Bits)	383
B.6.12	Density of Bit Error Free Interval Lengths within MPs (Header + Data Field + 128 Bits)	384
B.6.13	Density of Bit Error Free Interval Lengths within MPs (Header + Data Field + 128 Bits)	385
B.6.14	Coded and Uncoded High Speed Modem Performance .	386

LIST OF TABLES

TABLE		PAGE
4.1	Parameters Associated with Figure 4.19	147
4.2	Parameters for Examples in Figures 4.25 and 4.26 .	165
4.3	Parameters for Examples in Figures 4.25 and 4.26 .	166
4.4	Parameters for Example in Figure 4.27	172
5.1	Number of Bit Errors and Length of Bit EFI's within MP's	255
5.2	Transmission Delay on 9.6 kb/s Satellite Link	258
A.2.1	Algorithm Output for Even Number of Bulk Units ...	299
A.2.2	Algorithm Output for Odd Number of Bulk Units	300
B.5.1	Emulated Delay Parameters	364
B.5.2	Emulated Delay Parameters	365
B.5.3	Emulation Delay Results	366
B.5.4	Emulation Delay Results	367

LIST OF FREQUENTLY USED SYMBOLS

α_1	Probability of operating below saturation
α_2	Probability of operating above saturation
$B_j(k)$	jth substate: jth way of having k bulk units in the system
C	Total output service capacity (e.g. char/sec)
cv	Coefficient of variation
γ_i	Proportional service rate for type i messages
δ_i	Load balance for type i messages
E_b/N_0	Energy per bit/noise power density
g_i	Traffic balance for type i messages
i	Index over the set $s = \{1,2,4,8\}$ indicating message type
k	Total number of bulk units in service
k_i	Number of type i messages in service
$\bar{l}$	Mean message length for all message types (char/msg)
L_j	Truncation value of emulated delay in path j
λ	Total arrival rate (msg's/sec)
m	Number of servers
M	Number of terminals
M^*	Number of terminals at saturation
N_s	Mean number of messages in service
N_q	Mean number of messages in the queue
N_t	Mean number of messages in the system
P_k	Probability of k bulk units in the system
P_{k_i}	Probability of k_i messages in facility i
P_M	pdf for number of active terminals
P_o	Probability of system being empty
P_q	Probability of queueing
R	Server capacity (e.g. char/sec)
r	Mean number of bulk units in the system
r_i	Mean number of type i messages in the system
ρ	utilization factor
S_i	Service rate for type i messages (msg's/sec)
σ	Standard deviation

T_s	Mean time in service
T_q	Mean time in the queue
T_t	Mean time in the system
T_1	Mean time to request reservations
T_{21}	Mean time to locate reservations when operating below "saturation"
T_3	Mean time to deliver reservations
T_4	Mean time to find a free server on the first hop for multi-hop traffic
T_{22}	Mean time to locate reservations when operating above "saturation"
T_{qL}	Mean time spent in the queue (lower asymptote)
T_{qU}	Mean time spent in the queue (upper asymptote)
μ	Mean service rate per server (msg's/sec)

LIST OF ABBREVIATIONS

ASYNCR	Asynchronous
ATDM	Asynchronous time division multiplexing
BPSK	Binary phase shift keying
B/S	Balanced to single-ended
CDF	Cumulative distribution function
EFI	Error free interval
EFIL	Error free interval length
FCFS	First come - first served
F.U.	Fractional utilization
HDLR	High level data link control
IBE	Instantaneous bandwidth expansion
M/M/1	Poisson arrivals, exponential service, single server
M/M/ ∞	Poisson arrivals, exponential service, infinite servers
M/M/m	Poisson arrivals, exponential service, m servers
M/M/ ∞ /M	Poisson arrivals, exponential service, infinite servers, finite (M) population (closed)
M/M/m/M	Poisson arrivals, exponential service, m servers, finite (M) population (closed)
MODEM	Modulator-demodulator
MP	Multi-user packet
mp	Mini packet
MSG	Message
NAK	Negative acknowledgement
pdf	Probability density function
QPSK	Quadrature phase shift keying
RX	Receive or receiver
S/B	Single-ended to balanced
SFCFS	Selective first come - first served
STDM	Synchronous time division multiplexing
SYNC	Synchronous
TI-NET	Transparent intelligent network
TX	Transmit or transmitter
U.F.	Utilization factor
VADAC	Voice analyzer data converter

LIST OF CO-AUTHORED PUBLICATIONS

1. Ulug, M.E., Weir, D.F., Morris, L.R. and Gruber, J.G., "An experimental transparent intelligent network", ICCA Aug. 1976, Toronto, Canada, pp.323-329.
2. Ulug, M.E., Gruber, J.G., "Statistical multiplexing of data and encoded voice in a transparent intelligent network", Fifth Data Com. Symp., Snowbird, Utah, Sept. 1977, pp.6-14 to 6-20.
3. Ulug, M.E., Gruber, J.G., "An experimental satellite link for a transparent intelligent network", NTC, Dec. 1977, Los Angeles, Calif., pp.48:4-1 to 48:4-8.
4. Ulug, M.E., Gruber, J.G. and Janakiraman, N., "Admission delays in a transparent intelligent network", Published in ICC June 1978, Toronto Canada, pp. 27.6.1 - 27.6.11 .
5. Ulug, M.E., Janakiraman, N., Gruber, J.G., "Simulation of admission delays in a transparent intelligent network", Summer Computer Simulation Conference, July 1978, Newport Beach, Calif.
6. Ulug, M.E., Janakiraman, N., Gruber, J.G., "Instantaneous bandwidth expansion in a transparent intelligent network", To be submitted to IEEE Trans. on Com.

CHAPTER:1

INTRODUCTION

1.1 General

This thesis addresses itself to certain aspects of the general problem of transmitting voice with data in a computer communications network. It has been well established that the principal problem associated with voice transmission in a conventional packet switched network such as ARPA [1], is large end-to-end network delays - particularly variable delays [2,3,21]. More specifically, the thesis investigates "encoded" voice transmission in a network which has potential advantages for voice transmission, since its objective is to provide low fixed and variable delays. This network is called TI-NET, which is an abbreviation of Transparent Intelligent NETwork [4]. In simple terms, a "transparent" network strives to imitate a leased line by minimizing end-to-end delays (transparency in time), and by simplifying customer interfacing (transparency in protocol); an "intelligent" network is one which allows itself to be efficiently shared by both voice and data traffic, and provides such network functions as error control, flow control and routing.

The term "encoded" voice referred to in the thesis title, implies that the voice signals in question have not only been previously converted from analogue to digital form (i.e. digitized), but in addition have been processed

by one or another of various analysis/synthesis techniques (discussed later in Chapter 2 [5]), for the purpose of removing redundancies and thereby reducing their transmission rates to 9.6 kb/s or less. In this sense voice signals are made compatible with data messages. However, it will be evident that in general such encoded voice signals cannot be treated simply as data messages.

1.2 Nature of the Problem:
Effects of Delay on Voice Transmission

From the point of view of the response time encountered by data network users, total end-to-end network delay is a major concern for real-time-data applications such as inquiry/response and time-shared computing. Voice traffic is a relatively recent real-time category being considered for data networks [2,37,38]. However, the implications of data network delays for voice transmission are more subtle than for data traffic, since in addition to the response time aspect, they may affect users in a subjectively noticeable and perhaps objectionable way, as well as modify their conversational behavior patterns.

In order to review these effects, delays may be divided into two categories: fixed delays, as encountered in existing switched telecommunications networks, and variable delays, such as those encountered in conventional packet switched networks.

1.2.1 Fixed Delays in Switched Telecommunication Networks

The problem of fixed delays has a history which goes back to the 1920's and 1930's [6] and is very much related to another voice transmission problem - namely echo (echo arises mainly because of the need to interface two-wire subscriber lines to four-wire long distance facilities). Although delayed echo has been, and to some extent is still a problem, it is not discussed here; this is consistent with the view that initially at least, voice transmission in computer networks will likely be of a somewhat specialized nature, and network access will be via four-wire facilities. Also, it is expected that recent developments in echo-cancellation techniques may soon lead to an effective and economical solution to the echo problem [7].

It is mentioned here however, that related to the delayed echo problem, it has been found that interruptions during conversations are a fairly common occurrence; some tests indicate that while one party is talking there is about a twenty percent probability that the other will interrupt [8] (these occurrences are called "double talking"). Of interest here is the fact that both the duration and the number of incidents of double talking have been found to increase with longer transmission delays.

The delayed echo problem as well as other factors, stimulated the development and use of higher speed carrier systems, which since the 1930's have had propagation speeds

at least two-thirds that of light. This reduced the transmission delay to the point where echo suppressors were capable of minimizing most of the serious effects of echoes. Thus there was little interest and activity with regard to delay until the early 1960's, when it was realized that the communication satellite would again resurrect this problem. Most modern communication satellites are in "geosynchronous" orbit and are held stationary by telemetry means; therefore round trip delays are about 540 msec. There are also slow delay variations of typically much less than a few msec, which are important from the point of view of synchronizing digital transmission systems, but are otherwise of no concern here.

Since the coming of the satellite, considerable laboratory and field effort has gone into subjective evaluation of voice communication with long transmission delays [6,8-10]. The problems investigated in this regard have been categorized as those due to pure delay, and those related to echo (which again are not considered here).

Studies of pure delay have involved round trip delays of:

- 50 ms - long terrestrial circuits
- 100 ms - submarine cable
- 200 ms - combinations of terrestrial and submarine
- ≥ 600 ms - stationary satellite
 - combinations of terrestrial and satellite
 - multihop satellite

In addition, the pure delay condition has been investigated from two points of view:

(i). Subjective evaluation: determining the extent to which users actually notice the delay and their reaction to it, such as whether they are inclined to complain about a circuit or to reject it.

(ii) Conversational behavior: determining the extent to which conversational behavior patterns are modified even though the user may not be aware that this is happening.

1.2.1.1 Subjective Evaluation of Voice Transmission with Fixed Delays

Since the essence of a telephone conversation is the rapid exchange of ideas, it was originally anticipated that the inclusion of long round trip delays in the circuit would adversely slow down the communication process and lengthen call durations. For round trip delays of 600 and 1200 msec, this has largely proved to be untrue (although for longer delays conversational "sluggishness" becomes noticeable). In fact call durations have been found to stay about the same or in some cases to shorten; this is due to changes in the users' conversation habits, as well as perhaps to some degree, to subtle factors of annoyance and discomfort resulting from the greater incidence and duration of interruptions caused by double talking. However, although some users in subjective tests with normal conversation, complained about the confusion due to being interrupted, in general

they were basically not aware of the delay itself, and were not inclined to reject the circuit because of it; neither did they appear to become overly sensitive to the delay after repeated exposure to it, or to become more critical of the circuit with time [11].

On the other hand in more structured tests, participants were able to detect the presence of an added 600 msec round trip delay after about three minutes or so [12]. Still other tests revealed that the amount of time required to perform certain tasks in a teleconferencing environment increased by nearly thirty percent when delays of 300 and 600 msec were introduced, indicating a drop in performance for tasks other than naturally occurring conversations [13].

1.2.1.2 Effect of Fixed Delays on Conversational Behavior

The effects of delays on the dynamics of conversational behavior are subtle and usually unnoticed by the conversants, even though they may actually be experiencing difficulty in talking over circuits with delay. It is possible to obtain qualitative measures of the effects of delay by modelling the conversation process. A simple model for two speakers in a conversation consists of the four states defined below [8]:

	<u>State</u>	<u>State Description</u>
1	00	both speakers silent (mutual silence)
2	01	1st silent, 2nd talking
3	10	1st talking, 2nd silent
4	11	both talking (double talking)

Transitions are assumed to be possible between all states except for those requiring both speakers to change their status at the same time (i.e. transitions 00 ↔ 11 and 01 ↔ 10 are assumed to occur with negligible probability). The double talking state (4) is of most interest here. A transition from state (2) to (4) is an example of "voluntary" double talking from the point of view of the first speaker (he interrupts). However, a transition from state (3) to (4) results in "involuntary" double talking as far as the first speaker is concerned (he is interrupted). The point is that with no delay between the speakers, any instance of double talking is always voluntary to one speaker and involuntary to the other, and this is the normal situation when speakers occasionally interrupt one another during a conversation.

However, when there is significant delay between the speakers, (and assuming an initial condition of mutual silence) the first speaker may start talking and the second, not yet realizing this because of the delay, may very well do the same. Some time later therefore, double talking takes place which is involuntary on the part of both speakers. It is this type of involuntary double talking which is closely related to confusion in the conversation. This is particularly true when it results in the original speaker stopping and the second one continuing; test results in conjunction with the above model show that the incidence of this type

of double talking increased from two to four to six per minute, for zero, 600 and 1200 ms round trip delays respectively.

Another similar but more sophisticated six state model has also been used to demonstrate that although (inexperienced) test subjects noticed nothing unusual about circuits with 600 and 1200 ms round trip delays in them, the delay caused a statistically significant increase in the frequency of confusions and in the amount of both double talking and mutual silence [12]; the latter two parameters increased by about 40 and 9 percent respectively. There is also some evidence to suggest that the effect of the delay is to cause subjects to make some adjustments in their speaking behavior by waiting longer for responses, but also talking longer when interrupted. Related to these findings is the psychological factor that when the reinforcing feedback of an expected answer is delayed, the speaker may feel the other party is not paying proper attention, and may tend to become irritated; similarly when the other party tries to interrupt the speaker, that party may become annoyed since the speaker seems to be ignoring his attempt to interrupt.

1.2.2 Variable Delays in Conventional Packet Switched Networks

1.2.2.1 Introduction

In the late 1960's the new technology called packet switching appeared, and since 1970 has been exemplified mainly

by the ARPA network in the U.S.A. [14], as well as more recently by many others such as Cyclades in France [15], EIN in Europe [16], Telenet in the U.S.A. [17] and Datapac in Canada [18]. These networks were designed primarily to provide an efficient, accurate and reliable means for geographically separate locations to share processing power and to communicate with one another using shared facilities, and thereby reduce both processing and communication costs. Originally at least, these networks were strongly oriented toward data applications; however, recently a great deal of attention is being given to other services, particularly voice transmission.

The first experiments with packetized voice transmission appear to be associated with ARPA, and were done as a result of the concern to minimize network delays - both fixed and variable - between speaker and listener. Variable delays in a network such as ARPA arise because of the desire to share or utilize the network in an efficient manner by operating it at a high throughput (number of messages transmitted through the network per unit time). Thus the requirements for low delay and high throughput are conflicting and involve a trade-off; in this regard, it is usually desirable to either maximize the average throughput subject to an average delay constraint, or alternatively, for a given average throughput, to minimize the average delay.

For a low enough throughput, network delays will tend

to approach an approximately fixed minimum consisting of propagation delays as discussed previously, but in addition will include various others such as "packet formation", "packet servicing" and "store and forward" delays; these will be discussed in more detail in Chapter 2. Thus at a very minimum, the effects of fixed delays reviewed previously can be expected to be present in packet switched networks as well.

However, as the throughput increases, a variable component of delay arises partly as a result of packets queueing at network nodes, and it is this variable component which is known to have the most implications for voice transmission. The most basic and direct effect of variable delays, is to noticeably distort the time scale of voice messages received at the network exit node. In addition to queueing however, it is emphasized that another factor contributing to variable delays is inherent in the distributed ("k-connected") topology of most conventional packet switched networks, and to the fact that packets making up a message are transported over more than one route through the network. This point is discussed in more detail in Chapter 2.

1.2.2.2 Previous Work in Packetized Voice Transmission

Early experimental work at UCLA [19] consisted of the serial transmission of both fixed and variable length simulated digitized speech messages through ARPA, at both a constant rate and in a burst mode, for the purpose of measuring round

trip delays. Experiments were also conducted with various sending algorithms which attempted to adapt the packet size to the simulated speech source rate, and to the network dynamics, so as to minimize the delays.

Work reported at Stanford [20] included, among other things, the description of a method of formatting voice packets so that the fluctuation in packet arrival times can be removed. The scheme consists of including packet overhead bits which encode the relative times between successive speech packets at the originating node. At the destination node, this information (time stamp) can be used to restore the proper time scale between the received messages. The resolution and dynamic range of the relative times considered were 10 msec and 10 sec respectively. To handle the longer intervals between messages using this scheme, fixed time references need to be periodically transmitted.

More indirect, but nevertheless very important effects of variable delays arise when attempts are made to remove time scale distortion before delivering speech messages to their destination. In this respect work reported more recently at M.I.T. [22-25] involved encoded voice (from a low bit rate - 3500 b/s - voice encoder) which was transmitted through ARPA between M.I.T. Lincoln Laboratory and the University of Southern California. Measurements of network transit times showed a dispersion of packet delays varying from a minimum of about 200 ms to over one second. The very long delays

were thought to be due to the time required to retransmit those packets damaged by transmission errors, as well as due to queue overflow problems (which are more likely to occur for higher throughput rates).

Therefore, (from this data) in order to include all packets in reassembled speech messages, and smooth distorted packet interarrival times, requires a fixed reassembly delay of over 1 sec. Evidently, this is intolerable and a compromise is preferable wherein a smaller delay is used at the expense of not including some packets in reassembled messages. The distribution of packet arrival times is such that, for example, delays of 0.35 sec and 0.55 sec correspond to including 95% and 99% of packets respectively. The effect of missing packets, is gaps called "glitches" in the reproduced speech. For 1000 bit packets, and the above data, the mean times between glitches when 95 and 99% of packets are included would be about 1/2 sec and 3 sec respectively. Since the mean time of busy and idle periods in speech is about 1.1 and 1.7 sec. respectively, the above glitch rates would have a definite effect on speech quality [43]. Hence packet reassembly (reconstitution) algorithms are required to reduce the adverse effects of the lost packets on speech quality.

An approach being considered at Bell Laboratories [44, 83] for dealing with some of the problems encountered in packetized voice transmission is referred to as "variable-

quality coding and time-interval modification". Variable-quality coding is suggested as a means of accommodating fluctuating network demands by coding at a higher bit rate (higher quality) for light network loads, and at a lower bit rate (lower quality) during heavy loads. Time-interval modification is being considered as a means of allowing latitude in reconstructing the silent intervals in reassembled speech messages at the receiver, so as to relieve reassembly buffer requirements.

1.2.2.3 Subjective Effects of Anomalies on Packetized Speech [26-29]

(a) Late and lost packets

Various anomalies result in packetized voice transmission when speech messages cannot be reassembled properly because of variable network delays. The most common anomaly is gaps caused by late or lost packets. The subjective effect of gaps is likely to be imperceptible if the gap durations are short (20 to 50 ms) and fairly infrequent (about one every 10 sec). However, for more frequent gaps (about one every second), the reassembled speech takes on a very rough quality, although intelligibility is not seriously reduced unless gaps are frequent enough to destroy one quarter or more of the packets. Longer gaps (of 100 ms or more) can annihilate complete syllables, words and even phrases, and are perceived as more than a quality degradation since depending on the gap location, the meaning of a sentence could be seriously changed.

(b) Reassembly buffer underflow

It is possible that during a speech message, network delay could momentarily exceed the message reassembly delay and cause a buffer underflow, since the receiver then runs out of delayed speech to output. Although no speech is lost, the subjective effect of an underflow may be greater than that of a lost packet, since it produces a timing shift or hesitation in the output speech, that will seem very unnatural if for example it occurs in the middle of a word. Such hesitations may also impart subtle changes in meaning, whereas no such change may have been intended by the speaker.

(c) Reassembly buffer overflow

In the event that momentary network delays decrease during a speech message, packets may be delivered to the reassembly buffer faster than speech can be output to the terminal, and cause a buffer overflow. Such anomalies cause both a timing shift and a loss of information, and are therefore more damaging than the anomalies discussed previously.

(d) Reassembly buffer underflow followed by overflow

A combination of buffer underflow followed by buffer overflow may occur in the event that the network builds up packets at some point of congestion and then suddenly delivers them in a burst. The underflow would occur during the congestion, and the overflow during the release of the accumulated packets. Such an anomaly results in two

successive timing shifts as well as lost information due to the overflow, and therefore has a greater subjective effect than either of the previous anomalies mentioned.

1.3 Motivation for Research in Packetized Voice Transmission

Recently, considerable attention has been given to the prospect of voice transmission in computer communication networks. One of the reasons is that the transmission of voice messages in conjunction with such networks is expected to become increasingly important. Although most existing computer networks carry data only, there are many applications involving encoded voice transmission both to and from computers [30-36, 67]. For example, various automatic information services (for credit verification, airline reservations, stock quotations and inventory reporting, etc.) may employ voice transmission from computers. On the other hand, encoded voice transmission to computers introduces the possibility of performing such financial transactions as electronic funds transfer (EFT), and the accessing of privileged information by means of speaker verification and encryption techniques.

Another important motivation for investigating encoded voice and data transmission is based on economic considerations, and the desire to plan for integrated communication systems which can handle both voice and data efficiently [2,38,84-86]. One of the difficulties with integrated voice and data transmission on a large scale (apart from the critical delay requirements for voice), is that the total

volume of traffic is presently dominated by voice. Therefore in an integrated system, voice traffic would naturally tend to be the main cause of congestion, and thereby significantly influence such data traffic requirements as throughput and response time. An effective way of dealing with this disparity is to improve the efficiency of present digital voice transmission, which typically takes place in a synchronous time division multiplexing (STDM) mode, and requires 56 or 64 kb/s of channel capacity per voice channel [39].

Substantial reductions in channel capacity (of the order of two or three to one) are possible, and are presently being pursued, using digital speech interpolation (DSI) techniques [40,41]. DSI involves statistically multiplexing many voice channels in such a way as to take advantage of the idle periods in conversations [42,43] by transmitting portions of other conversations in those periods. DSI may also employ variable quality coding to minimize "freezeout" of certain portions of conversations under overload conditions.

However, much greater reductions in channel capacity are possible when statistical multiplexing is combined with other methods of speech compression. For example, using a vocoder operating at 2.4 kb/s, and accounting for speech idle time, the average transmission rate is only about 900 b/s. Although vocoder speech quality, intelligibility and naturalness at 2.4 kb/s is at present considerably poorer than voice on normal telephone circuits, both the quality

and cost of vocoders can be expected to improve considerably; this is evident from the recent advances being made with sophisticated analysis/synthesis algorithms (some of which are discussed later in Chapter 2), as well as the advent of LSI digital technology and microprocessor techniques [45]. In addition, savings in system bandwidth make it possible to offer service on vocoded speech circuits at lower cost to the user.

One approach with regard to integrated networks is to consider a combination (hybrid) of circuit switching for voice and packet switching for data traffic. The principal disadvantage of this method is that it does not exploit the idle periods present in speech to improve channel utilization. On the other hand in packet networks, information is only transmitted when a terminal is busy. Hence statistical multiplexing is used to combine the output of many terminals with high ratios of peak to average traffic. Similarly, voice circuits are normally idle about 60 to 65% of call durations on the average (one way), which makes possible a significant saving in network bandwidth [43]. Hence, the bursty nature of speech lends itself to the packet communication concept, and supports the idea of mixed data and voice transmission [87]. However, as indicated in the previous section, conventional packet switching is disadvantageous for voice traffic because of the potentially large variable delays which it introduces, and to which

voice traffic is particularly susceptible. The various approaches to integrated voice and data transmission are reviewed in more detail in Chapter 2.

1.4 Proposed Solution - TI-NET

1.4.1 TI-NET Description

TI-NET is intended as a public computer communications network which offers a solution to the problems of transparency in time and protocol, and which is therefore expected to be capable of transporting both data and encoded voice at 9.6 kb/s and lower rates [4,46-48]. TI-NET is also intended as a hybrid system consisting of integrated terrestrial and satellite subnet architectures, as well as satellite extension nodes to enable both local and remote regions of high data concentration to access the integrated system using 12-14 GHz satellite links [48,49]. An overview of the TI-NET system design is given in Chapter 3.

1.4.2 TI-NET Project History and Organization

TI-NET was system designed by Professor M.E. Ulug, and this design was submitted for publication in May 1975 [4]. At this time a team consisting of the following graduate students was formed for the purposes of analyzing and developing the TI-NET system:

D. Weir - software development for two terrestrial
TI-NET nodes [50]

- D. Guy - simulation of certain aspects of TI-NET flow control
- J. Gruber - experimental investigation of encoded voice transmission in TI-NET, including satellite extension nodes
 - preliminary analysis of TI-NET peripheral (admission) delays
- J. Lyrette - mathematical analysis of certain TI-NET flow control and queueing problems
- N. Janakiraman - simulation and analysis of other TI-NET flow control and queueing problems.

It is mentioned that the software development associated with the experimental work in this thesis uses the software package developed by D. Weir as a base [50]. In addition, starting in February 1976, this thesis project received hardware and software support from Professor L.R. Morris in the speech processing area [47,51-53]. Also, starting in October 1976, support was received from Carleton University's Wired City Laboratory [55] and NASA AMES Research Center in California, in the form of assistance from personnel and in the use of facilities for satellite experiments [48,49].

1.5 Thesis Organization

1.5.1 Objective and Significance

The objective of this thesis is to investigate in a primarily experimental, and to some extent analytical way,

the transmission of encoded voice in the transparent intelligent network called TI-NET. The significance of the thesis is in contributing to a basis for an efficient integrated voice and data network, which is capable of providing cost benefits to its users through savings in network bandwidth; these savings are made possible by the statistical multiplexing of data with encoded voice.

1.5.2 Scope and Focus

This thesis is primarily related to computer communications, but also involves certain aspects of data transmission (terrestrial and satellite), speech processing and software engineering. The major area of emphasis is (to some extent) in analysing, (but primarily in) implementing and evaluating a capability for the transmission of encoded voice statistically multiplexed with data in a transparent intelligent network. The focal point for most aspects of the work is network delays - particularly variable delays - as they apply to encoded voice transmission.

As already indicated, the thesis is both analytical and experimental. With regard to the analytical work, an analysis of the TI-NET "admission" delay problem in its entirety is a difficult undertaking for two reasons. First, it appears that a new approach is required to analyse a system such as TI-NET which services a non-homogeneous population in parallel, rather than in series as in conventional packet switching. Secondly, the admission delays can be expected to be a

function of many parameters; some of these are: traffic balance (e.g. among synchronous and asynchronous traffic, as well as among different types of synchronous traffic), population sizes, link utilizations, priorities (e.g., for different types of synchronous traffic), and dynamic resource (channel capacity and reservations) allocation strategies.

In this thesis a beginning is made by dividing the problem into two parts, the first dealing with single hop traffic (not involving reservations) and the second with multi-hop traffic (involving reservations). Most of the analysis deals with various results for the single hop case. As might be expected closed form analytical solutions are not likely to be easily achieved in all cases, so that simulation results presented elsewhere are being relied on heavily [56,57]. However, analysis using simulation techniques does not form a direct part of this thesis. A start is made in the analysis of admission delays for multi-hop traffic in TI-NET, by presenting an approximate model in which the dependencies of admission delay (at a particular entrance node) on numerous network nodes are lumped.

With regard to the experimental aspects of the work, it is pointed out that the thesis does not concern itself particularly with speech processing, but rather with the transmission of processed speech in a shared network environment. However, since the experimental facilities available were not developed to the point where conflicts (delays) due

to sharing would arise, these delays were emulated by software means. In general, software techniques are employed in the thesis as a convenient implementation method, and measurement tool.

It has been mentioned previously that TI-NET, in its entirety, is intended to consist of an integrated terrestrial and satellite subnet architecture as well as satellite extension nodes. The specific area of research related to the integration of the terrestrial and satellite subnets is beyond the scope of the thesis. The work related to satellite transmission pertains mainly to satellite extension nodes, although it has provided useful results which are also applicable to some aspects of the more involved integrated system.

The motivation for including satellite transmission in the thesis, apart from economic and other considerations, is that it represents another dimension in the area of transmission delays. On the one hand, conventional packet switched networks are characterized by large fixed and variable end-to-end delays, while on the other hand a transparent network is expected to have comparatively small fixed and variable delays. However, somewhere in between, the use of satellite transmission to enable extension nodes to access a transparent terrestrial network, introduces a fairly large fixed component of delay due to satellite propagation, but still provides an advantage in terms of small variable delays; this is particularly important for voice applications.

A major purpose of the satellite transmission experiments was to gather data pertaining to characterizing the 12-14 GHz link error performance. This data is then used to evaluate the feasibility of a satellite subnet and satellite extension nodes for the TI-NET system, as well as error control based on terrestrial retransmission in this system. To this extent the thesis involves aspects of data transmission.

1.5.3 Contributions

1.5.3.1 Analytical [56,57]

A start is made in the analysis of "admission" delays for single-hop synchronous traffic in TI-NET, by providing the solution for the number of "bulk-units" in an $M/M/\infty$ system with non-homogeneous job population; the results are representative of TI-NET entrance node operation under light load conditions. The solution is provided for two system representations, one using a number of independent parallel service facilities - one for each job type, and another using a single combined service facility. For the parallel facility representation, two approaches - one algorithmic and the other more classical - are shown to provide the same solution. The parallel representation is also used to obtain the solution for the number of "bulk-units" in an $M/M/\infty/M$ closed system with non-homogeneous job population.

A first step is made in the study of multi-hop traffic in TI-NET by presenting an approximate model for such traffic.

This model accounts for the various delays associated with requesting, locating and delivering "reservations" to a TI-NET entrance node, before packetization and transmission of a message can begin.

1.5.3.2 Experimental [47-49]

A capability for statistically multiplexing data and encoded voice in an experimental network has been implemented and demonstrated. This capability consists of two parts: first a software implementation of an external link protocol which enables an encoded voice source to access the experimental network, in such a way that advantage can be taken of the 60 to 65% idle time in conversations (one way). Secondly, with regard to packetizing and transmission within the network, implementation of an internal protocol for synchronous voice and data traffic, makes possible the transparent flow of messages through the network.

A preliminary subjective evaluation of encoded voice transmission subjected to emulated "admission" delays, suggests that fairly large random delays are imperceptible provided they occur predominantly between phrases and sentences rather than (unnaturally) between syllables and words. This is consistent with TI-NET's fixed routing strategy and transportation mechanism, together with the use of a "hangover" after voice messages to ensure that "talk spurts" have ended.

The feasibility of 12-14 GHz satellite links for use with TI-NET extension nodes was demonstrated by the transmission of data and encoded voice over an experimental link,

and by a detailed link error performance study in terms of packet error free interval lengths (EFIL's) as well as the number of bits in error and bit EFIL's per packet. The results for packet EFIL's suggested a simple model for satellite transmission with terrestrial retransmission for error control purposes.

1.5.4 Thesis Structure

The thesis has six chapters. Chapter 1 has presented the nature of the problem, the motivation for its study and the proposed solution. In addition, the objective and significance, scope and focus, and contributions of the thesis have been summarized. Chapter 2 contains the background to past and present relevant research, including speech analysis/synthesis techniques, approaches to integrated data and voice transmission, and delay related issues in packetized voice transmission. Chapter 3 gives an overview of the TI-NET system design, upon which the following chapters of the thesis are based. Chapter 4 presents the analytical aspects of the thesis and includes certain results for single and multi-hop traffic in TI-NET. Chapter 5 presents the experimental aspects of the thesis; these include entering encoded voice into TI-NET, packetizing and transmission within the network, the emulation of certain network delays, and transmission over an experimental satellite link. Finally, Chapter 6 contains a summary and the conclusions, as well as suggestions for further work.

Associated with Chapters 4 and 5 are various appendices. The appendices for Chapter 4 contain supplementary analytical results, and details of derivations and discussions. The appendices for Chapter 5 contain supplementary experimental results and software and hardware documentation.

CHAPTER 2

RELEVANT PAST AND PRESENT RESEARCH

2.1 Speech Analysis/Synthesis Techniques

2.1.1 General

Since this thesis is concerned with the transmission of processed speech in a shared network environment, a brief review of applicable speech analysis/synthesis techniques, and their evaluation from the point of view of packetized voice transmission is appropriate.

The processing of speech signals has a history which goes back to the 1930's [58], but is still a subject of active present day research. Dozens of analysis, synthesis and encoding techniques have been, and are still being developed for various purposes, involving numerous technologies, covering a wide range of bit rates (or bandwidths), and providing various levels of quality and intelligibility. Therefore, this review will only emphasize those techniques which are most applicable to the work described in this thesis. Digital (as opposed to analogue) analysis/synthesis techniques are emphasized, since this is in accordance with present day technology. Secondly, the transmission rates of interest here are those which are compatible with synchronous transmission over voice grade lines, and range from 1.2 to 9.6 kb/s. Therefore such (high quality and intelligibility, but high bit rate)

"waveform preserving" schemes as linearly quantized and companded PCM, differential PCM (DPCM), delta modulation (DM), adaptive DPCM (ADPCM) and adaptive DM (e.g. continuously variable slope DM-CVSD), are not considered here since their bit rates vary from 16 kb/s for CVSD to over 64 kb/s for linear PCM [59]. In addition, the very low bit rate, but relatively poor quality schemes, based on stored phonemes and phoneme concatenation techniques are not considered [60]; at present the aim of such schemes is intelligibility only, and they are considered for the purposes of relatively specialized educational and other voice response systems.

The emphasis here is mainly on real time (conversational) analysis/synthesis techniques, of which there are many with transmission rates in the region of interest here [25,61,62]. These techniques are largely all based on a model of the human voice producing mechanism [5]; this model consists of an excitation function (glottle source) and a time varying filter (vocal tract). Physically, the glottle source excites the vocal tract by passing air through the vocal chords. For voiced sounds, the vocal chords vibrate with a fundamental frequency (pitch), while for unvoiced sounds (e.g. whispers) the vocal chords do not vibrate. The vocal tract consists of the mouth (and nasal passage - which is usually not modelled specifically) with the tongue and lips etc., acting as articulators, which modify the excitation function in accordance with the current utterance.

Thus the essential parameters that must be incorporated in the model are: the intensity of the excitation, whether it is voiced or unvoiced, and if voiced what the pitch period is; in addition, parameters are required which describe the vocal tract shape. Parameters are extracted at a rate which is fast enough to follow voiced/unvoiced transitions, and changes in pitch and vocal tract shape; a rate of about once every 20 ms is usually used (this also corresponds approximately to the maximum pitch period of interest). The parameters are then coded and formatted for transmission usually in the form of frames which are usually transmitted at a fixed rate of 40 to 50 per second.

At the receiver the frames are deformatted, the parameters are decoded and the speech is synthesized as follows. For voice sounds a periodic source with fundamental frequency equal to the received pitch is used to effectively excite a time varying filter, the parameters of which are determined from the received vocal tract information. For unvoiced sounds a noise generator rather than a periodic source is used for the excitation. In each case the reproduced speech level is set in accordance with the received intensity level.

The numerous analysis/synthesis methods which are based on a model of human voice production, may be divided into those employing frequency domain, and those employing time domain techniques. A few examples are reviewed below;

emphasis is placed on those techniques which have been, and are still popular and useful, as well as those which are more recent and indicate the trend that contemporary analysis/synthesis research is taking.

2.1.2 Frequency Domain Techniques

2.1.2.1 The Channel Vocoder [5,63,64]

The modern digital channel vocoder evolved from the analogue channel vocoder, in which the vocal tract is modelled by a bank of usually 16 to 20 parallel (≈ 200 Hz bandwidth) filters covering the audio band from about 200 to 3500 Hz. These filter outputs are then rectified and smoothed to remove residual excitation (pitch related) components, and the resulting spectral parameters are encoded for transmission, along with the excitation parameters. At the receiver a similar bank of filters is excited in accordance with received excitation information, which is first modulated by the received spectral information. All resulting filter outputs are then combined to form the reproduced speech.

In digital versions of the channel vocoder, the above analogue functions are replaced by equivalent digital functions based on the short term discrete Fourier transform and digital filtering. A more detailed description of a specific commercially available digital channel vocoder (the VADAC) is given later in Chapter 5 [70-73]. Modern

channel vocoders are typically frame synchronous devices which operate at 2.4 or 4.8 kb/s.

2.1.2.2 The Formant Vocoder [5,65]

In order to reduce the transmission rate of the channel vocoder to less than 2.4 kb/s, the approach is usually to encode only those channel outputs which are significantly large (peak picking). The ultimate in this approach is to recognize that the spectrum of the vocal tract is usually characterized by the presence of three or four peaks called "formants". Thus the formant vocoder estimates the amplitudes of these peaks and the frequencies at which they occur; these parameters are then encoded for transmission along with the usual excitation parameters. This approach can reduce the bit rate to 1.2 kb/s and lower, with a corresponding reduction in voice quality and naturalness.

One of the principal problems with this method is the difficulty in distinguishing formant peaks when they occur at frequencies which are close together. However recent work on formant based vocoders has dealt effectively with this problem by obtaining initial estimates of the formants from an all pole model of the vocal tract, and improving the estimates by iteration [75].

2.1.2.3 The Homomorphic Vocoder [5,65,66]

This vocoder is based on the premise that speech can be modelled as the convolution of the excitation function with

the impulse response of the vocal tract. Thus homomorphic analysis consists of deconvolving the speech wave into two functions, from one of which the excitation parameters can be determined, and the other which is a particular representation of the vocal tract impulse response ("cepstrum"). At the receiver the actual impulse response is determined from the transmitted cepstral parameters and is convolved with the local excitation function to reproduce the speech.

This technique has not received much attention in the past because of the large amount of processing involved. However, with the advent of fast LSI digital technology, this may no longer be the case. An advantage of the technique is that it provides one of the best known methods of pitch extraction, and it has been used for this purpose alone in vocoders which employ other means of parameterizing the vocal tract. For example, the digital channel vocoder described later in this thesis uses a pitch extraction method based on cepstral analysis. Homomorphic vocoders operate at bit rates of the order 3.6 to 7.2 kb/s and provide somewhat better quality speech than channel vocoders.

2.1.2.4 Voice Excited Vocoder (VEV) [5]

One of the most difficult aspects of vocoding is the need for accurate and reliable pitch extraction and voiced/unvoiced decisions, since poor performance in this regard results in very poor quality speech reproduction. In order to avoid this problem, it is possible to transmit -

unprocessed, the lower few hundred Hz of speech bandwidth which contains the vital pitch and voicing information. In addition, the speech spectrum is processed using the channel vocoder method, to extract the spectrum parameters corresponding to the vocal tract.

At the receiver, instead of controlling local excitation as in the other vocoders described previously, the received low frequency portion of the speech spectrum is first subjected to spectral flattening, then modulated by the spectrum parameters and finally used to excite a bank of channel filters covering the entire audio band. The outputs of these filters are then combined as in the channel vocoder. As a result of transmitting the low frequency portion of the spectrum unprocessed, the bit rate of voice excited vocoders is of the order of 9.6 kb/s, but the quality and naturalness of the reproduced speech is superior to that of the other methods described so far.

2.1.3 Time Domain Techniques

2.1.3.1 Linear Predictive Coder (LPC) [68,69]

The LPC vocoder is based on the assumption that a speech sample can be predicted from a linear summation of a certain number (usually 10 or 12) of suitably weighted previous samples. The weights correspond to the tap coefficients of a transversal filter in the LPC analyser. The coefficients are adjusted to minimize the short term (mean square) difference between actual and predicted speech samples.

Thus the coefficients correspond to a short term description of the vocal tract at discrete time intervals. The coefficients (or a transformation of them called the "reflection coefficients") are then coded for transmission along with excitation parameters.

At the receiver, the decoded coefficients are used in a recursive digital filter which performs the inverse function of the predictor in the LPC analyser, and which when fed by the locally generated excitation, reproduces the original speech.

It is mentioned that the analyser predictor is based on an (all pole) model of the vocal tract only, and therefore does not take into consideration the excitation of the vocal tract. As a result, the error between actual and predicted speech samples is substantial at pitch pulse instances, and quite small between pitch pulses. Therefore the predictor error (residual) may be used as a basis for pitch extraction (there are numerous other pitch extraction techniques available as well). A particular software implementation of an LPC synthesizer is described later in Chapter 5.

2.1.3.2 Adaptive Predictive Coder (APC) [74,75]

Analogous to the voice excited vocoder discussed previously, the APC technique is a time domain method which attempts to avoid the difficulties associated with voiced/unvoiced decisions and pitch extraction. This is

accomplished by transmitting both the analyser error signal as well as the extracted parameters. Included in this technique is an improvement in the LPC predictor to include the effects of excitation and thereby adapt itself in such a way as to minimize the residual error both at and between pitch pulse instances.

At the receiver, the residual error signal is used to excite a digital filter which performs the inverse operation of the analyser predictor. The coefficients of this filter are established from the received excitation and vocal tract parameters. As a result of transmitting the residual error signal, the APC technique requires a higher bit rate (7.2 to 9.6 kb/s) than the LPC technique, but in return provides superior quality and intelligibility, and is more robust with respect to a wide range of speakers, and the effects of an acoustically noisy environment at the analyser location.

2.1.3.3 LPC and APC Vocoder with Quantization and Encoding Schemes

Recently considerable attention has been given to the most appropriate choice of certain analysis/synthesis parameters, and to efficient methods of encoding them for transmission [77-79]. A useful objective of this work is to be able to reduce the transmission rate of higher quality, but higher bit rate techniques such as APC, to rates which are of the order of 2.4 kb/s.

In this regard it has been recognized that performing

certain logarithmic transformations on the analyser linear predictor coefficients (which results in "reflection coefficients"), is equivalent to non-linear quantizing (companding) of the linear LPC or APC coefficients. This has the effect of nearly halving the required bit rate.

Another proposed method of quantization uses DPCM to transmit the LPC parameters [80]. This scheme exploits the fact that the LPC parameters are themselves predictable from previous samples.

In addition, the redundancy between LPC parameters can be exploited. I.e., a particular parameter can be predicted not only from its past values, but also in some sense from values of the other parameters as well. This scheme involves the transmission (using DPCM techniques) of only the most significant eigen-vectors of the reflection coefficients.

An extension of the above and other techniques is to recognize that speech is a highly variable information rate source, and to employ encoding techniques which result in further reductions in average data rate [81,82]. A variable rate or dynamic encoding scheme transmits parameters more frequently when speech characteristics are changing rapidly (as in "phoneme" transitions), and less frequently when speech characteristics are relatively constant as in steady-state sounds. The simplest of such approaches is to recognize steady-state portions of speech and ensure that only new information is encoded. The synthesizer maintains the previous parameter values unless new values are transmitted.

Another aspect of variable-rate coding is called variable-quality coding, which trades off vocoder quality against transmission rate [44,83]. This scheme is considered to be applicable in a store and forward shared network environment; i.e. as the total network load decreases, higher bit rates with corresponding higher fidelity may be used, while as the load increases, reduced bit rates would help to relieve congestion and reduce delays at the expense of poorer fidelity.

2.1.4 Vocoder Evaluation

For the purposes of this thesis, vocoders are discussed here mainly in terms of their suitability for packetized voice transmission. Such factors as analyser digital output formatting and transmission rate, susceptibility to impairments, and quality and intelligibility are considered. Output formatting is typically that of a frame synchronous device, with bit rates of 2.4 to 9.6 kb/s - suitable for transmission over voice grade lines (with modems and line conditioning as required).

Later in Chapter 5 of this thesis the implications of the continuous frame structure of vocoder digital output formatting is discussed in detail. However it is pointed out here that in order to provide the capability of statistically multiplexing voice and data in an integrated network, it is necessary that only those vocoder frames which are active (transporting speech) be identified as such and

transmitted through the network. It is therefore evident that in order to supply the vocoder synthesizer (receiver) with a continuous frame sequence, idle frames (those transporting no speech) must be locally supplied in a manner which is consistent with arriving sequences of active frames. However, because of variable network delays, active frame sequences will in general arrive at arbitrary times with respect to the locally generated idle frames, and must therefore be suitably buffered and delayed to compensate for the random arrival times.

In the event that variable network delays are small (as they are expected to be using a transparent intelligent network approach, discussed later in this chapter), the number of idle frames required between active frame sequences is determined implicitly by the arrival of the active sequences. However, using this 'implicit' method when variable network delays are large (as they can be with conventional packet switching) may result in perceptible and possibly objectionable time scale distortion of the received speech messages as discussed previously in Chapter 1. Therefore timing information (time stamps) needs to be transmitted with the messages to effectively indicate how many idle frames should be inserted to at least approximately restore the original message inter-arrival times.

In this regard it is noted that the accuracy to which speech idle times need to be maintained depends strongly on the length of the idle times. I.e., within sentence idle

times (between syllables and words) are fairly short, and in order to be unobjectionable if not imperceptible, must be restored more accurately than the longer idle times between phrases and sentences. This is because a hesitation in the output speech seems more natural between phrases and sentences, since these are times when a speaker may very well be inclined to hesitate; however, the same is not usually true between syllables and words, and therefore such hesitations seem much more unnatural.

This may be interpreted in terms of vocoder output transmission speed and network packet size. I.e., even for fairly long packets (1000 bits) and fairly low vocoder speeds (2.4 kb/s), one packet accounts for about 400 ms of real time speech, which is of the same order as the duration of monosyllabic words. Therefore in general such packets would have to be timed fairly accurately. In this sense longer packets, and/or a lower vocoder speed, would be preferable and vice-versa; however, the use of long packets together with low speed terminals is undesirable in terms of network "entrance delay", as discussed later in this chapter.

Another network aspect which has implications in terms of vocoder frame structure and transmission rate is the occurrence of missing (lost, or too late to be used) packets. A missing packet has a more serious effect if it is long and the vocoder output speed is slow. For example, loss of a 1000 bit packet containing information from a 2.4 kb/s vocoder would eliminate over 400 ms of speech, which again

corresponds to the duration of mono-syllabic words. On the other hand, losing a 500 bit packet of information from a 9.6 kb/s vocoder would only destroy about 50 ms of speech and would likely be imperceptible. In these examples it is assumed that lost packets contain an integer number of vocoder frames; otherwise the vocoder receiver would also lose frame synchronization (this is briefly discussed again later), unless measures are taken to ensure that in spite of lost packets (or lost speech due to a buffer overflow anomaly), a continuous sequence of complete frames is provided.

With regard to transmission errors, most vocoders employ internal forward error control techniques based on known properties of speech, in order to minimize the effects of such errors. For example pitch fundamental frequencies are normally within the range of 50 to 350 Hz and changes in pitch (inflections) normally do not occur faster than a certain rate. Similarly, voiced/unvoiced transitions are also rate limited as are vocal tract parameter changes. Thus by the use of relatively simple rules and previous parameter values, transmission errors can be prevented from grossly affecting received synthesis parameters and hence quality and intelligibility.

In some vocoders the most vulnerable aspect of the output format is the frame structure, which however is protected against transmission errors by the usual forward and backward guard time strategies. Beyond this, failure to maintain a continuous frame structure causes the vocoder synthesizer to

lose frame synchronization, and to subsequently reframe; this is generally quite perceptible if not objectionable to a user since reframe times are typically of the order of one-quarter second. However, reframe times may be considerably longer if they take place during active speech; which may imitate the framing pattern.

Intelligibility refers to the extent to which a user is able to determine what has been spoken, while quality refers to the naturalness of the reproduced speech and the extent to which the user is inclined to accept it. The intelligibility of vocoders varies but in most cases is considered acceptable although somewhat poorer than that provided by standard telephone circuits. Quality on the other hand is usually considerably poorer than that of a telephone, especially for vocoder techniques which are based purely on a model of the voice producing mechanism such as the channel vocoder and LPC. The principal reason for this is the difficulty in accurately and reliably extracting the pitch, and in making voiced/unvoiced decisions. This is complicated by the fact that some sounds (such as the "zzz" in buzz) are actually combinations of voiced and unvoiced sounds. Another reason is that the models discard the phase relationship between the excitation and vocal tract functions and are thereby evidently incapable of reproducing the finer aspects of speech which give it a natural and human quality.

However, the more sophisticated vocoding techniques such as APC avoid these problems by transmitting enough of the

original speech waveform so as to preserve its more subtle characteristics, and thereby provide superior quality as well as intelligibility. The penalty of higher transmission rate can be countered by sophisticated quantization and encoding techniques. The cost and complexity of these techniques is no longer a serious problem with the advent of high speed LSI digital technology and micro-processor techniques [45].

2.2 Approaches to Integrated Data and Voice Transmission and Their Implications with Respect to Delay

Before concentrating specifically on issues in packetized voice (and data) transmission, which is done in the last section of this chapter, a brief and general overview of various approaches to integrated voice and data transmission is given here.

2.2.1 Circuit Switching [88]

Circuit switching of course refers to the present switched telecommunication network, which evolved principally in accordance with the requirements of voice traffic. However, the existence and accessibility of such networks made them an obvious first choice for data traffic as well. In circuit switching, end-to-end switching and transmission facilities are effectively reserved on a call basis for a pair engaged in communication. Once a call is set up, communication takes place over a fixed "reserved" route for the duration of the call. However, a call cannot always be established at the time its origination is attempted, due to insufficient circuits (servers) for the intended destination. Thus there

is a blocking probability (of perhaps one percent) associated with attempting to set up a call. If a call is not blocked (which is usually the case) a set up time (of typically a few hundred ms) is required to establish connection with the destination. Once a call is set up however, delay is basically fixed and minimal; throughput is also fixed and limited only by circuit capacity. Thus the characteristics of circuit switching are: access with blocking, call set up time, and fixed throughput and delay. These characteristics are considered to be acceptable for voice transmission, and adequate for certain bulk data applications for which message transmission times are typically much longer than the call set up time, and which have relatively low peak to average traffic ratios.

For real time data applications however, call set up, as well as low utilization and poor error performance are disadvantages of circuit switching. These disadvantages can be avoided by the use of leased lines together with a means of data concentration; however this represents a costly solution unless high utilization and extensive sharing of facilities is possible.

The characteristics of a basic leased line are: guaranteed connection, no call set up time and fixed throughput and delay. However, the sharing of such a facility by means of statistical multiplexing (in a store and forward concentrator) introduces variable throughput and delay

characteristics. (It is mentioned that throughput and delay are also affected by error control by retransmission on such a facility, although this point is not pursued here [107]). Such characteristics are considered acceptable for data traffic provided response time requirements for real time applications are met, and the number of buffer overflows and the cost of buffering is within acceptable limits. On the other hand, a facility with variable throughput and delay, is in general, not suitable for voice traffic, unless appropriate precautions are taken which minimize such variations.

2.2.2 Hybrid Switching

Hybrid switching refers to combinations of circuit switching (usually for voice, but perhaps for certain synchronous data applications as well), and either store and forward message or packet switching (usually for data, particularly asynchronous data applications). Such an approach attempts to retain basically separate switching technologies, with each having the distinctive characteristics appropriate to its traffic load, and reduces costs by sharing transmission facilities and making possible the sharing of processing power.

A popular method with regard to hybrid systems is associated with the U.S. Defence Communications Agency (DCA) and other organizations [2,21,38,89-92]. Although evidently not at the experiment or implementation stage as yet, this

method proposes the use of digital T1 carrier transmission facilities (1.544 mb/s). Adaptive sharing of this high bandwidth transmission capacity is facilitated by the use of a STDM frame (envelope) structure. Each channel is synchronously clocked and thereby divided into frames of fixed duration (e.g. 10 ms); each frame is further decomposed into a certain number of slots (i.e. slotted envelope). The slot size is chosen in accordance with the voice digitization rate and frame duration. Circuit switched voice requires a single slot per user per frame, whereas packet switched data may require multiple slots (because of queueing). Each frame is partitioned via a boundary mechanism into two distinct regions: one dedicated to circuit switched voice, the other for packet switched data. All circuit switched traffic which cannot seize channel capacity upon arrival is blocked (managed on a loss basis); all packet switched traffic that is unable to be transmitted is buffered (managed on a delay basis). The boundary may be "fixed" so that no dynamic sharing between traffic classes can take place, or "movable" whereby packet data can seize currently idle circuit voice slots during a particular frame.

Although this approach is a definite improvement over separate networks, its main disadvantage is that it does not provide statistical multiplexing for voice traffic, or between voice and data traffic on a message basis.

2.2.3 Conventional Packet Switching

Conventional packet switching is exemplified by the ARPANET and is strongly characterized by variable throughput and delay [22]. As such it is suitable for data traffic but has disadvantages for voice. Nevertheless, because of the advent of new applications, and the economic advantages of having both integrated switching and transmission facilities, a great deal of attention has been given recently to the prospects of packetized voice transmission [3,20,22,25,27,44,87,93]. The problems associated with these endeavors are reviewed later, and strongly suggest a trend away from a strictly packet switched approach, towards an approach which to some extent resembles circuit switching, but still provides the capability for sharing resources efficiently. This trend is evidenced in ARPA by the proposal of a "packetized virtual circuit" approach [25,93].

2.2.4 Transparent Intelligent Network Approach

2.2.4.1 General

In order to readily and efficiently accommodate both data and voice traffic in an integrated switching and transmission system, a trend away from the characteristics of variable throughput and delay, toward fixed throughput and delay, is indicated. This trend dictates that the integrated system objectives should include transparency in time (which to a large extent implies transparency in protocol as well). Another essential objective is network intelligence, which

makes possible efficient sharing by both voice and data traffic, and provides such services as error control, flow control and routing for all traffic types.

Two important factors which are responsible for the distinctions between a transparent intelligent network and a conventional packet switched network, are packet structure and routing. Conventional packet switching uses relatively large private packets for each user, while transparent intelligence networks transmit public packets at regular intervals, which contain many relatively small packets - one for each user; this serves to eliminate a large component of delay required to assemble a private packet.

Also, in packet switching, packets making up a message are normally sent over more than one route toward the message destination in order to reduce transit delays. However, this approach causes packets to arrive at the destination at random times, and possibly in the wrong order; therefore delays are encountered in the reassembly of multipacket messages. Furthermore, when adaptive routing is used to avoid congestion, undesirable side effects such as "loops" and "lockups" may occur which must be avoided at the expense of routing overhead [54].

In order to achieve transparency in time and maintain simplicity with regard to sequencing, a fixed routing strategy with alternate routing for failures is appropriate. Congestion may be controlled by a more sophisticated flow control scheme.

An early network which employed the principles of fixed routing and a public packet structure is TYMNET [94-96] - although this network is not considered to be transparent in time (because of the low speed - 2.4, 4.8 or 9.6 kb/s - links interconnecting its nodes, and other reasons). Also, it is used mainly for low speed asynchronous traffic. However, in TYMNET, a version of shortest-path or least-cost routing is used to determine the appropriate path from source to destination node over which to send messages for a particular user. The path for each user is selected on a call basis, and in this sense routing is similar to that used in circuit switching; however, in TYMNET messages are switched in a store and forward manner from node to node, and the path can be shared by other users.

Messages arriving at a node (TYMSAT), whether from local terminals or from adjacent nodes, are stored in character buffers, and each message has a channel (path) number associated with it. Message transmission to adjacent nodes is accomplished by assembling a block of characters (public packet) for each outgoing link. A block is assembled by searching through the buffers on a first-come, first-served basis for characters belonging to messages with channel numbers associated with the link on which the block is to be transmitted. The format of the block consists of a header (with synchronization pattern, block size, block number and acknowledgement); user messages (each with a channel number

and size) and error detection characters. -The total maximum block size is 66 (8 bit) characters, and blocks are transmitted one after the other on for example a 2.4.kb/s link. Thus the maximum time interval between buffer scanning is 220 ms, which is fairly slow, but adequate for low speed asynchronous terminals. By using a higher link speed and transmitting blocks at regular intervals, transparency in time as far as entrance into the network is concerned, can be achieved for synchronous as well as asynchronous traffic; this concept will be discussed further in the overview of TI-NET in Chapter 3.

Three other more recent networks which have to some extent adopted the transparent intelligent approach are the BTL SPIDER, TRAN M3200 and the CODEX 6000 networks. These will be briefly described here, while in the next chapter an overview of TI-NET is given; this overview suggests a broader application of transparency and intelligence.

2.2.4.2 SPIDER Network [97,98]

SPIDER is an experimental packet switched network at Bell Laboratories which is used to interconnect several medium scale laboratory computers, and enables them to share various common resources (disks, printer-plotter and fast processor). The network topology is based on a loop to which the computers and resources are connected through Terminal

Interface Units (TIU). The TIU is responsible for packet formation, error control and flow control. Also connected to the loop is a switching center which controls loop access, call set up and packet routing. Potentially, numerous loops can be interconnected by transfer stations.

Transparency in time is achieved as a result of the high speed T1 carrier loop (1.544 mb/s) and the fact that user computers are connected to the TIU's through high speed (500 kb/s) asynchronous lines. Conversations between computers etc. are based on virtual channels which specify the route only, and have no dedicated bandwidth. Data are transmitted and received in packets each bearing an address. Each packet occupies one slot on the T1 loop and empty slots have zero in the address position, so that the loop can be utilized on a demand-shared basis.

A disadvantage of SPIDER as it stands is that it is not terminal oriented, although presumably terminals can gain access through computers. Another disadvantage is that loops are vulnerable to malfunction in a way that other systems are not, since one TIU carries the traffic for many other TIU's. In order to prevent malfunction at one user site from denying service to, or seriously inconveniencing others, rapid automatic error detection and loop reconfiguration is required.

2.2.4.3 TRAN M3200 System [99-101]

The Computer Transmission Corporation M3200 Network Switching and Management System uses the "Pacuit" concept,

which is an example of a hybrid system in which packet and circuit switching are combined. TRAN's objectives are minimum transmission delay, optimum bandwidth usage, flexibility to support terminal protocols, and error checking and recovery. Thus TRAN attempted to separate external (i.e. network access) and internal data transfer mechanisms in order to achieve transparency.

Synchronous traffic is circuit switched while asynchronous traffic is packetized using public 'pacuits' (or alternatively, customer private packets); however, 'pacuits' are only processed at source and destination nodes, whereas intermediate nodes operate in a pure circuit switched mode for through traffic consisting of both synchronous data and 'pacuits'. Also, switching is based on calls, and routing is fixed between source and destination nodes except for failures.

With regard to asynchronous traffic, a 'pacuit' is a highly efficient public packet structure containing data from many asynchronous terminals; however, the data in a particular 'pacuit' must be from terminals which are all at the same source node, and communicating with other terminals which are all at the same destination node. Thus intermediate nodes are not required to buffer, and error control (for asynchronous traffic) is performed on an end-to-end basis. 'Pacuits' are transmitted at regular intervals and contain all terminal data in that interval. There is minimal overhead for inactive terminals. Each user's data is identified

by its relative location in the 'pacuit', and each amount of data is specified by a byte count. Data compression is provided as a network function to enhance the 'pacuit' efficiency.

In general, small node buffers are required for asynchronous terminals and buffer size is dependent on the 'pacuit' generation interval. This time interval also determines the 'pacuit' overhead, resulting in a tradeoff between overhead and buffers. Any data transmitted by a user (i.e. after the call is set up) will be delivered to the receiving station, leaving any line protocols up to the two connected stations.

The complete network uses a synchronous master-clock operation. On the trunks between nodes, synchronous traffic and 'pacuits' are time division multiplexed. Time slots are dynamically reassigned depending on the traffic offered. This dynamic bandwidth allocation is coupled with adjustment of the terminal clock in order to optionally provide flow control for synchronous users, which consists of emptying the terminal buffers in accordance with available bandwidth. Error control for synchronous traffic is left to the users.

2.2.4.4 CODEX 6000 System [102-104]

The Codex Corporation has recently introduced the 6000 Series of Intelligent Network Processors which is another example of a hybrid system which provides circuit switching for synchronous traffic and packet switching with error

control for asynchronous traffic (as well as character oriented synchronous traffic using BSC). The objectives of the Codex system are similar to TRAN's M3200 and include transparency, efficiency and minimal delay. The packet structure is similar to that of TRAN in that data from many asynchronous users is contained in a single packet and packets are transmitted at regular intervals. An important exception however is that packets may contain data from various terminals which do not necessarily have common source and destination nodes. That is, packet disassembly and reassembly are performed at intermediate nodes so that individual user's data may be routed to different destinations. Transmission efficiency can also be improved by translating each user's data characters in a packet into variable-length prefix codes (e.g. Hoffman coding).

At present user-initiated switching, adaptive routing and congestion control remain to be implemented.

2.3 Delay Related Issues in Packetized Voice Transmission

In Chapter 1, the subjective effects of various anomalies on packetized speech were reviewed. These anomalies are caused by variable network delays, and arose specifically in ARPA. However, they would be expected to occur in any network using conventional packet switching for voice transmission. Also, in the first section of this chapter, vocoders were discussed in terms of various impairments including variable

network delays, the above mentioned anomalies, and transmission errors. In the present section, the relevant aspects of conventional packet switching are reviewed in more detail, and the fundamental problem areas with regard to encoded voice transmission are pointed out.

2.3.1 Overhead Versus Packet Formation Delay

The main idea in a conventional packet network is that when messages from terminals enter the network through an entrance node, they are broken up into packets. Overhead bits are then added to each packet in order to implement such network functions as addressing, packet sequencing and error control. Entrance delay is incurred in the formation of packets. For example it takes over 400 msec to load enough data from a 2.4 kb/s terminal into a node to form a 1000 bit packet (e.g. the typical packet size in ARPANET). Therefore, it is desirable to use packets which are as short as possible. However, the packet overhead then becomes high resulting in inefficient link utilization.

Recently a proposal has been put forward for dealing with this, and other problems associated with voice transmission in ARPA [25,93]; the proposed solution lies in what is referred to as a packetized virtual circuit approach. With this approach, a fixed packet size of 128 bits and 25% overhead is contemplated, and necessitates a virtual connection convention, with all packets belonging to a particular destination following the same route through the network. This permits the use of abbreviated headers (control, sequence

number and check bits - if used), since much of the packet header information (type, priority, source and destination address) is fixed, and can be communicated at the time the virtual connection is set up.

2.3.2 Topology and Connectivity

Once packets are formed, they are put on outgoing queues for transmission to other nodes in the network. Depending on the network topology, packets may have to traverse many nodes before reaching their destination. For example in a distributed partially (k) connected network such as ARPA [105], as many as a dozen or more nodes may be encountered in some cases [21,25]. Since queues may potentially develop at many or all of these nodes, such a scheme is disadvantageous for voice traffic. Thus a topology which minimizes the number of nodes to be traversed is indicated. For a distributed network, this implies an increase in connectivity.

On the other hand, a centralized or star topology by its very nature involves relatively few nodes through which a message must be transported before it reaches its destination. This topology is discussed further in the next chapter.

2.3.3 Routing, Packet Sequencing and Message Reassembly

After a packet is received at a node and its integrity has been verified, its address is examined and it is either put on a queue for transmission to some other node or is delivered to a local terminal. Based on such factors as node

and link failures, and congestion, each node may make independent decisions as to the best route to use in forwarding packets. In addition, queuing delays may be encountered at some of the nodes traversed by a packet, since each must wait its turn to be forwarded to the next node. It is therefore evident that since the packets making up a message may not all follow the same route, they may arrive at the destination out of sequence. This necessitates reordering of packets at the destination, and reassembly of multipacket messages. Even though some packets of a message may arrive in "reasonable" time, one or more may be "late" or "lost". In the previous case the entire message is normally delayed by an amount related to the latest packet arrival time, while in the latter case the message cannot be completed and is normally NAK'd by the customer.

For data transmission, the above situations, although undesirable, can be tolerated. For encoded voice, however, the implications are different since late packet arrivals distort the time base of reassembled speech messages thereby affecting both intelligibility and quality. This affect is sufficiently important that if a packet is later than a certain amount, it must be judged to be lost and the speech message is reassembled without it. The result is gaps in the reproduced speech, called "glitches", due to the missing packets.

The routing techniques in a network can be selected from a wide range of alternatives varying from full adaptive

routing to one with a fixed virtual circuit through the network [96,106]. As indicated earlier, the design of the packet header is involved with the routing procedure used.

Thus in order to minimize packet overhead and eliminate sequencing problems, message reassembly delays and other complications associated with adaptive routing, a fixed routing strategy, with alternate routing for failures, is considered to be more suitable for an integrated voice and data network.

2.3.4 Network Protocols

2.3.4.1 Error Control [107,108]

Packets are transmitted between network nodes in a store and forward manner. That is, the sending node must store a copy of a transmitted packet until the receiving node positively acknowledges (ACKS) that the packet in question has been received correctly, otherwise a negative acknowledgement (NAK) is used to request the retransmission of the packet. Depending on the number of retransmissions allowed per packet and other factors, such as congestion, this method of error control can cause occasional very long transit delays which result in "late" packet arrivals.

However, the requirement for error control is in general not nearly so critical for voice as it is for data traffic. Therefore error control for voice traffic may be dispensed with, particularly in cases where the implementation of such

a control would introduce substantial fixed and variable delays.

On the other hand, in the next chapter an error control scheme for synchronous traffic is described which involves a relatively small fixed delay. In this case error control for voice traffic can be provided as an option; this may be useful in specialized very low bit rate voice applications, if the voice encoding is not suited for transmission and therefore quite vulnerable to transmission errors. However, the voice encoders of interest in this thesis (and described earlier in this chapter) are fairly immune to transmission errors.

2.3.4.2 Flow Control [96,106,109]

Flow control is discussed here in terms of the objective of avoiding network congestion, and hence long transit delays, while at the same time attempting to maintain continuous and high throughput. Established flow control methods may be categorized as local, end-to-end, global or combinations of these.

Local control is applied by a node on the basis of its own traffic statistics, as well as, protocol messages from immediate neighbouring nodes; these messages may request a reduction of traffic over a particular link, or pass on information about delays which are likely to be experienced at a certain node.

End-to-end control applies to virtual circuits between source and destination nodes, and is effected by limiting the

number of messages that can be in transit on any such circuit; i.e. the destination node signals the source node by means of an end-to-end protocol (e.g. ARPA's request for next message - RFNM) when another message can be accepted [110].

Global control is applied to an entire network, and attempts to prevent generalized overloading which can cause a reduction in throughput or eventually a complete seizure. A unique global scheme is called "isarithmic" congestion control [111], and is described briefly here since the flow control scheme used in TI-NET, is similar in limited respects to this method.

The isarithmic method of congestion control allows a network to place an upper limit on the number of packets in the network at any instant. This is achieved by providing each node with a permit allocation. A permit is picked up on entry to the network and released on delivery; if there are no permits available at the entry node, the packet must wait, and this constitutes an admission delay.

To make permits available where needed, they may be distributed around the network in random manner or according to well defined rules, when not in use. Thus during light traffic periods, permits are distributed around the network in special packets, while during heavy traffic, redistribution can be done mostly in the headers of data packets. During steady state, performance is good since arriving packets provide empty permits for traffic in the opposite direction.

However, if the traffic pattern is asymmetrical, there is a tendency for permits to become concentrated in one section of the network, and they may filter back very slowly. Thus for transient changes in traffic, no permits - or very few - may be located where they are required, which can result in long admission delays.

The flow control methods just discussed were developed largely from the point of view of data transmission in packet networks. With the advent of packetized voice transmission however, other schemes have been considered which take into account the characteristics of voice traffic. For example, since voice transmission is much more sensitive to delay variations than data, a common policy is to give voice messages priority, in order to minimize these delays [87]. On the other hand, voice traffic is much less critical than data in terms of information being received in error or being lost. Thus in the event that congestion in a particular node is imminent (i.e. a shared data and voice buffer is about to overflow, or a delay constraint for voice is about to be exceeded) an approach that has been considered is to simply discard a certain portion of voice messages in order to relieve the situation [25]. Such a measure however, is likely to be perceptible if not objectionable to listeners, and is only acceptable if it is rarely required, or if objectionable effects can be adequately eliminated by suitable algorithms which fill in gaps (glitches) at the listener's location.

A more sophisticated (and more complicated) approach is to exploit the trade off which exists between voice quality and the degree to which voice may be encoded [44]; in general variable-quality coding implies that higher quality speech is associated with a higher bit rate, and vice-versa. Thus voice traffic considered as a highly variable-rate source, can potentially provide considerable flexibility in handling fluctuating traffic conditions. I.e., when incoming voice or data traffic begins to congest the network, flow control procedures can be invoked which lower the voice transmission rate but still allow intelligible speech.

It has been mentioned previously that a packetized virtual circuit approach to routing has been proposed to more readily accommodate voice traffic in ARPA [93]. Associated with this scheme is a discipline called "statistical flow control". This control method has been devised since the flow control mechanisms normally employed in packet networks introduce delays and considerable overhead, which are particularly inappropriate in a network intended to handle a high percentage of voice traffic.

Statistical flow control is not performed on a packet-by-packet basis (which is usually the case), but rather the assignment of virtual connections to various network links is done in such a way as to reduce the probability of internal network overflows to values which are small. This is

accomplished by using average rate requirements (rather than peak - as in circuit switching) and statistically obtained safety margins in determining whether or not sufficient capacity remains on a link being considered as a possible route for a connection. It is proposed that this technique be supplemented by the discarding of voice packets when necessary, as well as a form of end-to-end control on the virtual connections.

CHAPTER 3.

OVERVIEW OF TI-NET SYSTEM DESIGN

In the thesis introduction, a brief description was given which revealed that TI-NET in its entirety, consists of an integrated terrestrial and satellite subnet architecture, as well as satellite extension nodes [4,46]. The purpose of the overview which follows is to provide the necessary background for the analytical and experimental work presented in later chapters. The overview touches on most aspects of TI-NET, but those aspects which pertain to the above mentioned work are emphasized. Accordingly, the overview concentrates on the terrestrial subnet [47,48] and to some extent on satellite extension nodes [48,49]. The features and advantages of TI-NET for encoded voice transmission are summarized [48,56].

3.1 Subnet Access and Protocol Transparency

TI-NET offers a solution to the problems of transparency in time and protocol. The objective in protocol transparency is to have an intelligent network for sharing facilities and providing such functions as packet sequencing, error control and flow control, while at the same time providing an interface as simple as possible - ideally accepting only a serial bit stream consisting of user's data and protocol. Thus TI-NET accepts user data as a serial bit stream through a

simple physical RS 232-C interface, and attaches no specific meaning to the format of data, except to distinguish between customer busy and idle periods. For asynchronous traffic the start/stop bits perform this function, while synchronous messages are delimited by the user's protocol; for example by flags in HDLC and SDLC [123,124].

This is all the user information that is required when operating in the permanent circuit mode, except for a communication link between the customer and TI-NET, and either a small software package or verbal communication to perform morning sign-on and evening sign-off; for switched circuit users, a similar capability is required for call set-up.

It is mentioned that the intention in TI-NET is to also accommodate users with other than a simple serial bit stream, so that for example a host computer with high speed line and the new X.25 international standard access protocol, may communicate with remote terminals using TI-NET.

3.2 Transparency in Time and Packet Structure

Of particular concern in this thesis, from the point of view of encoded voice (as well as data) transmission, is the issue of transparency in time. The objective is to have transparency in time manifested in TI-NET by both minimal entrance delays into the network, and low average message delays through the network. The extent to which such low delays can be achieved will be influenced by such factors as the packet

structure, topology, flow control scheme and transportation mechanism proposed for TI-NET.

The packet structure is based on public "multi-user" packets - MP's containing user "minipackets" - mp's. Thus a user's message is effectively broken up into mp's. mp's are formed at specific intervals in order to transport data from source to destination node. For example, an asynchronous mp would carry one byte of user data every 100 ms, for a 110 baud asynchronous terminal. Similarly, a three byte, synchronous mp every 10 ms supports a 2.4 kb/s synchronous terminal. The format of the synchronous mp is discussed in detail later but basically consists of a data field followed by a three bit trailer which is required for flow control and other purposes (see Figure 3.1(a)). The asynchronous mp structure consists of a one bit header which may be followed by a two bit character count and then the variable length (one to four character) data field (Figure 3.1(b)). The header bit indicates the presence (1) or absence (0) of a character count plus data field. Thus an asynchronous mp for each user is transmitted repetitively even for durations when there is no user data, but during these times only the header bit (0) is transmitted.

As mentioned previously, synchronous traffic consists of messages delimited in some manner by the user's protocol (e.g. by HDLC flags). Only the messages need to be transmitted in mp's. Thus, when a synchronous terminal is idle,

T = 3 BIT TRAILER

DATA = DATA FROM 1 SYNCHRONOUS USER

(a) SYNCHRONOUS mp STRUCTURE

H = 1 BIT HEADER

CC = 2 BIT CHARACTER COUNT

DATA = DATA FROM 1 ASYNCHRONOUS USER
(1 to 4 CHARACTERS)

(b) ASYNCHRONOUS mp STRUCTURE

F = HDLC FLAG BYTE

H = HEADER

BC = 2 CRC-16 BYTES

DATA = DATA FIELD CONTAINING USER mp's

(c) MP STRUCTURE

FIGURE 3.1 TI-NET PACKET STRUCTURE

no mp's are sent. The current status of a terminal is stored in each node's destination table in the form of an "activity field". This activity field indicates those users that are currently active - i.e. transmitting a message - and is updated using the synchronous MP header described later. The same mechanism is used for asynchronous traffic, but on a per call rather than per message basis. The philosophy behind the different synchronous and asynchronous mp formats is discussed in a later chapter.

The size of the data field and transmission time (service) interval for all terminal speeds accepted by TI-NET are:

asynchronous:	110 baud = 1 byte/100 ms.
	150 baud = 1.5 bytes/100 ms. (average)
	300 baud = 3 bytes/100 ms.
	600 baud = 1.5 bytes/25 ms. (average)
	1200 baud = 3 bytes/25 ms.
	1800 baud = 4.5 bytes/25 ms. (average)
synchronous:	1.2 kb/s = 1.5 bytes/10 ms.
	2.4 kb/s = 3 bytes/10 ms.
	4.8 kb/s = 6 bytes/10 ms.
	9.6 kb/s = 12 bytes/10 ms.

While a synchronous message is being transmitted through TI-NET, mp's will always be filled. However, the last mp may be partially filled. A protocol is described later in the thesis for "padding" such mp's and forcing their transmission.

To provide sequencing, error control, flow control and routing of mp's, multi-user packets - MP's are used. As shown in Figure 3.1(c), MP's are delimited by HDLC flags and bit stuffing is used to implement data link transparency. An MP consists of a header for control purposes, a data field of mp's, one for each user, and a trailer of 2 CRC block check characters for error detection. There are synchronous MP's in which the mp's contain data and protocol from synchronous users; similarly, asynchronous MP's have mp's containing data and protocol from asynchronous users. An example of a synchronous MP is discussed in the next chapter in connection with Figure 4.6, and is seen to contain a number of mp's from various 1.2, 2.4, 4.8 and 9.6 kb/s synchronous terminals. The MP header consists of MP number, acknowledgement and priority/format (e.g. MP type and user status) bytes.

The synchronous and asynchronous type MP's are formed and transmitted alternately every 5 ms - synch, asynch, synch, asynch - etc. Thus synchronous terminals are serviced at a rate of once every 10 ms. Therefore such a terminal only loads data into a node for 10 ms to fill an mp, at which time transmission can begin; e.g., a 2.4 kb/s terminal fills 24 bit mp's every 10 ms. In general, entrance delay for synchronous messages varies between 1 and 2 service intervals (10-20 ms) depending on message arrival times with respect to the 10 ms node servicing times; this is discussed and illustrated in more detail in the next chapter (see Figure 4.2(c) and (d)), in which TI-NET delays are examined.

In general terminal status is either active or idle; however, as indicated previously, asynchronous and synchronous terminal status is on a call and message basis respectively. When a terminal becomes active or inactive, the priority/format field is used to update node activity fields as to the new terminal status. Thus mp's are continually transmitted for an active terminal, and a TI-NET node can always identify the owner of an mp, since each active user's mp (as indicated by the activity field) will always be in the same relative position in the MP. In addition to using the priority/format field for user status, it is also used for MP type, flow control and service type (e.g. loop back and trace).

The synchronous MP header consists of MP number and priority/format bytes. The asynchronous MP header consists of these two bytes plus an acknowledgement byte for error control. MP's are numbered sequentially for error correction. If there is a link failure, MP's are rerouted and the MP number is used to maintain sequencing. Error control is also based on the MP.

Although user data enters a TI-NET node at low (terminal) speeds, it is transported to its destination in mp's at regular time intervals (e.g. every 10 ms). In terms of entrance delay, this is equivalent to entering a block of data (to form a private packet) over a high speed line. This not only minimizes entrance delays, but also minimizes node storage requirements since a complete block of data forming

a private packet (e.g. 128 bytes) does not have to be buffered as in conventional packet switched networks. In addition, an approximately constant MP overhead of about 25% is maintained; in this regard it is mentioned that the TI-NET (star) link capacities are 112 kb/s, 56 kb/s for synchronous and 56 kb/s for asynchronous traffic - this is discussed again later with regard to Figure 3.2.

3.3 Terrestrial Subnet Architecture

3.3.1 Topology and Transportation Mechanisms

The TI-NET terrestrial subnet was designed primarily to service real time data application categories. The topology is hierarchial as shown in Figure 3.2. The first level is the star net with a central node and various star nodes with near neighbour links for alternate routing in case of link failure. The second level is a system of star nets with their central nodes fully connected. Thus it is possible to reach any node from any other with a maximum of only three hops. Alternate routing in case of link failure is also provided at this level.

A disadvantage of this topology is the high cost of fully connecting many central nodes with high speed links (see Figure 3.2). However, as will be evident somewhat later in this overview, the terrestrial subnet architecture is integrated with a satellite subnet architecture. This integrated approach provides a variety of capabilities ranging from (a) full connection of a small number of central nodes with

FIGURE 3.2 TERRESTRIAL SUBNET TOPOLOGY:
FULLY CONNECTED STARNETS

high speed terrestrial links, to (b) full connection of a relatively large number of central (and star) nodes with satellite links. In case (b) the central nodes would only be fully connected with low speed terrestrial links to handle acknowledgements and short protocol messages. Case (a) may be adequate for a country such as Canada where most traffic tends to be concentrated near the east and west coasts, so that relatively few central nodes (e.g. one or two near each coast) are required.

The advantage of a (pure) starnet (with no near neighbour links) is that total average network delay (queueing plus servicing) is less than that of other k connected networks for light network loads (and under various other conditions). However, this result is not directly applicable here since it is for conventional store and forward networks, whereas TI-NET is transparent with respect to service time, and would only encounter network delays due to queueing.

However, for heavy network loads queueing delay is dominant and the results indicate that the delay performance of a starnet degrades beyond that of other k connected networks. Thus in TI-NET a global flow control scheme based on reservations is proposed to control the flow of traffic within the network.

In general this scheme can be expected to reduce steady state queue sizes and waiting times within the terrestrial subnet; however as a result, admission delays will be incurred on the periphery of the network.

A disadvantage of the starnet is the reliability of the central node. In TI-NET reliability is provided as a result of the integrated terrestrial and satellite subnet architectures, outlined later in this overview.

The mechanism for transportation of traffic through TI-NET is such that each node in the terrestrial network transmits MP's at the same time every 5 ms in a synchronous fashion. The MP's travel between two nodes in a store and forward manner carrying mp's. An mp belonging to a particular message is transported (along with other mp's belonging to other messages) in successive MP's through the network from source to destination node over a fixed route (except in the case of a link failure) using different MP's in each hop of the route; this avoids packet sequencing problems and message reassembly delays.

3.3.2 Error Control

Error correction is based on retransmission. Thus each node is responsible for storing the MP's it transmits until it receives the appropriate acknowledgement. Since the characteristics of synchronous and asynchronous traffic are different, initiation of an MP retransmission is also different.

Retransmission of synchronous MP's will cause gaps in the continuous mp flow for a message. In order to maintain

message continuity, it is necessary to initially delay transmitting data to the destination customer. Thus with a delay of 4 mp's (40 ms), TI-NET can support 1 central link retransmission and 1 or 2 star link retransmissions. Loss of message integrity will depend both on the message size and the number of retransmissions (i.e. error-free interval on the link). There is no initial delay with asynchronous data.

Given that the continuity of synchronous mp's must always be maintained, a synchronous MP is retransmitted as soon as possible in order to minimize error correction delays. To achieve this, an explicit NAK/implicit ACK scheme is used. When a synchronous MP is received in error, the asynchronous MP that is currently being transmitted is aborted (via HDLC abort) and a short null NAK packet is sent. This assumes propagation plus transmission delays of less than 2.5 ms within starnets. The original node will receive the NAK before its next synchronous MP transmission and will retransmit the MP in both synchronous and asynchronous time slots. A copy of this synchronous MP is also sent in a circuit-switched mode through the near-neighbour star node (the priority/format field indicates retransmission through the alternate path). Thus a NAK results in three retransmissions in the starnet which should ensure at least one correct transmission.

Asynchronous MP's do not have such stringent time constraints and can use an explicit ACK/explicit NAK scheme. Error retransmission is initiated via the acknowledgement in the asynchronous MP header and retransmission occurs in an asynchronous MP time slot. The acknowledgement consists of two bytes, one byte is the MP number of the last MP correctly received and the other byte is the acknowledgement vector. The vector covers 8 MP's, where 1 indicates ACK and 0 indicates NAK. The high-order acknowledgement bit in the vector corresponds to the MP number of the last MP received correctly. Thus if this bit is 0 (i.e. NAK), then the last MP was received incorrectly or not at all.

The links between central nodes typically operate at twice the speed of starlet links (see Figure 3.2). Thus in a 10 ms time slot the synchronous and asynchronous MP's are transmitted in the following sequence:

SYNC/SYNC/ASYNC/SYNC

These MP's are the same size as those on star links. However, when the distance between central nodes is large, the propagation, and hence retransmission time is long and the number of practical retransmissions is limited. Thus triple redundancy can increase the probability of an error-free synchronous MP. Retransmission on a central link is similar to a star link in that a synchronous MP is retransmitted in double redundancy but there are no retransmissions through an alternate path.

3.3.3 Flow Control

Since incoming and outgoing terminal speeds are the same, flow control in TI-NET amounts to control of congestion in the network. Initially flow control will apply only to synchronous traffic. Furthermore synchronous traffic is always given priority over asynchronous traffic.

Channel flow control is based on reservations which are associated with the outgoing link (to another node) from a central node. A reservation "quantum" of channel capacity corresponds for example, to 12 bits of user data in a synchronous mp, travelling from a starnet central node to an exit node (i.e., travelling on the last link to the user's destination). When a reservation quantum for a star link is passed from one central node to another central node, it also reserves bandwidth on the central link.

If synchronous terminals are serviced every 10 ms, the number of reservation quanta required per message for 1.2, 2.4, 4.8 and 9.6 kb/s terminals are respectively 1, 2, 4 and 8; hence a reservation quantum corresponds to 1.2 kb/s of channel capacity. Reservations are secured on a message basis (one per message) by an entrance node, and enable a user at the entrance node to send his entire message to another user at a certain exit node. After the message is sent, the reservation is left at the entrance node to be used by subsequent messages from users at the same node, or to be cancelled later by the central node, and reassigned to messages from other users originating at other nodes. This is

possible since reservation quanta can be passed between nodes very quickly.

Recall that it is possible to reach any TI-NET node from any other with a maximum of 3 hops. Reservations are not required for single hop traffic but only for traffic traversing 2 or 3 hops, and then, the reservations are only for the last 1 or 2 hops to be traversed respectively. For single hop traffic, or the first hop of multi-hop traffic, there must be sufficient capacity available on this hop. Hence, for single hop traffic, admission delay is the time the user must wait for sufficient capacity to become available. For multi-hop traffic, admission delay can be the time for sufficient capacity to become available on the first hop, or the time the user must wait to secure reservations for the remaining hops, or both. Reservations are obtained by an entrance node by means of a subnet flow control protocol; the admission delay incurred in obtaining reservations consists of: the time for the entrance node to request them, the time for the central node to find them (if it is not holding any) and finally, the time for the network to deliver them to the entrance node.

In general, admission delays may be controlled by traffic engineering as well as by giving priority to synchronous traffic over asynchronous traffic. It has been mentioned that synchronous and asynchronous MP's are transmitted alternately every 5 msec. An effective way of realizing the

above priority scheme is by allowing synchronous MP's to preempt asynchronous MP's as required. As a result of this priority scheme, significant asynchronous queues can be expected to exist within the subnet. However, this is compensated for to some extent when reductions in synchronous traffic occur as a result of blocking at the entrance due to lack of reservations; during these times the asynchronous MP's would not be preempted, and this would tend to reduce the asynchronous queues.

3.4 Satellite Subnet Architecture

All TI-NET nodes are also earth stations and the satellite, which operates in a TDMA mode, provides a fully connected network with 12-14 GHz links. These links, although susceptible to signal absorption during heavy rain or snowfall, are protected by rate 1/2 convolutional encoding with soft decision Viterbi decoding, as well as adaptive power control; the links are also protected by error control based on retransmission which will be outlined in the next section.

In TI-NET, the satellite subnet is used to support bulk data application categories which, because of the large volume of traffic, have been customarily serviced mainly by messenger services as well as leased lines and circuit switched networks. The bulk data is handled by earth station switches which do minimal processing but have very high I/O capability. These switches form satellite multi-user packers (SMP's) every 10 msec and thereby transport users'

bulk data to their destinations in one hop over the satellite links.

The TI-NET terrestrial and satellite subnet architectures are integrated in such a way as to provide a reliable and cost effective system. This can be seen from Figure 3.3 which shows an example of two TI-NET terrestrial switches (star nodes) with their associated earth station switches. The Figure shows that under normal conditions (i.e. no failures or outages), bulk data is handled by the earth station switches and transmitted in SMP's over the satellite links, while real time traffic is handled by the terrestrial switches and transmitted in MP's over the terrestrial links. Transmission in the satellite subnet takes place end-to-end in one hop, while in the terrestrial subnet, transmission may take place node-to-node with a maximum of 3 hops. Only bulk data messages longer than 6 characters are transmitted over satellite links, while shorter protocol messages, as well as SMP NAK's and retransmissions are carried over the terrestrial links, which have minimum channel capacities of 112 kb/s. Hence, large transmission delays and storage requirements normally associated with satellite transmission are minimized.

The protocol for SMP error control based on terrestrial SMP retransmission is similar to the implicit ACK-explicit NAK protocol used for synchronous MP's in the terrestrial TI-NET; however, SMP retransmissions are carried in an end-to-end circuit switched mode (rather than a node-to-node

FIGURE 3.3 INTEGRATED TERRESTRIAL AND SATELLITE SUBNET ARCHITECTURES:
NORMAL OPERATION

store and forward mode) over a maximum of 3 terrestrial links. To further improve error performance, NAK's are also generated as a result of the satellite echoing its received signals back to the transmitting stations.

When a terrestrial link fails the MP's of real time traffic generated in terrestrial switches at the ends of this link are directed to the earth station switch and sent over the satellite link. On the other hand, if an outage occurs due to fading on the satellite link, high priority bulk data is carried by the terrestrial network in an end-to-end circuit switched mode over a maximum of 3 links.

3.5 Satellite Extension Nodes

The integrated terrestrial and satellite TI-NET system would have its nodes located in major cities where there is a very high concentration of data. However, there will be many other similar regions both local to, and remote from these nodes. In TI-NET these regions can be accommodated by establishing extension nodes which use 12-14 GHz satellite transmission, to avoid interference from existing 4-6 GHz microwave facilities. This can be done by identifying regions where considerable traffic is generated and putting up inexpensive earth stations, i.e., extension nodes, in these locations as illustrated in Figure 3.4. An extension node would generate "satellite multi-user packets" every 10 ms's and transmit real time, bulk data and voice traffic to the

FIGURE 3.4 SATELLITE EXTENSION NODE

most appropriate TI-NET entrance node; this should be the destination for the bulk data so that it is transmitted in one hop (250 ms. propagation delay). The real time and voice traffic will start from the entrance node and reach their destinations in the usual way over the terrestrial links of TI-NET. TDMA is used as the means of accessing the TI-NET node from the extension node and vice-versa.

There are potential advantages of accessing TI-NET by satellite extension nodes as compared to accessing conventional packet switched networks using terrestrial facilities. For a given geographical location and volume of traffic there is a distance beyond which the use of an extension node is more economical than the use of local loop and long haul facilities to reach the nearest TI-NET node. Also, when comparing TI-NET extension nodes to conventional networks, the 250 ms propagation delay compares favourably with the entrance delay incurred in an interface processor in order to form packets. For example, it takes 333 ms to form a 50 character packet from a 1.2 kb/s polled synchronous terminal.

Figure 3.4 also shows a terrestrial link joining the extension node to the TI-NET entrance node with which it is possible to improve error performance by using the link to carry SMP NAK's and retransmissions. The speed of this link

(e.g. ≤ 9.6 kb/s) would be much less than the satellite link capacity, but high enough to ensure that most retransmissions would be received within an acceptable time.

3.6 Summary of TI-NET Characteristics

The packet structure used in TI-NET is based on public "multi-user" packets containing user "minipackets", and is responsible for providing small entrance delay into TI-NET of 10-20 ms. By comparison, access to a conventional packet switched network such as ARPA, for a 2.4 kb/s terminal for example, involves the formation of 500 or 1000 bit packets with corresponding entrance delays of over 200 and 400 ms respectively. Shorter packets may be used at the expense of increased overhead. A packetized virtual circuit approach to solving this problem was explained in Chapter 2.

The TI-NET terrestrial topology is structured such that any node can be reached from any other with a maximum of 3 hops. In addition, the proposed flow control scheme is based on reservations, and is intended - in general - to reduce steady state synchronous queue sizes and waiting times within the subnet. In ARPANET, queues are permitted to exist within the network and transit delays depend, among other things, on network utilization and the number of hops traversed (since ARPA is k-connected, the maximum number of hops can be much greater than TI-NET's 3 hop maximum). However, an

average transit delay reported appears to be of the order of 250 ms with variations reaching from less than 200 ms to 350 ms and higher [23-25].

Admission delays incurred in waiting for reservations before entering TI-NET may be controlled by giving priority to synchronous traffic over asynchronous traffic, and by traffic engineering. Reservations are obtained on a message basis, and therefore different admission delays could be incurred for successive encoded voice messages. The effects of variable admission delays for encoded voice transmission can therefore be reduced by the use of a "hangover" after each active period in speech to ensure that once a reservation is secured, it is not prematurely released. Admission delays are also effectively incurred in ARPANET since certain resources (e.g. destination buffer space for multi-packet messages) must be reserved before message transmission begins. In addition however, ~~queues exist within the network, and are~~ to a large extent responsible for the large transit delay variations mentioned previously.

In TI-NET, the intention is to transport messages over fixed routes rather than use adaptive or alternate routing as in conventional packet switching. This transportation mechanism avoids packet sequencing problems and message reassembly delays. Also, the synchronization of MP transmission and processing ensures continuity of a message as

it is transported through the network. By comparison, the conventional packet switching used in ARPANET can result in message reassembly delays exceeding 1 sec. due to late and lost packets.

Packetized voice transmission over the satellite links associated with TI-NET extension nodes, and the satellite subnet is suggested by the intended transparent nature of the terrestrial network. Thus even though the satellite links introduce a fairly large fixed component of delay (propagation), the fixed and variable components of entrance and other terrestrial delays are expected to be relatively small. On the other hand packetized voice transmission through a conventional packet switched network with satellite links, would not only involve satellite propagation delay but also fairly large fixed and variable entrance and other terrestrial delays.

CHAPTER 4

AN ANALYSIS OF TI-NET DELAYS

4.1 Summary of TI-NET Delays

The end-to-end delay incurred by a synchronous message traversing TI-NET may be separated into two components [4,46,56]:

- i) Admission delays: the delays incurred on the periphery of the subnet (at the entrance nodes) before packetization and transportation of the message begins.
- ii) Internal delays: the delays incurred within the subnet as the message is processed and transmitted toward its destination.

By way of introducing the more involved admission delay components, internal delays are described first, in conjunction with Figures 4.1 and 4.2. Figure 4.1 shows a typical situation in which an entrance star node is connected to an exit star node through (a maximum of) three TI-NET links. Transmission is full-duplex, but only one direction is shown for illustrative purposes.

Figure 4.2 demonstrates how a synchronous message is processed and transported from the entrance node input to the exit node output; also shown are the various internal delays encountered along the route. - Figure 4.2(a) represents the 10 ms starnet synchronous service times, and provides a time reference for the remaining parts of Figure 4.2; synchronous

FIGURE 4.1 EXAMPLE OF 3 HOP TRAFFIC IN TI-NET

MP transmissions to and from all star nodes begins in accordance with these service times.

Figure 4.2(b) shows a synchronous message, which appears at the entrance node at an arbitrary time with respect to the node service times. As the message arrives it is stored in the entrance node input buffer until the first service time at which at least an equivalent of one mp is present in the buffer (e.g. ≥ 24 bits for a 2.4 kb/s message), as in Figure 4.2(c). At this point (ignoring admission delays for this discussion) servicing of the message begins; that is, the appropriate amount of data is removed from the input buffer, an mp is formed and is multiplexed with other mp's to form a synchronous MP for transmission to central node 1, as in Figure 4.2(d). In this example the packet formation, or entrance delay, incurred by the message before transmission begins is (an average of) 15 ms; however in general this delay may vary between one and two service times (10 to 20 ms) depending on message arrival times with respect to node servicing times.

Figures 4.2(e), (g) and (i) simply represent MP arrival times at central nodes 1 and 2 and the exit node after (maximum permissible) propagation and transmission delays of 2.5, 15 and 2.5 ms respectively.

Since MP's travel between any two nodes in a store and forward manner, a 5 ms delay (equivalent to the MP length)

is incurred to service (receive) an MP at each of these nodes (recall from the TI-NET overview in Chapter 3 that two synchronous MP's are transmitted in double redundancy within 5 ms on central links, since the transmission speed is (typically) twice that of star links).

After an MP is received it is immediately demultiplexed, and at central nodes 1 and 2, new MP's are formed for transmission to the next node. Transmission takes place immediately from central node 1 - Figure 4.2(f) - but there is a 2.5 ms alignment delay at central node 2 so that transmission to the star exit node is in agreement with the synchronous service times - Figure 4.2(h).

After MP demultiplexing at the exit node, data in the mp for the message in question is immediately put into the exit node output buffer - Figure 4.2(j). Finally, the explicit NAK - implicit ACK error control scheme for synchronous (data) messages, requires the equivalent of four mp's in the output buffer during message delivery to the destination; thus in the event of one or more retransmissions, the buffer will not be depleted thereby preserving message continuity. This results in a 40 ms delay as illustrated in Figure 4.2(k). Since transmission errors are not nearly so critical for encoded voice as for data traffic, error control can be considered optional for voice in which case the 40 ms delay would not be encountered.

The internal delays are summarized as follows:

Packet formation (entrance)	15 ± 5 ms
Store and forward (3 hops)	15
Central/star node alignment	2.5
Error control (optional for voice)	40
	72.5 ± 5
Processing (not shown in Figure 4.2)	2.5
	75 ± 5
Propagation (maximum)	20
	95 ± 5

4.2 Aspects of Parallel Servicing

4.2.1 Parallel Servicing Vs a Single Server System

For any queueing system, one of the important parameters which characterizes the system, is the mean time a message spends waiting in a queue before servicing begins (T_q); other parameters are the mean time to service the message (T_s) and the total mean time spent in the system ($T_t = T_q + T_s$) [112].

At this point it is emphasized that with a transparent approach, apart from a relatively small entrance delay (e.g. 10-20 ms in TI-NET), the mean time to service a message (T_s) corresponds essentially to the mean time to transmit the message on a leased line. (As illustrated in the next section, this is done by assigning the required number of parallel servers - analogous to leased lines - to the messages, and then transmitting the messages in parallel "on the fly" as they arrive). Thus, since it is only meaningful and interesting to consider those components of T_t which are over and above those that occur on

a leased line, the particular component T_s is of no consequence when considering a transparent approach.

For a finite number of servers however, queueing will take place and result in a variable component of admission delay, which is important from the point of view of voice transmission [48,56]. For the purposes of the work described here, the useful characteristic of a parallel server system is that the time spent in the queue is less than for a single server system for the same utilization and with the same total output service rate. This is illustrated in Figures 4.3 and 4.4 in which the normalized mean and standard deviation for the queueing time are compared for the M/M/1 and M/M/m systems (with homogeneous populations), for various utilizations. In both Figures, the total output service rate is $\mu = C/l$ (msg's/sec), where C is the total output capacity (char/sec) and l is the mean message length (char/msg). Therefore, the capacity per server in the M/M/m system is $R = C/m$ (char/sec). It is also assumed that each terminal in the population has transmission speed R ; apart from non-homogeneity, this is analogous to TI-NET.

4.2.2 Physical Description of TI-NET Entrance and Exit Node Operation

Figure 4.5 is a diagram of the TI-NET terrestrial subnet in which an entrance node is shown with synchronous terminals communicating with corresponding terminals at an exit node. The subnet is assumed to be operating in the "permanent circuit" mode, and the terminals are assumed to be signed on. (By permanent circuit mode, it is meant that any user at a source node always communicates with another user at the same

FIGURE 4.3

NORMALIZED MEAN TIME SPENT WAITING IN
THE QUEUE FOR M/M/m SYSTEM

FIGURE 4.4

NORMALIZED STANDARD DEVIATION OF TIME
SPENT WAITING IN THE QUEUE FOR
M/M/m SYSTEM

FIGURE 4.5 TI-NET ENTRANCE AND EXIT NODE OPERATION

destination node, over what is analogous to a private line between the two users). Although end-to-end transmission is full duplex, only one direction of transmission is shown for convenience. Note that there are four types of synchronous terminals considered (1.2, 2.4, 4.8 and 9.6 kb/s) and that there are a number (M_1 , M_2 , M_3 , and M_4) of terminals of each type. In general therefore, the population is non-homogeneous and finite.

At the entrance node, messages from the terminals are loaded into the input buffers (e.g., one character at a time on an interrupt basis). All input buffers are then serviced by the node every 10 msec. In TI-NET an mp corresponds to the amount of data that a terminal can load into an input buffer during the 10 msec service interval. For 1.2, 2.4, 4.8 and 9.6 kb/s terminals the mp sizes are therefore 12, 24, 48 and 96 bits respectively. A message is said to have arrived, at the first service time at which there is more than one mp of data in the input buffer. The first mp is then removed from the input buffer, and if sufficient channel capacity and reservations (resources) are available, is put into an MP for transmission; similarly all subsequent mp's for that message are removed one at a time every 10 msec and put into subsequent MP's. If there are insufficient resources, the mp's are put into the entrance queue until sufficient resources become available. A message is said to be queued at the entrance as soon as the first mp is put into the

entrance queue, and to have departed from the queue (been de-queued) when the first mp is taken from the queue and put into an MP for transmission. All of the input buffers are serviced in parallel every 10 msec in the manner described above.

The mp's from the various messages may be "packed" into MP's for transmission according to a "selective" FCFS (SFCFS) discipline [56,113]. I.e., consider the example MP in Figure 4.6, in which the MP data field is shown subdivided into a number of 12 bit compartments, each of which correspond to 1.2 kb/s of channel capacity on the output link leaving the entrance node. One may therefore think of a TI-NET node as consisting of a number of parallel 1.2 kb/s servers. The example MP is shown to be "packed" with mp's (or equivalently-servicing messages) from a number of terminals of each type. (Note, that the order in which the mp's are shown packed is only a convenient example, and that the actual packing order is determined by "activity fields" [4], which were mentioned in the TI-NET overview in Chapter 3). If one imagines that this MP is of maximum length then it is seen to be packed in the fortunate way that all 1.2 kb/s servers are utilized.

However, because of the non-homogeneity of the terminal population, it is quite possible in some cases for newly arrived messages not to be able to use the servers available and to have to wait in the entrance queue. For example,

FIGURE 4.6 EXAMPLE OF A TI-NET MULTI-USER PACKET-MP CONTAINING MINIPACKETS - mp's

suppose that service on one of the 2.4 kb/s messages is completed and it departs from the system thus freeing two servers. If a 4.8 kb/s message then arrives, it will have to queue, since four servers are required to service it, whereas only two are available. The objective of the SFCFS discipline therefore, is to select messages for service not only on an FCFS basis, but also in such a way as to reduce the number of unused 1.2 kb/s servers. For example, suppose that at a particular service time there are 5 free servers and that the following messages are in the entrance queue: 2.4, 1.2, 4.8, 1.2, 2.4, and that they arrived in the order shown. Servicing on a strictly FCFS basis would mean selecting only the first 2.4 and 1.2 messages (provided reservations are available), thus leaving 2 servers unused; however, on an SFCFS basis the 4.8 would be by-passed and the following 1.2 would also be selected (provided a reservation was available), thus leaving only 1 unused server. Furthermore, any message which has been by-passed is given priority over other waiting messages at the next service time.

It is evident that there are other (more complex) disciplines which would reduce the number of unused servers even more (e.g., in the example, by-pass the first 2.4 and select the following 1.2 and 4.8 messages, thus leaving no unused servers). However, in general although such disciplines will likely reduce the average admission delay, they may also increase the delay variance.

At the exit node, received MP's are unpacked every 10 msec and the contents of the mp's are put into their appropriate output buffers and delivered to the terminals.

4.3 Admission Delays for Single-Hop Synchronous Traffic

4.3.1 Non-Homogeneous Job Population in an M/M/ ∞ System [56]

4.3.1.1 Motivation and Assumptions

At this point a start is made in the analysis of admission delays for single-hop synchronous traffic in TI-NET. From previous discussions it is evident that a TI-NET node is similar in many respects to an M/M/m system, where each of the m servers has 1.2 kb/s channel capacity. However, a non-homogeneous population brings about complications by reducing the efficiency with which the servers are utilized (i.e., MP packing efficiency).

Some of these complications disappear when an infinite number of servers is considered, since service can always begin on messages as soon as they arrive; i.e., there is no queueing and hence packing efficiency becomes a non-issue. Some analytical results have been obtained for this M/M/ ∞ case, and are representative of TI-NET node operation under light load conditions.

An M/M/ ∞ system with non-homogeneous jobs may be modeled in two different ways. The first uses a number of parallel service facilities - one for each job type. This model will be described first since it is more readily

interpreted in terms of the physical system (TI-NET node), and is easier to analyse. The second model uses a single (combined) service facility; it is somewhat more abstract and will be presented later.

In what follows (and in what has been said already), the terms 'job', 'bulk' and 'message' are synonymous and will be (have been) used interchangeably; furthermore, the term 'bulk size' does not refer to message length (see items 4 and 6 below). The following assumptions are made with regard to the model:

1. Infinite job population.
2. Four types of jobs: 1.2, 2.4, 4.8 and 9.6 kb/s.
3. Bulk Poisson arrivals.

4. Bulk size at each arrival instant is random:

i.e., $g_i \triangleq P(\text{bulk size is } i)$,

where $i \in S = (1, 2, 4, 8)$, and g_i represents the traffic balance; i.e., the proportion of messages arriving that are of type i .

(a bulk of type or size i consists of i bulk units; each bulk unit corresponds to 1.2 kb/s or 150 char/sec).

5. The total bulk arrival rate or throughput is:

$$\lambda = \sum_{i \in S} \lambda g_i = \lambda \sum_{i \in S} g_i = \lambda \text{ msg's/sec}$$

where λg_i is the fractional throughput for messages of type i .

6. Each of the infinite identical servers has transmission Rate $R = 1.2 \text{ kb/s}$ or 150 char/sec ; i such servers are required to service a type i message, i.e., one server for each of the i bulk units comprising the message.

7. Message lengths for the four different job types are distributed independently. Furthermore, each length has an exponential distribution with mean message length ℓ char/msg. Thus the service rate per server is $\mu = R/\ell = 150/\ell$, and the service rate per message is $i\mu = iR/\ell$ msg's/sec.
8. System behavior under equilibrium conditions is being sought.

4.3.1.2 System Representation Using Parallel Service Facilities

The system to be analysed is shown in Figure 4.7. Each facility draws as many servers as it requires from an infinite pool. These servers service all the messages currently in that facility. For the i th facility, each server provides an average service rate of $\mu = R/\ell$ msg's/sec, where R and ℓ have been previously defined.

Let the state of the i th facility be the number of messages k_i in, (or equivalently being served by) that facility. Each message of type i requires i servers and therefore an average service rate of $i\mu$ msg's/sec. If there are k_i such messages in facility i , then the net service rate for that facility is state dependent and given by $i\mu k_i$ msg's/sec. The rate transition diagram for facility i is shown in Figure 4.8. The density for the number of messages in this facility is Poisson (as in a conventional M/M/ ∞ system); i.e.,

$$P_{k_i} = \frac{r_i^{k_i}}{k_i!} e^{-r_i}, \quad (4.1)$$

FIGURE 4.7 NON-HOMOGENEOUS JOBS IN AN M/M/ ∞ SYSTEM:
REPRESENTATION USING 4 PARALLEL SERVICE FACILITIES

FIGURE 4.8 RATE TRANSITION DIAGRAM FOR THE i th FACILITY IN AN $M/M/\infty$ SYSTEM WITH NON-HOMOGENEOUS POPULATION.

with mean and standard deviation being respectively:

$$r_i = \frac{\lambda g_i}{i\mu} \quad \text{and} \quad \sigma_i = \sqrt{\frac{r_i}{i}} \quad (4.2)$$

Since the average number of messages in facility i is r_i , the net average service rate for that facility is $i\mu r_i$ msg's/sec. which equals the average arrival rate (λg_i).

Consider again, the entire system in Figure 4.7 which consists of the four independent parallel service facilities. In general, the total number of messages and bulk units in this system are respectively:

$$\sum_{i \in S} k_i \quad \text{and} \quad \sum_{i \in S} i k_i \triangleq k \quad (4.3)$$

Similarly, the total average number of messages and bulk units in the system are respectively:

$$\sum_{i \in S} r_i \quad \text{and} \quad \sum_{i \in S} i r_i \triangleq r \quad (4.4)$$

Since messages of every type can be expressed in terms of bulk units:

$$\text{i.e., } k_i \text{ messages} = i k_i \text{ bulk units} \quad (4.5)$$

then it is convenient to let the state of the entire system be the number of bulk units (k) in the system.

At this point there are two different but equivalent approaches that may be taken to determine the probability P_k of having k bulk units in the system. The first approach is intuitive and algorithmic in nature, and perhaps provides

a more thorough understanding of the problem, by exposing its combinatorial aspects, while the second approach is more classical. The two approaches will be treated separately and in the order indicated above.

(a) First Approach to Determining the pdf for the Number of Bulk Units in the System

The four independent Poisson densities for the number of messages in each facility - see (4.1) - may be used to determine the probability of k bulk units in the system as follows:

Let A be the event that there are k bulk units in the system, and define $P_k \triangleq P(A)$. Also, let B_j be the event that there are k_1, k_2, k_4 and k_8 messages in the four facilities respectively; i.e., B_j represents the j th way of having k bulk units in the system. Now since the P_{k_i} 's in (4.1) are independent:

$$\begin{aligned} P(A, B_j) &= P_{k_1} \cdot P_{k_2} \cdot P_{k_4} \cdot P_{k_8} \\ &= \left(\frac{r_1^{k_1}}{k_1!} e^{-r_1} \right) \left(\frac{r_2^{k_2}}{k_2!} e^{-r_2} \right) \left(\frac{r_4^{k_4}}{k_4!} e^{-r_4} \right) \left(\frac{r_8^{k_8}}{k_8!} e^{-r_8} \right) \\ &= \frac{r_1^{k_1}}{k_1!} \frac{r_2^{k_2}}{k_2!} \frac{r_4^{k_4}}{k_4!} \frac{r_8^{k_8}}{k_8!} e^{-\sum_{i \in S} r_i} \end{aligned} \quad (4.6)$$

In general for some fixed value of k , there are many different values of k_1, k_2, k_4 and k_8 such that $k = \sum_{i \in S} k_i$. Therefore,

$$P_k = P(A) = \sum_{j=1}^L P(A, B_j) \quad (4.7)$$

where the sum covers all of the L ways of having messages in the facilities such that the equality in (4.3) holds.

Some illustrative examples follow:

$$\begin{aligned}
 k = 0 \quad P_0 &= e^{-\sum r_i} \\
 k = 1 \quad P_1 &= r_1 e^{-\sum r_i} = r_1 P_0 \\
 k = 2 \quad P_2 &= \left(\frac{r_1^2}{2} + r_2 \right) P_0 \\
 k = 3 \quad P_3 &= \left(\frac{r_1^3}{3!} + r_1 r_2 \right) P_0 \\
 k = 4 \quad P_4 &= \left(\frac{r_1^4}{4!} + \frac{r_1^2}{2} r_2 + \frac{r_2^2}{2} + r_4 \right) P_0 \\
 &\text{etc.}
 \end{aligned} \tag{4.8}$$

where $P_0 = P(\text{system is empty})$.

For a particular k , the terms comprising P_k are interpreted as follows: e.g., for $k = 4$, the first term $P_0 r_1^4 / 4!$, is the probability that there are 4 type 1 (1.2 kb/s) messages in the first facility. The second term $P_0 r_1^2 r_2 / 2$, is the probability of 2 type 1 messages in the first facility, and 1 type 2 (2.4 kb/s) message in the second facility, etc. Thus in general P_k is a finite sum of the probabilities of each way of having k bulk units in

the system. This finite sum is analogous to the expansion of the pdf for the sum of homogeneous Poisson random variables. For example, the pdf for the sum of two homogeneous Poisson random variables with means "a" and "b" is:

$$P_k = \frac{(a+b)^k}{k!} P_0$$

where $P_0 = e^{-(a+b)}$;

expanding for $k = 3$ gives

$$P_3 = P_0 \left(\frac{a^3}{3!} + \frac{a^2}{2}b + a\frac{b^2}{2} + \frac{b^3}{3!} \right)$$

For the non-homogeneous case of interest here, however, it appears that the finite sum is not factorable. Thus to find P_k in general using the approach leading to (4.7) and (4.8), it is necessary to find all ways of having k bulk units in the system. An algorithm has been devised for this purpose and is described in Appendix A.2; also discussed there are various other combinatorial aspects of this problem.

Thus using equations (4.6) and (4.7) and the algorithm just mentioned, it is possible to compute P_k as illustrated by the first few terms in (4.8). Various complete examples are plotted and discussed in detail after the next section.

(b) Second Approach to Determining the pdf for the Number of Bulk Units in the System

This approach makes explicit use of the fact that in general the number of bulk units in the system is given in

(4.3). Also, using (4.5), the probability P_{k_i} of k_i messages in facility i is equivalent to the probability of ik_i bulk units in facility i . Therefore, P_k is the convolution of the four independent Poisson P_{k_i} 's given in (4.1),

$$\text{i.e., } P_k = P_{k_1} * P_{k_2} * P_{k_4} * P_{k_8} \quad (4.9)$$

For this purpose, the P_{k_i} 's in (4.1) are re-written using (4.5), and are indexed with respect to bulk units;

$$\text{i.e., } P_{k_i} = P_i(k) = \frac{r_i^{k/i}}{(k/i)!} e^{-r_i} \quad (4.10)$$

where $k=0, i, 2i, 3i \dots$ etc. bulk units. For $i=1$ and 2 in (4.10), and noting Figure 4.9, the convolution is as follows:

$$\begin{aligned} P_1(k) * P_2(k) &= \sum_{n=0}^k P_1(n) P_2(k-n) \\ &= e^{-(r_1+r_2)} \sum_{n=0}^k \frac{r_1^n}{n!} \frac{r_2^{[(k-n)/2]}}{[(k-n)/2]!} \end{aligned}$$

where n is even when k is even, and n is odd when k is odd.

Let $y = (k-n)/2$, then $n = k-2y$, and

$$P_1(k) * P_2(k) = e^{-(r_1+r_2)} \sum_{y=0}^{\left\lfloor \frac{k}{2} \right\rfloor} \frac{r_1^{k-2y}}{(k-2y)!} \frac{r_2^y}{y!},$$

where the notation $\lfloor x \rfloor$ means: x if x is an integer, or next smallest integer of x if x is not an integer.

FIGURE 4.9 ILLUSTRATION FOR THE
CONVOLUTION $P_1(k) * P_2(k)$

Similarly the remaining two convolutions are:

$$P_1(k) * P_2(k) * P_4(k) = e^{-(r_1+r_2+r_3)} \sum_{y=0}^{\left\lfloor \frac{k}{4} \right\rfloor} \sum_{m=0}^{\left\lfloor \frac{k-4y}{2} \right\rfloor} \frac{r_1^{k-4y-2m}}{(k-4y-2m)!} \frac{r_2^m}{m!} \frac{r_4^y}{y!}$$

and finally noting (4.9) and (4.10)

$$P_k = P_0 \sum_{y=0}^{\left\lfloor \frac{k}{8} \right\rfloor} \sum_{m=0}^{\left\lfloor \frac{k-8y}{4} \right\rfloor} \sum_{n=0}^{\left\lfloor \frac{k-8y-4m}{2} \right\rfloor} \frac{r_1^{k-8y-4m-2n}}{(k-8y-4m-2n)!} \frac{r_2^n}{n!} \frac{r_4^m}{m!} \frac{r_8^y}{y!} \quad (4.11)$$

$$\text{where } P_0 = e^{-\sum_{i \in S} r_i} \quad (4.12)$$

$k = 0, 1, 2, 3, \dots$ bulk units

and $y, m,$ and $n = 0, 1, 2, 3, \dots$ up to their respective limits.

Equation (4.11) is in the form of a finite sum, and reflects the complications introduced when convolving four Poisson densities which not only have different parameters (r_i) but also have different bulk sizes (are non-homogeneous). When (4.11) is expanded for any k (a non-negative integer), it displays the probabilities for all the ways of having k bulk units in the system. These expansions are identical to those illustrated by example in (4.8).

P_k in (4.11) is only a function of the r_i 's, i.e. the mean number of messages in each facility. From (4.2) each r_i depends on the fractional throughput (λg_i) and service

rate $\lambda(i)$; however, this dependence is not unique since there are many values of λg_i and i which result in the same r_i . Three examples of (4.11) are plotted in Figures 4.10(a), (b) and (c) for various proportions of r_i . In order that these figures may have a common point of reference, the total mean number of bulk units - see (4.4) - was made the same (and equal to 22.4) for each. Figure 4.10(a) is for equal r_i 's which results in progressively more bulk unit in the system for messages of types 2, 4 and 8 than for type 1. This also accounts for the somewhat "jagged" appearance of the envelope of P_k ; i.e., the probabilities for odd numbers of bulk units in the system are somewhat suppressed because this specifically requires the presence of the relatively rare bulk units from messages of type 1. This "jagged envelope" phenomenon is not present in Figure 4.10(b) in which the r_i 's have been made unequal in favour of type 1 messages; in this example, the mean number of bulk units are the same for each message type. The total mean and standard deviation for the number of messages in the system has now increased since with this proportion of r_i 's, more messages are required to have the same total mean number of bulk units. However, the total standard deviation of bulk units is less since type 8 messages are no longer dominant, as they were in Figure 4.10(a). Finally in Figure 4.10(c), both the r_i 's and the mean number of bulk units per message were made unequal in favour of type 8 messages. In this example, the presence of bulk units from type 1 messages

FIGURE 4.10(a) pdf FOR AN M/M/ ∞ SYSTEM WITH NON-HOMOGENEOUS POPULATION

FIGURE 4.10(b) pdf FOR AN M/M/∞ SYSTEM WITH
NON-HOMOGENEOUS POPULATION

FIGURE 4.10(c) pdf FOR AN M/M/1 SYSTEM WITH NON-HOMOGENEOUS POPULATION

is less likely than in Figure 4.10(a) and this amplifies the jaggedness of the pdf envelope. Of course in the limit as r_1 approaches zero, P_k for k odd also approaches zero.

(c) Comments and Observations

Another parameter tabulated in Figure 4.10 is δ_i . The physical significance of this parameter may be seen by referring again to Figure 4.7. In this regard, a question worth asking is: of the total numbers of servers being utilized at some instant, what proportion is being used by jobs of type i ? Since the number of messages being served is just the number in the system, then the instantaneous total number of servers used is given by (4.3) and the proportion being sought is:

$$\delta_i(k) = \frac{ik_i}{k} \quad (4.13)$$

Therefore, using a similar argument and (4.4), the average proportion of servers used by jobs of type i is:

$$\delta_i = \frac{ir_i}{r} \quad (4.14)$$

Further physical significance may be attributed to the parameter δ_i by first recognizing that for the m server case, the fractional utilization (F.U.) of the system for type i messages is the ratio of the average arrival rate to the maximum service rate for such messages:

$$\text{i.e., } F.U. = \lambda g_i / m\mu = ir_i / m \quad (4.15)$$

where (4.2) has been used, and r_i is now the mean number of type i messages in service. Now using (4.4), the total utilization factor (U.F.) for the system is:

$$U.F. = \sum_{i \in S} F.U. = \frac{r}{m}, \quad (4.16)$$

where r is now the total mean number of bulk units in service.

Finally, from (4.14) the ratios of fractional to total utilizations are:

$$\frac{F.U.}{U.F.} = \frac{ir_i / m}{r / m} = \delta_i \quad (4.17)$$

Since the term "utilization" generally refers to the extent to which a system is "loaded", the parameters δ_i may be considered to represent the system load balance. For example in Figure 4.10(b) where $\delta_i = 1/4$ for all message types, the load is balanced, whereas in each of Figures 4.10(a) and (c) the load is unbalanced.

A further observation with regard to (4.16) is that for any U.F., it is possible to calculate $m = r/U.F.$, and sum (4.11) from m to ∞ to obtain an optimistic estimate for the probability of queueing for the M/M/m system with non-homogeneous jobs.

Another significant question with respect to Figure 4.7 is: of the total instantaneous service rate, what proportion is applicable to jobs of type i ? From the figure, it is

evident that this proportion is:

$$\gamma_i(k) = \frac{i\mu k_i}{\sum_{i \in S} i\mu k_i} = \frac{ik_i}{\sum_{i \in S} ik_i} = \frac{ik_i}{k} \quad (4.18)$$

Similarly on the average and using (4.2) and (4.4),

$$\gamma_i = \frac{i\mu r_i}{\sum_{i \in S} i\mu r_i} = \frac{\lambda g_i}{\sum_{i \in S} \lambda g_i} = g_i \quad (4.19)$$

In other words, the probability that a departure is of type i , equals the probability that an arrival is of type i . In the case considered here where all job types have identical mean message lengths, it is seen from equations (4.4), (4.14) and (4.19) that the load balance δ_i equals the traffic balance g_i (or γ_i).

The mean time spent in the system by a type i message is:

$$T_{s_i} = \frac{1}{i\mu} \quad (\text{sec}) \quad (4.20)$$

which is just the mean time in service since no queueing is involved. The mean time averaged over all messages is from Little's Result [114], the total mean number of messages in the system divided by the total throughput λ . Therefore from (4.2) and (4.20)

$$T_s = \frac{\sum_{i \in S} r_i}{\lambda} = \sum_{i \in S} g_i T_{s_i} \quad (4.21)$$

which is the sum of the service times for individual message types weighted by the traffic balance parameter.

Although a solution for an M/M/m system with non-homogeneous jobs is not provided here, some useful observations can be made. The first is that if messages of type i strongly dominate over all other message types, (i.e. traffic balance g_i close to 1), then the behavior of an M/M/m system with non-homogeneous jobs can be expected to approach that of the same system with homogeneous jobs, and m/i servers each having capacity iR kb/s. For example, if $m = 32$ servers each having capacity $R = 1.2$ kb/s, and 2.4 kb/s messages are dominant (g_2 nearly 1), then behavior will approach that of an M/16 system with each server having 2.4 kb/s capacity. Figure 4.3 (discussed previously) shows the mean time spent in the queue for 4, 8, 16 and 32 servers, which correspond to dominance of 9.6, 4.8, 2.4 and 1.2 kb/s messages respectively. Furthermore, for an arbitrary mix of message types, the behavior can be expected to lie somewhere above the curve for 32 servers in Figure 4.3 (and 4.4). However, because of packing inefficiency it is not clear that behavior will always lie (on or) below the curve for 4 servers.

An estimate as to how poor performance may become, is indicated by the fact that for this example the maximum packing inefficiency is $(7/32) \times 100 \approx 22\%$; i.e. a type 8 (9.6 kb/s) message arrives and there are only 7 free servers, and so this message cannot be served. Therefore, at that instant, it may be considered that the system is only

operating with $32 - 7 = 25$ servers. Thus for dominating 9.6 kb/s messages and worst case packing (7 unused servers), the system could instantaneously behave as poorly as an M/M/m system with $25/8 \approx 3$ servers, each having 9.6 kb/s capacity. Thus on the average for an arbitrary mix of message types, behavior can be expected to lie somewhere between curves for 32 and 3 servers in Figure 4.3 (and 4.4).

Before ending this discussion, a useful point is made with regard to F.U. and U.F.. Thus far these parameters have been defined in terms of the maximum number of (1.2 kb/s) servers (m). However, for a non-homogeneous population it may not be possible for all message types to make use of all servers. For example, if $m = 30$ servers, the type 4 and type 8 messages can only potentially each use as many as $4 \times 7 = 28$ and $8 \times 3 = 24$ servers, respectively. Hence the F.U. for these message types has effectively increased by the ratios $30/28$ and $30/24$ respectively, which means that the system is actually more heavily loaded than if all servers could be used. This suggests that m should always be a multiple of 8 for the system described here.

4.3.1.3 System Representation Using a Single (Combined) Service Facility

The combined service facility is shown in Figure 4.11, and since it holds all messages of every type in the system, there are again $k = \sum_{i \in S} ik_i$ bulk units in the system. Recalling (4.5), it would again be desirable to let the system state be the number of bulk units (k) in the system.

FIGURE 4.11 NON-HOMOGENEOUS JOBS IN AN M/M/ ∞ SYSTEM:
REPRESENTATION USING A SINGLE SERVICE FACILITY

However, it will be seen that in order to retain the Markov property, the state description cannot be that simple.

Each bulk unit in the system requires one server with service rate $\mu = R/\ell$ msg's/sec; therefore, the total service rate is $\sum_{i \in S} i\mu k_i$ msg's/sec (as in Figure 4.7).

Expressed another way, it may be said that when k servers are busy, the probability of any job of any type being completed in time Δt is proportional to $\Delta t \sum_{i \in S} i\mu k_i + o(t)$ (since service is exponential). However, it will be necessary to know the probability of any job of type i being completed in time Δt . This is $\Delta t \sum_{i \in S} i\mu k_i + o(t)$ multiplied by that proportion of the service rate which is applicable to jobs of type i . This proportion however, depends not only on k , but also on the way in which there are k in the system, and is given by (4.18). The probability of any job of type i being completed in time Δt is therefore

$$\frac{i\mu k_i}{\sum_{i \in S} i\mu k_i} \cdot [\Delta t \sum_{i \in S} i\mu k_i + o(t)] = i\mu k_i \Delta t + o(t),$$

and the corresponding service rate is $i\mu k_i$ (as in Figure 4.7).

From the above discussion it is evident that the system behavior from the point of view of departure, depends on both k and on the way in which there are k in the system. Before proceeding further therefore, it is necessary to give a careful state description and ensure that it represents a Markov chain.

In this regard Appendix A.3 illustrates by way of discussion and an example, that the ways of having k bulk units in the system are the elements of a cartesian product in a particular four dimensional discrete state space. It is shown that at any instant the "state" of the system (e.g., $k = 6$) may be any one of a number of substates (ways of having k in the system; i.e., elements of the cartesian product). However, a single step transition between substates is not possible, and in fact results in a complete change of "state" (e.g., from $k = 6$ to $k = 4$). On the other hand, the particular new "state" in which the system lands as a result of a single departure, depends on which substate the system was in before the departure.

It is therefore appropriate to consider a state description based on a vector $[k, B_j(k)]$, where $B_j(k)$ represents the j th substate of k (i.e., the j th way of having k in the system). This state description is Markovian since it contains all the information relevant to the future system development [112]. Hence a rate transition diagram may be drawn which includes all of the substates associated with the probability of having k in the system; an example is given in Appendix A.3.

However, such a diagram cannot be readily generalized since (as shown in Appendix A.1) even for moderate values of k , the number of substates is exceedingly large.

It would therefore be desirable to obtain a rate transition diagram which does not explicitly involve substates. In order to do this it is necessary to combine the dependencies of the departure rates on the substates. This may be done using the probabilistic approach which follows, and for which the assumption still holds, namely that all message types have identical mean message lengths.

Recall (4.6) in which A was the event that there are k bulk units in the system. Also recall that B_j was the event: the j th way of having k in the system, and that there are L such ways ($1 \leq j \leq L$). In addition, let S_i be the instantaneous service rate for type i messages, and let S be the total instantaneous service rate for all messages. Thus it is possible to write the conditional expected service rate

$$E[S_i | A, B_j] = (\mu k_i)_j = \mu (ik_i)_j \quad (4.22)$$

where the notation $(x)_j$ refers to the service rate x for the j th way of having k in the system. Also the assumption of equal mean message lengths means that (4.22) is only dependent on the number of type i bulk units in the system (ik_i) .

Summing (4.22) over all L ways gives

$$E[S_i | A] = \mu \sum_{j=1}^L (ik_i)_j P(B_j | A).$$

But $P(B_j|A) = P(A, B_j)/P(A)$ where $P(A) \triangleq P_k$.

$$\therefore E[S_i|A] = \frac{\mu}{P_k} \sum_{j=1}^L (ik_i)_j P(A, B_j) \triangleq \bar{S}_i(k) \quad (4.23)$$

where $P(A, B_j)$ is given by (4.6) and $\bar{S}_i(k)$ is the average service rate for type i messages given k bulk units in the system.

Equation (4.23) is all that is required to draw a compact state diagram in which substate dependencies have been absorbed in $\bar{S}_i(k)$. However, it is useful to also determine the total average service rate for all messages given k bulk units in the system. Thus

$$\begin{aligned} E[S|A] &= \sum_{i \in S} E[S_i|A] \triangleq \bar{S}(k) \\ &= \frac{\mu}{P_k} \sum_{i \in S} \left[\sum_{j=1}^L (ik_i)_j P(A, B_j) \right] \end{aligned}$$

It can be shown that interchanging the above summations, gives the same expansion for $\bar{S}(k)$, therefore:

$$\bar{S}(k) = \frac{\mu}{P_k} \sum_{j=1}^L \left[\sum_{i \in S} (ik_i)_j P(A, B_j) \right]$$

However for any j the terms $\sum_{i \in S} (ik_i)_j$ are identical and given by (4.3). Therefore

$$\bar{S}(k) = \frac{\mu k}{P_k} \sum_{j=1}^L P(A, B_j)$$

Finally using (4.7) leaves

$$\bar{S}(k) = \mu k . \quad (4.24)$$

This is an expected result, and could have been determined directly from the denominator of (4.18) and using (4.3). The assumption of equal mean message lengths which led to (4.24) means that the total average service, when there are k bulk units in the system, is independent of the substates. From (4.23) and (4.24), the proportion of the total service rate applicable to type i messages is therefore

$$\frac{\bar{S}_i(k)}{\bar{S}(k)} = \frac{1}{k P_k} \sum_{j=1}^L (i k_i)_j P(A, B_j) \triangleq \bar{\gamma}_i(k)$$

using (4.18), leaves

$$\bar{\gamma}_i(k) = \frac{1}{P_k} \sum_{j=1}^L (\gamma_i(k))_j P(A, B_j) . \quad (4.25)$$

Summarizing, $\bar{\gamma}_i(k)$ represents the average proportions of the total service rates from state k , which are applicable to messages of type i , when there are a total of k bulk units in the system. Note that this proportion still depends on k but has been obtained by averaging over all substates of k . Thus these average proportions are the sum of the individual proportions for each substate (refer to Equation (4.18)), weighted by the probabilities of those substates, given k bulk units in the system.

A general rate transition diagram for the combined service facility in Figure 4.11 is shown in Figure 4.12. A simpler example is given in Appendix A.3 in which the service rates

$$\bar{S}_i(k) = \bar{S}(k) \bar{\gamma}_i(k) \quad (4.26)$$

are worked out in detail.

Using Equations (4.25), (4.6) and (4.18), it is shown in Appendix A.3 that for departures from state k to states $k-i$,

$$\bar{\gamma}_i(k) = \frac{ir_i}{k} \frac{P_{k-i}}{P_k} \quad (4.27)$$

and that since a departure must be to one of four possible states

$$\sum_{i \in S} \bar{\gamma}_i(k) = 1 \quad (4.28)$$

Similarly, it is shown that for departures from states $k+i$ to state k ,

$$\bar{\gamma}_i(k+i) = \frac{ir_i}{k+i} \frac{P_k}{P_{k+i}} \quad (4.29)$$

Hence, for the case where the $P(A, B_j)$'s (e.g. in (4.25)), are the products of four independent Poisson pdf's (see (4.6), the $\bar{\gamma}_i$'s can be compactly represented in terms of state probabilities as in (4.27) and (4.29). This provides certain other useful information, as will be pointed out later.

FIGURE 4.12 RATE TRANSITION DIAGRAM FOR SINGLE SERVICE FACILITY

However, with regard to Figure 4.12, the equilibrium global balance equation is

$$(\lambda + \mu k)P_k = \lambda \sum_{i \in S} g_i P_{k-i} + \mu \sum_{i \in S} (k+i) \bar{\gamma}_i(k+i) P_{k+i} \quad (4.30)$$

where on the left hand side it has been recognized that

$$\sum_{i \in S} g_i = 1 \text{ and } \sum_{i \in S} \bar{\gamma}_i(k) = 1.$$

The left hand balance equation is

$$\mu k P_k = \lambda \sum_{i \in S} g_i P_{k-i} \quad (4.31)$$

It is fortunate that (4.31) depends neither on initial conditions nor on the $\bar{\gamma}_i(k)$'s; it may be solved using the z transform method. Using Equations (4.2) and (4.4), gives $r = \lambda/\mu$, and defining the z transform

$$P(z) = \sum_{k=0}^{\infty} P_k z^k;$$

then

$$\begin{aligned} \sum_{k=0}^{\infty} k P_k z^k &= r \sum_{i \in S} g_i \sum_{k=0}^{\infty} P_{k-i} z^k \\ &= \sum_{i \in S} i r_i z^i \sum_{k=0}^{\infty} P_{k-i} z^{k-i} \end{aligned}$$

where $r_i = \lambda g_i / i \mu$ from (4.2).

$$z \frac{dP(z)}{dz} = \sum_{i \in S} i r_i z^i P'(z) ,$$

$$\frac{dP(z)}{P(z)} = \sum_{i \in S} i r_i z^{i-1} dz ,$$

$$\ln P(z) = \sum_{i \in S} r_i z^i + c ,$$

and

$$P(z) = e^{\sum_{i \in S} r_i z^i + c} = e^c e^{\sum_{i \in S} r_i z^i} .$$

At

$$z = 1, \quad P(1) = e^c e^{\sum_{i \in S} r_i} = 1 ,$$

$\therefore$

$$e^c = e^{-\sum_{i \in S} r_i} = P_0 \text{ as in (4.12)} .$$

$\therefore$

$$P(z) = e^{\sum_{i \in S} r_i z^i - \sum_{i \in S} r_i} = e^{-\sum_{i \in S} r_i (1-z^i)}$$

$$= \prod_{i \in S} e^{-r_i (1-z^i)} \quad (4.32)$$

Note that $P(z)$ in (4.32) is expressed as a product of four exponentials, which means that the inverse z transform is the convolution of four independent pdf's. Recall that the second approach used to determining P_k for the parallel facility case, consisted of convolving four independent Poisson pdf's. In order to show that the inverse transform of (4.32) is the same as P_k in (4.11), it is sufficient to show that the product of the z transforms of the independent

Poisson pdf's is equal to $P(z)$.

Using (4.10) and defining the z transform

$$P_i(z) = \sum_{k=0}^{\infty} P_i(k) z^k \quad k=0,1,2,3, \dots$$

gives

$$\begin{aligned} P_i(z) &= \sum_{k=0}^{\infty} \frac{r_i^{k/i}}{(k/i)!} e^{-r_i} z^k \\ &= e^{-r_i} \left[1 + r_i z^i + \frac{r_i^2}{2} z^{2i} + \frac{r_i^3}{3!} z^{3i} + \dots \right] \\ &= e^{-r_i} e^{r_i z^i} \end{aligned}$$

$$\therefore \prod_{i \in S} P_i(z) = \prod_{i \in S} e^{-r_i (1-z^i)} = e^{-\sum_{i \in S} r_i (1-z^i)} = P(z)$$

Hence the inverse transform of (4.32) is given by (4.11), and examples have been presented in Figures 4.10(a), (b) and (c).

Other interesting aspects of the balance equations and the equations for the $\bar{\gamma}_i$'s are discussed in Appendix A.3. For example, it is shown that the $\bar{\gamma}_i$'s satisfy the balance equations but not in an independent manner; in fact the $\bar{\gamma}_i$'s can be obtained directly from the balance equations, in a way which suggests that in addition to the global left and right balance over all message types (inherent in (4.30)), there also exists local left and right balance for each

message type. The general balance equation for each message type is given by

$$(\lambda g_i + \mu k \bar{\gamma}_i(k)) P_k = \lambda g_i P_{k-i} + \mu(k+i) \bar{\gamma}_i(k+i) P_{k+i} \quad (4.33)$$

and the corresponding state transition diagram is shown in Figure 4.13.

Equations (4.27) and (4.29) for the $\bar{\gamma}_i$'s are also useful in themselves since they represent the average proportions of the total service rate applicable to jobs of type i , when there are k bulk units in the system. However, for the special case of equal mean message lengths each $\bar{\gamma}_i$ also represents the average proportion of the total number of bulk units in the system; this in turn is equivalent to the average proportion of occupied servers associated with messages of type i . This was referred to earlier (in connection with the parallel facility model of Figure 4.7 and Equations (4.14) and (4.19)), where it was shown that the traffic balance $\gamma_i = g_i$, equals the load balance δ_i .

An example of $\bar{\gamma}_i(k)$ using (4.27) is plotted in Figure 4.14; the mean number of bulk units for this example is given by (4.4) and is $r = 31.25$. Figure 4.14 shows that for small numbers of bulk units in the system, $k \ll r$, most messages are of type i for i small; (e.g., if $k < 8$ there can be no messages of type 8 in the system). Similarly, for $k \gg r$, most messages are of type i for i large; e.g. note in Appendix A.1 how the number of ways of having k in

FIGURE 4.13 LOCALIZED RATE TRANSITION
DIAGRAM FOR TYPE i JOBS

FIGURE 4.14

AVERAGE PROPORTIONAL STATE DEPENDENT SERVICE RATES
VS
NUMBER OF BULK UNITS IN THE SYSTEM

the system is dominated by the case which includes messages of type 8 for large k . Finally, for average numbers in the system, $k \approx r$, it is seen that average departure rate from the system is approximately equal to the average arrival rate, i.e., $\bar{\gamma}_i(r) \approx g_i$.

*In fact evaluating (4.27) at $k = r$ and using (4.19) gives:

$$\bar{\gamma}_i(r) = \frac{ir_i}{r} \frac{P_{r-i}}{P_r} = g_i \frac{P_{r-i}}{P_r}, \quad (4.34)$$

where P_r is the probability of having the mean of $k = r$ bulk units in the system. Thus the reason why $\bar{\gamma}_i(r)$ approximately equals g_i , is that in the region $k \approx r$, P_r is approximately the same as P_{r-i} , although this also depends on the 'jaggedness' of the pdf's as discussed previously. However, for larger average numbers of bulk units in the system (r), there is a general tendency for the pdf's to be more uniform in the region of the mode so that the $\bar{\gamma}_i(r)$'s tend to approach the g_i 's as r increases. This is illustrated by the example in Figure 4.15 where the g_i 's happen to be equal. In this example $\bar{\gamma}_i(r)$ is within $\pm 10\%$ of g_i for average numbers in the system as low as 30. In another example for unequal g_i 's agreement to within $\pm 5\%$ occurred for r as low as 30; i.e., in that example there were a larger percentage of messages of type 8, which made agreement for $i = 8$ much closer than in Figure 4.15.

The $\bar{\gamma}_i$'s in (4.27) and (4.29) are conditioned on there being k bulk units in the system. This dependence may be eliminated by averaging the $\bar{\gamma}_i$'s over all k 's using P_k . For example with (4.27),

$$\begin{aligned}\bar{\gamma}_i &= \sum_{k=i}^{\infty} \bar{\gamma}_i(k) P_k \\ &= ir_i \sum_{k=i}^{\infty} \frac{1}{k} P_{k-i}\end{aligned}\quad (4.35)$$

Evaluating (4.35) shows that for large average numbers of bulk units in the system (r)

$$\bar{\gamma}_i \rightarrow g_i \quad \text{and that} \quad \sum_{i \in S} \bar{\gamma}_i \rightarrow 1.$$

An example showing $\bar{\gamma}_i$ approach g_i for large r is shown in Figure 4.16 for equal g_i 's. Agreement is poorer than for the $\gamma_i(r)$'s in Figure 4.15 since the $\bar{\gamma}_i(k)$'s are now weighted for all values of k not just evaluated at the 'popular' value $k = r$.

4.3.2 Non-Homogeneous Job Population in an M/M/ ∞ /M System

In this section, the infinite job population assumption is relaxed and a finite population consisting of M terminals is considered. Furthermore, the terminals are such that having transmitted a message, they do not send another until after a duration equal to the time to receive a reply plus a memory less think time. Thus the M/M/ ∞ /M system being

considered is a closed queueing system [115], and is appropriate for such data application categories as inquiry response. A queueing system with voice traffic cannot be judged to be either strictly open or closed. That is, while one person is speaking, more than one message may very well be transmitted from the voice terminal before that person pauses and waits for a reply. However, the fact that conversants do eventually pause and wait for replies, does mean that the system is to some extent self regulated in the manner of a closed system. The analytical results obtained are again, representative of TI-NET node operation under light load conditions.

The parallel facility model described previously (Figure 4.7) is modified slightly as shown in Figure 4.17. I.e., the population is now finite, and the parameter λ now represents the average arrival rate (or inverse think time) for each terminal (in msg's/sec). Also because the system is closed, the arrival rates per job type are state dependent and given by $\lambda (M_i - k_i)$ msg's/sec. The total arrival rate is therefore

$$\sum_{i \in S} \lambda (M_i - k_i), \quad (4.36)$$

and M is the total number of terminals;

$$\text{i.e.,} \quad M = \sum_{i \in S} M_i. \quad (4.37)$$

FIGURE 4.17 NON-HOMOGENEOUS JOBS IN AN $M/M/\infty/M$ SYSTEM: REPRESENTATION USING 4 PARALLEL SERVICE FACILITIES

The rate transition diagram for facility i is given in Figure 4.18. The density for the number of messages in this facility is well known and is

$$P_{k_i} = \frac{r_i^{k_i} \binom{M_i}{k_i}}{(1 + r_i)^{M_i}} \quad 0 \leq k_i \leq M_i \quad (4.38)$$

where

$$r_i = \frac{\lambda}{\mu} \quad \text{and} \quad (4.39)$$

$$\binom{M_i}{k_i} = \frac{M_i!}{k_i! (M_i - k_i)!} \quad (4.40)$$

The mean number of messages in the facility is

$$\bar{N}_i = \frac{M_i r_i}{1 + r_i} \quad (4.41)$$

and it is shown in Appendix A.4 that the standard deviation for the number of messages is

$$\sigma_i = \frac{\sqrt{M_i r_i}}{1 + r_i} \quad (4.42)$$

The average service rate for facility i is $\mu \bar{N}_i$ (msg's/sec), which from (4.39) and (4.41) is seen to equal the average arrival rate $\lambda (M_i - \bar{N}_i)$.

$$0 \leq k_i \leq M_i$$

FIGURE 4.18 RATE TRANSITION DIAGRAM FOR THE i th FACILITY IN AN M/M/ ∞ /M SYSTEM WITH NON-HOMOGENEOUS POPULATION

Finally, the total number of messages and bulk units in the entire system in Figure 4.17 is given by (4.3), whereas the corresponding total averages in the system are respectively

$$\sum_{i \in S} \bar{N}_i \quad \text{and} \quad \sum_{i \in S} i \bar{N}_i \triangleq \bar{N}. \quad (4.43)$$

At this point, the algorithmic approach for finding the pdf for the number of bulk units in the system may be applied to the M/M/ ∞ /M system, by using (4.38) in (4.6) and (4.7), and using the algorithm in Appendix A.1 as outlined previously. Alternatively, (4.9) may be used to convolve the four P_{k_i} 's in (4.38). This convolution is similar to the one which led to (4.11) and is given by:

$$P_k = X \sum_{y=0}^{\left\lfloor \frac{k}{8} \right\rfloor} \sum_{m=0}^{\left\lfloor \frac{k-8y}{4} \right\rfloor} \sum_{n=0}^{\left\lfloor \frac{k-8y-4m}{2} \right\rfloor} \frac{r_1^y}{y! (M_1 - y)!} \cdot \frac{r_2^n}{n! (M_2 - n)!} \cdot \frac{r_4^m}{m! (M_4 - m)!} \cdot \frac{r_8^y}{y! (M_8 - y)!} \quad (4.44)$$

where $X = P_0 \prod_{i \in S} M_i!$

$$P_0 = \prod_{i \in S} \frac{1}{(1+r_i)^{M_i}}$$

$$Y = k - 8y - 4m - 2n$$

$$k = 0, 1, 2, 3, \dots, \sum_{i \in S} i M_i \text{ bulk units}$$

and y, m and $n = 0, 1, 2, 3, \dots$ up to their respective limits.

Thus (4.44) is similar to (4.11) in that it is a finite sum which when expanded for any k (such that $0 \leq k \leq \sum_{i \in S} iM_i$), it displays the probabilities for the ways of having k bulk units in the system. P_k in (4.44) is a function of both the r_i 's and the M_i 's. From (4.39) each r_i depends on the think time per terminal ($1/\lambda$) and the service rate (μ). For the purpose of evaluating (4.44), $\bar{N}_i$'s rather than r_i 's will be specified since the $\bar{N}_i$'s are the mean number of messages in the facilities. This also provides a basis for comparing these results with those of the $M/M/\infty$ system (with non-homogeneous jobs) presented previously. Also in this regard (as with Figure 4.10), the mean number of bulk units in the system is held fixed (at 22.4).

Two examples of (4.44) are plotted in Figure 4.19 for total numbers of terminals M of 30 and 60. Note that although (4.44) is a discrete density, only the envelopes are plotted in Figure 4.19 for convenience in making comparisons. The mean number of messages per facility is the same as in Figure 4.10(b). In general (4.44) approaches the infinite population case as M is increased, and the envelope for the infinite population example in Figure 4.10(b) is included in Figure 4.19 for reference purposes. Similarly other parameters (tabulated in Table 4.1) such as the standard deviations for the numbers of messages and bulk units, and the probabilities of some or all of the facilities being empty also approach those for the infinite population case as M is increased.

FIGURE 4.19 pdf's FOR AN $M/M/\infty/M$ SYSTEM WITH NON-HOMOGENEOUS POPULATION

NOTE: THESE pdf's ARE DISCRETE BUT ONLY THEIR ENVELOPES ARE SHOWN FOR CONVENIENCE IN MAKING COMPARISONS.

TABLE 4.1
PARAMETERS ASSOCIATED WITH FIGURE 4.19

Msg. Type i	M/M/ ∞ /30				M/M/ ∞ /60				M/M/ ∞			
	M _i	σ_i	i σ_i	Prob. Facility i Empty	M _i	σ_i	i σ_i	Prob. Facility i Empty	M _i	σ_i	i σ_i	Prob. Facility i Empty
1	16	1.908	1.908	.1015x10 ⁻²	32	2.149	2.149	.2121x10 ⁻²	∞	2.366	2.366	.3698x10 ⁻²
2	8	1.349	2.698	.3186x10 ⁻¹	16	1.520	3.040	.4605x10 ⁻¹	∞	1.673	3.347	.6081x10 ⁻¹
4	4	.9539	3.816	.1785	8	1.075	4.299	.2146	∞	1.183	4.733	.2466
8	2	.6745	5.396	.4225	4	.7599	6.080	.4633	∞	.8367	6.693	.4966
Totals	30	2.613	7.389	.2440x10 ⁻⁵	60	2.943	8.325	.9710x10 ⁻⁵	∞	3.240	9.165	.2754x10 ⁻⁴

Using the same arguments which led to (4.13) and (4.14) previously, it is seen that the instantaneous proportion of all occupied servers applicable to jobs of type i is still given by (4.13). However, the average proportion is now obtained by substituting $k_i = \bar{N}_i$ in (4.13) and using (4.43) to give

$$\delta_i = \frac{i\bar{N}_i}{\bar{N}}, \quad (4.45)$$

which is analogous to (4.14).

Since the arrival rates are state dependent, the parameter g_i (defined previously to be the probability that an arrival is of type i) is also state dependent and is given by

$$g_i(k) = \frac{\lambda (M_i - k_i)}{\sum_{i \in S} \lambda (M_i - k_i)} = \frac{M_i - k_i}{\sum_{i \in S} (M_i - k_i)} \quad (4.46)$$

For an average number of messages of each type in the system,

$$k_i = \bar{N}_i,$$

$$g_i = \frac{\lambda (M_i - \bar{N}_i)}{\sum_{i \in S} \lambda (M_i - \bar{N}_i)} = \frac{M_i - \bar{N}_i}{\sum_{i \in S} (M_i - \bar{N}_i)} \quad (4.47)$$

Similarly on the departure side, the instantaneous proportion of the total service rate applicable to jobs of type i is given by (4.18), while on the average using $k_i = \bar{N}_i$ in (4.18) and applying (4.41) to (4.47) gives

$$\gamma_i = \frac{\sum_{j \in S} \lambda_j \bar{N}_i}{\sum_{j \in S} \lambda_j \bar{N}_i} = g_i \quad (4.48)$$

i.e., the proportional average arrival and departure rates are equal.

4.4 Admission Delays for Multi-Hop Synchronous Traffic

4.4.1 An Approximate Model for Admission Delays

It has been mentioned previously that messages require reservations in order to travel two or three hops in TI-NET, and that the admission delay incurred in obtaining reservations consists of the sum of the delays due to requesting, locating and delivering them [46,56]. Since all entrance nodes request reservations for a particular exit node according to their volumes of incoming synchronous traffic for that exit node, the admission delay at a particular entrance node may depend to some extent, on numerous other nodes in the network. Hence, a detailed network model for the purposes of analysing admission delays would be quite involved.

As a first step in the study of admission delays for multi-hop traffic, a simple approximate model is described in which the dependencies of admission delay on numerous nodes are lumped, and the random times to request, locate and deliver reservations, as well as to find sufficient channel capacity on the first link, are assumed to behave according to independent exponential distributions. The justification for this assumption is that many of the above

mentioned times depend on the nature of the traffic which in turn is assumed to have exponential interarrival times and message lengths. However, some of the times, such as the time to deliver reservations, may be somewhat more deterministic and in this sense the assumption represents a worst case situation.

Figure 4.20 shows an example of three hop traffic in TI-NET in which terminals at various entrance nodes are transmitting messages to other terminals (or a host computer), at a single exit node. (Again only one direction of transmission is shown for convenience). The objective is to model (approximately) the average admission delay incurred by a message at an entrance node before it begins transmission to its exit node. Another respect in which the model to be presented is an approximation, is that a homogeneous population is assumed in order that existing analysis pertaining to multiple resource models may be readily adapted to the TI-NET situation [46,115,116].

The admission delay model is shown in Figure 4.21, and consists of various service mechanisms which are each labelled with their mean service times; these in turn correspond to the mean times to request, locate, and deliver reservations, as well as to find sufficient capacity on the first (entrance) link in Figure 4.20. The mean service times in Figure 4.21 will be defined more precisely later.

FIGURE 4.20 EXAMPLE OF 3 HOP TRAFFIC IN TI-NET

FIGURE 4.21 MODEL FOR ADMISSION DELAYS IN A TI-NET ENTRANCE NODE

For the purposes of the admission delay model, TI-NET is assumed to be operating as a closed computer communications network, and as such the time required to find reservations may be approximated by a multiple-resource model, where the resources correspond to the various TI-NET links and nodes. However in TI-NET, the flow control scheme is expected to prevent steady state queues at the central nodes, and the capacity of the central link is potentially larger than that of the entrance and exit links. In addition, the exit link is expected to be more heavily utilized than either the entrance or central links. Thus, from the point of view of the multiple resource model, the exit link is expected to be the first resource to "saturate". This behavior is assumed to affect an entrance node as follows: an entrance node may not have any reservations for the "saturated" exit link and when it needs and requests some (from its central node) they may not arrive until one or more messages originating from other entrance nodes, have reached the same exit node, and have thereby freed their reservations.

Thus the model in Figure 4.21 illustrates that with probability α_1 the exit link operates below "saturation", in which case reservations arrive in a fairly short mean time. On the other hand with probability $\alpha_2 = 1 - \alpha_1$, the exit link will be "saturated", and it will take a longer mean time for reservations to arrive; this mean time is assumed to increase in accordance with the queueing component of the response

time of a saturated resource in the multiple resource model.

That is, the response time of the multiple resource model consists of service time, plus the time spent waiting in the queue. Since the objective here is to model admission delays only, the mean service time is subtracted from the response time (and replaced by a relatively small nominal delay - T_{21} - defined later). The behavior of the multiple resource model, as it applies to the TI-NET situation is now described.

4.4.2 A Simplified Multiple Resource Model for the Time to Locate Reservations in TI-NET

With only one saturating resource (the exit link) the general multiple resource model simplifies to a closed multi-server queueing system with finite (and homogeneous) population, i.e. $M/M/m/M$ [112,115]. The M terminals now consist of all the terminals at the various entrance nodes in Figure 4.20. Since the entrance and central link are assumed not to saturate, they are absorbed into the exit link. Also, the destination terminals (or host computer) at the exit node are assumed to be non-saturating resources with negligible response times, and are lumped into the exit node. The resulting simplified multiple resource model is shown in Figure 4.22.

The pdf for the number of messages in an $M/M/m/M$ system (P_k) is well known and is defined in Appendix A.5 along with other parameters.

FIGURE 4.22 SIMPLIFIED MULTIPLE RESOURCE MODEL
FOR THE TIME TO LOCATE RESERVATIONS
IN TI-NET

The mean number of messages in the system may be computed from the first moment of P_k , i.e.

$$N_t = \sum_{k=0}^M k P_k, \quad (4.49)$$

and the total mean time (T_t) in the M/M/m/M system may be determined from Little's result [114], which states that T_t is N_t divided by the average throughput. Since this is a closed system, the maximum throughput is $M\lambda$ msg's/sec and occurs when the system is empty, while the average throughput occurs when there are N_t in the system and is given by $\lambda(M - N_t)$. Thus

$$T_t = \frac{N_t}{\lambda(M - N_t)} \quad (4.50)$$

or the normalized mean time in the system is

$$m\mu T_t = \frac{m\mu}{\lambda} \cdot \frac{N_t}{M - N_t} \quad (4.51)$$

Another approach leading to (4.50) is to recognize that the average throughput per server must equal the think time per terminal ($1/\lambda$) plus T_t ; i.e. a terminal does not generate a new message until it receives a reply and "thinks" about the next one it will send. Therefore

$$T_t = \frac{N_t}{M/(1/\lambda + T_t)} \quad (4.52)$$

and solving for T_t gives (4.50).

As usual T_t consists of the time a message spends waiting in the queue T_q , plus the time in service T_s . Since the objective here is to model admission delays only, T_q is obtained by subtracting the service time $T_s = 1/\mu$ from (4.51) to give a normalized time spent in the queue of

$$m\mu T_q = m \left(\frac{\mu}{\lambda} \frac{N_t}{M - N_t} - 1 \right). \quad (4.53)$$

It has been mentioned previously that a closed network approach is not strictly applicable to voice traffic. However, such an approach is flexible since in the limit as the number of terminals becomes very large, and the arrival rate λ per terminal becomes very small, the average throughput $\lambda(M - N_t)$ approaches a constant λM , corresponding to the open infinite population case. This point is discussed further in Appendix A.5 where the behavior of an M/M/m/M system is compared to that of an M/M/m system by plotting the normalized mean time spent in the queue against the utilization factor for both systems.

A more conventional approach to representing the performance of a multiple resource model is to plot the mean time in the system against the number of terminals. In line with this practice is an example of (4.50) in Figure 4.23 for $m = 32$ and $\lambda/\mu = 1$. Both the true and asymptotic behavior are shown. The equations for the lower and upper asymptotes

FIGURE 4.23

NORMALIZED MEAN TIME SPENT IN THE
QUEUE FOR M/M/m/M SYSTEM

$$m = 32$$

$$\lambda/\mu = 1$$

(SEE APPENDIX A.5)

NORMALIZED MEAN TIME IN THE QUEUE μT_q

TRUE
BEHAVIOR

UPPER
ASYMPTOTE

$$\mu T_{qu} = M - M^*$$

LOWER
ASYMPTOTE

$$\mu T_{ql} = 0$$

NUMBER OF TERMINALS M

$$M^* = 64$$

of the normalized mean time in the system may be obtained by simplifying the corresponding equations for the general multiple resource model, and are given by:

$$\mu m T_{tL} = M^* - \frac{m\mu}{\lambda} \quad (4.54)$$

and

$$\mu m T_{tU} = M - \frac{m\mu}{\lambda} \quad (4.55)$$

where M^* is defined to be the maximum number of terminals which could be accommodated without interference (queueing), if all messages required exactly the deterministic amounts of service and "thinking" equal to $1/\mu$ and $1/\lambda$ respectively. Thus the maximum number of terminals which could be serviced in a time equal to the cycle time (service time plus think time) is

$$M^* = m \frac{1/\mu + 1/\lambda}{1/\mu} = m \left(1 + \frac{\mu}{\lambda} \right) \quad (4.56)$$

The equations for the asymptotes for the mean time spent in the queue (as in Figure 4.23) are obtained by subtracting the normalized service time $\mu m T_S = m\mu/\mu = m$ from (4.54) and (4.55) and using (4.56) to give

$$\mu m T_{qL} = 0 \quad (4.57)$$

$$\mu m T_{qU} = M - M^* \quad (4.58)$$

Finally by taking the derivative of (4.58) with respect to M , it is noted that the slope of the upper asymptote in Figure 4.23 is unity.

4.4.3 Formulation of the Admission Delay Model

It is now possible to define more precisely the various mean service times for the admission delay model in Figure 4.21.

- T_1 is the mean time to request reservations
- T_{21} is a nominal mean time for the central node to find reservations and is assumed to be relatively small, and constant with respect to the number of terminals M , since the exit link is operating at or below "saturation". I.e., for $M \leq M^*$, $T_{qL} = 0$.
- T_{22} is similar to T_{21} except that it is for the above "saturation" region and is assumed to vary accordingly. I.e. for $M \geq M^* + 1$, $T_{22} = T_{21} + T_{qU}$.

Somewhat later, a pdf for the number of signed on terminals will be assumed and T_{22} is then averaged for $M^* < M \leq M_{\max}$, and it is this averaged value of T_{22} (i.e. $\bar{T}_{22}$) which will eventually be used in the model in Figure 4.21.

- T_3 is the mean time to deliver reservations to the entrance node.
- T_4 is principally the mean time to find sufficient capacity on the first (entrance) link; also, in the event that 'activity fields' [4,56] cannot be updated at the same time

that a message is ready to be transmitted, the extra time required to update is also considered to be lumped in with T_4 .

Note that as usual, the large "oval" enclosing all of the service mechanisms in Figure 4.21, means that there can only be one "customer" in the oval at a time [112]. This implies that reservations for a particular destination are only requested (and acquired) one at a time, whereas it is possible that at a particular time, more than one reservation may be required for that destination. In fact TI-NET is capable of requesting more than one reservation at a time for the same destination node [46], so that in this sense as well, the model in Figure 4.21 is conservative and should provide a conservative estimate of TI-NET performance, as far as admission delays are concerned.

For convenience the previously mentioned mean times are converted to rates as follows:

$$a = 1/T_1, b_j = 1/T_{2j}, c = 1/T_3 \text{ and } d = 1/T_4. \quad (4.59)$$

The density for the total service time in each path of the admission delay model in Figure 4.21, is the convolution of four different exponential densities and is shown in Appendix A.6 to be:

$$f_j(t) = A_j e^{-at} + B_j e^{-b_j t} + C_j e^{-ct} + D_j e^{-dt}, \quad (4.60)$$

where $j=1$ for the upper and 2 for the lower path in Figure 4.21,

$$a \neq b_j \neq c \neq d,$$

and the constants A_j , B_j , C_j and D_j are defined in Appendix A.6. Also given in this appendix are the mean ($\bar{t}_j$), variance (σ_j^2), second moment ($\bar{t}_j^2$) and coefficient of variation (cv_j) for $f_j(t)$. (If any of the parameters a , b_j , c and d are equal, then from a computational and practical point of view, $f_j(t)$ can still be evaluated by making those parameters differ very slightly).

Thus the net service density for the admission delay model is:

$$f(t) = \alpha_1 f_1(t) + \alpha_2 f_2(t) \quad (4.61)$$

where α_1 and α_2 are the probabilities of being above and below "saturation" respectively.

Similarly the combined mean, second moment, variance and coefficient of variation for $f(t)$ are: (see (A.6.7) to (A.6.10))

$$\bar{t} = \alpha_1 \bar{t}_1 + \alpha_2 \bar{t}_2 \quad (4.62)$$

$$\bar{t}^2 = \alpha_1 \bar{t}_1^2 + \alpha_2 \bar{t}_2^2 \quad (4.63)$$

$$\sigma^2 = \bar{t}^2 - \bar{t}^2 \quad (4.64)$$

$$cv = \frac{\sigma}{\bar{t}} \quad (4.65)$$

The probabilities α_1 and α_2 can be related to the number of

terminals M , which is usually the independent variable against which, the response time for the saturated resource in a multiple resource model, is plotted (e.g. Figure 4.23). Thus in a system where the number of terminals is dynamically changing (i.e. signing on and off), one may consider

$$\alpha_1 = P(M \leq M^*) \text{ and } \alpha_2 = P(M > M^*) \quad (4.66)$$

In this regard, and for simplicity, a uniform pdf for the number of signed on terminals feeding the saturated resource in Figure 4.22 is assumed, and is shown in Figure 4.24(b). By inspection, the pdf is

$$P_M = \frac{1}{\sum_{M=M_{\min}}^{M_{\max}} 1} = \frac{1}{M_{\max} - M_{\min} + 1}, \quad M_{\min} \leq M \leq M_{\max} \quad (4.67)$$

and using (4.66) and (4.67)

$$\alpha_1 = \sum_{M=M_{\min}}^{M^*} P_M = \frac{M^* - M_{\min} + 1}{M_{\max} - M_{\min} + 1} \quad (4.68)$$

and

$$\alpha_2 = \sum_{M=M^*+1}^{M_{\max}} P_M = \frac{M_{\max} - M^*}{M_{\max} - M_{\min} + 1} \quad (4.69)$$

It is now possible to obtain effective mean times to obtain reservations for the non-"saturated" and "saturated" regions of Figure 4.23, for which the asymptotic behavior is given by (4.57) and (4.58). This is illustrated in Figure 4.24(a).

FIGURE 4.24

- (a) ASYMPTOTIC MEAN TIMES TO LOCATE RESERVATIONS
 (b) pdf FOR THE NUMBER OF ACTIVE TERMINALS

Thus for $M_{\min} \leq M \leq M^*$, which occurs with probability α_1 , using (4.57) gives $T_{2j} = T_{21} + T_{qL} = T_{21}$, a relatively small and constant mean time to locate reservations, since the exit link is operating at or below "saturation". However, for $M^* + 1 \leq M \leq M_{\max}$ (with probability α_2) $T_{2j} = T_{21} + T_{qU} = T_{22}$, a delay which increases with the number of terminals since the exit link is now "saturated". In general from (4.58) and (4.69),

$$T_{22} = T_{21} + T_{qU} = T_{21} + \frac{1}{m\mu} (M - M^*). \quad (4.70)$$

$$\therefore T_{22_{\min}} = T_{21} + T_{qU_{\min}} = T_{21} + \frac{1}{m\mu} \quad (4.71)$$

$$\text{and } T_{22_{\max}} = T_{21} + T_{qU_{\max}} = T_{21} + \frac{1}{m\mu} \frac{\alpha_2}{P_M}. \quad (4.72)$$

Now $\bar{T}_{22}$ is considered to be the average of the mean times corresponding to the "saturated" region. Thus from (4.71) and (4.72)

$$\bar{T}_{22} = \frac{T_{22_{\min}} + T_{22_{\max}}}{2} = T_{21} + \frac{1}{2m\mu} \left(1 + \frac{\alpha_2}{P_M} \right) \quad (4.73)$$

It is now possible to evaluate (4.61), the pdf for the overall service time. To do this, various input parameters are specified. An example of these is given in Tables 4.2 and 4.3, and they include the individual mean service times T_1 , T_{21} , T_3 and T_4 , and the probability of operating below "saturation" α_1 . From the input parameters, various other parameters are computed using the equations already presented,

TABLE 4.2

PARAMETERS FOR EXAMPLES IN FIGURES 4.25 AND 4.26

SPECIFIED INPUT PARAMETERS

$m = 32$ servers
 $R = 150$ char/sec per server
 $\ell = 35$ char/msg
 $M_{\min} = 32$ terminals
 $T_1 = 2.5$ ms
 $T_{21} = 5$ ms
 $T_3 = 7.5$ ms
 $T_4 = 10$ ms

COMPUTED PARAMETERS

EQUATION

$C = 4800$ char/sec	mR
$\mu = 4.286$ msg's/sec per server	R/ℓ
$T_S = 233.3$ ms	$1/\mu$
Slope = .00729 sec/terminal	$1/m\mu$
$\bar{t}_1 = 25$ ms	(A.6.7)
$\sigma_1 = 13.69$ ms	(A.6.8)
$cv_1 = .5477$ ms	(A.6.10)
$T_{22\min} = 12.29$ ms	(4.74)

TABLE 4.3

PARAMETERS FOR EXAMPLES IN FIGURES 4.25 AND 4.26

COMPUTED PARAMETERS	SPECIFIED INPUT PARAMETERS			EQUATION
	$\alpha_1 = .7$ $M_{\max} = 100$	.5 100	.3 100	
P_M	.01449	.01449	.01449	(4.67)
M^*	79	66	52	(4.65) and (4.66)
λ/μ	.6809	.9412	1.600	(4.56)
$T_{22\max}$	158.1 ms	252.9 ms	355.0 ms	(4.72)
$\bar{T}_{22}$	85.21 ms	132.6 ms	183.7 ms	(4.73)
$\bar{t}_2$	105.2 ms	152.6 ms	203.7 ms	(A.6.7)
σ_2	86.16 ms	133.2 ms	184.1 ms	(A.6.8)
cv_2	.8189	.8730	.9040	(A.6.10)
$\bar{t}$	49.41 ms	87.88 ms	149.3 ms	(4.62)
σ	61.25 ms	113.6 ms	174.3 ms	(4.63) and (4.64)
cv	1.240	1.293	1.168	(4.65)

and referenced in Tables 4.2 and 4.3. These parameters include the mean, standard deviation and coefficient of variation for the combined service time, as well as for the service time of each path of the model in Figure 4.21; they also include M^* , the number of terminals at which "saturation" starts, and $\bar{T}_{22}$, the overall average time to obtain reservations when in the "saturated" region. $\bar{T}_{22}$ is the value used for T_{22} in (4.59) and (4.60) for $j = 2$.

The parameters in Tables 4.2 and 4.3 correspond to the examples of $f(t)$ plotted in Figures 4.25 and 4.26. Figure 4.25 illustrates both the individual service time pdf's for operation below and above "saturation", as well as the combined pdf for the model in Figure 4.21; for this example the corresponding mean service times (from Tables 4.2 and 4.3) are $\bar{t}_1 = 25$, $\bar{t}_2 = 105$ and $\bar{t} = 49$ ms respectively. Also, a degree of "hyperexponential" behavior [112] is indicated, in that the coefficient of variation for $f(t)$ is greater than one (i.e. 1.24). The reason is that $f(t)$ has essentially two regions: a low delay peaked region contributed by $f_1(t)$, and a high delay fairly flat region contributed by $f_2(t)$.

Figure 4.26 illustrates the effect on $f(t)$ of reducing α_1 (the proportion of time spent in the "unsaturated" region) from 0.7 (in Figure 4.25) to 0.5 and 0.3. There are different ways in which α_1 can be varied. In Figure 4.26, the range of terminals is held fixed (i.e. $M_{\min} = 32$ and $M_{\max} = 100$), and the think time per terminal is allowed to change in order

3

FIGURE 4.25

FIGURE 4.26

SERVICE TIME PDF FOR THE TI-NET ADMISSION DELAY MODEL

THINK TIME VARIES WITH α_1

RANGE OF TERMINALS FIXED: $M_{min} = 32$, $M_{max} = 100$
(SEE TABLES 4.2 AND 4.3)

SERVICE TIME IN MULTIPLES OF 10 MSEC

to obtain the desired variations in α_1 . In practice this is analogous to comparing admission delay performance for two different job applications with the same range of terminals but different think times per terminal. In Figure 4.26 the peak of $f(t)$ remains in the same location since this is contributed by $f_1(t)$ which is not altered unless T_1 , T_{21} , T_3 or T_4 are varied. However, since the minimum and maximum number of terminals is held fixed, reducing α_1 causes "saturation" to take place at a lower M^* as in Table 4.3. This results in increased times to find reservations in the "saturated" region and thereby affects $f_2(t)$, and hence $f(t)$. I.e., for $\alpha_1 = 0.5$ and 0.3 , the mean service time has increased from $\bar{t} = 49$ to 88 and 149 ms respectively. It is also noted in Table 4.3 that the coefficient of variation is higher for $\alpha_1 = 0.5$ than for $\alpha_1 = 0.7$ or 0.3 ; the reason is that in the limit as α_1 approaches either 1 or 0 , $f(t) = f_1(t)$ or $f_2(t)$ respectively, for which the coefficient of variation is less than one.

Another way to vary α_1 , is to keep the think time per terminal fixed, and allow the range of terminals to change. This is analogous to comparing admission delay performances for a different range of terminals, for the same job application. Figure 4.27 together with Tables 4.2 and 4.4 give an example of such a case; the results look similar to those in Figure 4.26 except that now the mean admission delay is much larger. I.e., for $M_{\max}$ increased from 100 to

FIGURE 4.27

SERVICE TIME pdf FOR THE TI-NET ADMISSION
DELAY MODEL

THINK TIME FIXED; $\lambda/\mu = .6809$

RANGE OF TERMINALS VARIES WITH α_1 .

(SEE TABLES 4.2 AND 4.4).

SERVICE TIME IN MULTIPLES OF 10 MSEC

TABLE 4.4
PARAMETERS FOR EXAMPLE IN FIGURE 4.27

COMPUTED PARAMETERS	SPECIFIED INPUT PARAMETERS		
	$a_1 = .0.7$ $\lambda/\mu = .6809$	0.5 .6809	0.3 .6809
P_M	.01449	.01042	.00625
M^*	79	79	79
$M_{\max}$	100	127	191
$T_{22\max}$	158.1 ms	355.0 ms	821.7 ms
$\bar{T}_{22}$	85.21 ms	183.7 ms	417.0 ms
$\bar{t}_2$	105.2 ms	203.7 ms	437.0 ms
σ_2	86.16 ms	184.1 ms	417.2 ms
cv_2	.8189	.9040	.9547
$\bar{t}$	49.41 ms	114.3 ms	313.4 ms
α	61.25 ms	158.2 ms	396.9 ms
cv	1.240	1.384	1.266

127 and 191 terminals, $\bar{t} = .114$ and $.313$ ms at $\alpha_1 = 0.5$ and 0.3 respectively. This is reflected by the tails of the pdf's which are now much more uniform (flat) than in Figure 4.26, as well as by the coefficients of variation which are larger than in Figure 4.26.

CHAPTER 5

EXPERIMENTAL WORK ON ENCODED VOICE TRANSMISSION IN TI-NET

5.1 Background and Preliminaries

5.1.1 Voice Terminals

5.1.1.1 An LPC Voice Response System

The first experimental work involving voice transmission in TI-NET [47] used a stored source of encoded voice feeding one network node, and a software speech synthesizer receiving the encoded voice from another node. The source of encoded voice and the speech synthesizer were provided as a result of previous work at Carleton University in the area of voice response systems [51-53]. An existing voice response system consisted of a data base of speech synthesis parameters stored on disk, and a software implementation of a linear predictive (LPC) speech synthesizer operating in real time on a PDP-11/45 (see Figure 5.1). A request for a voice response is made through the keyboard terminal. This causes appropriate synthesis parameters to be read from disk into memory and, then synthesized into digitized speech samples. The corresponding audio output is then produced by means of a digital-to-analog converter.

The data base synthesis parameters for the voice response system are stored on disk in groups according to the discrete English words they represent. That is, each word is described by a synthesis parameter file of at most 1024 bytes.

FIGURE 5.1 FUNCTIONAL DIAGRAM OF EXISTING VOICE RESPONSE SYSTEM

For the purposes of the voice transmission experiment, and in order to avoid the disk response time (of several tens of milliseconds) for each request for a file transfer, a subset of the data base was loaded into core at the beginning of run time. The subset occupied a 5K-word array and consists of files for the digits 0 to 9. This array was used as the encoded voice source for the experiment.

The experiment demonstrated the feasibility of transmitting data and encoded voice in packets, but was not pursued further because of the high bit rate requirements of the LPC synthesizer (≈ 15 kb/s), which necessitated non real time transmission over the 9.6 kb/s link between the two TI-NET nodes. Also, there was insufficient processing time available to decode speech synthesis parameters, which if suitably encoded by known techniques [77-80] could have reduced the bit rate to 2.4 kb/s or lower. Finally, the voice response system in its existing form, only operated one word at a time, whereas experimentation with continuous and conversational speech was desired. For these reasons two real time commercial speech processors were used for further work.

5.1.1.2 The VADAC Channel Vocoder [70-73, 117-122]

Most of the experimental work involving voice transmission in TI-NET used two commercial speech processors known as Voice Analyzer Data Converters. The VADAC is a digital 16 channel pitch excited vocoder which operates at 2.4 (or 4.8) kb/s. The major functions of the VADAC are illustrated

in Figure 5.2. Speech signals from a telephone (microphone or tape recorder) enter the signal conditioning circuitry which may optionally normalize the speech power spectrum by means of a Voice Operated Gain Addjustment Device (VOGAD). The signal is then sampled at 7 kHz and digitized with an 8 bit linear quantizer. Following this, the signal is weighted by a 7 bit Hamming window; then the spectrum analyser performs high resolution 256 point by 8 bit discrete Fourier transforms, on 37 msec real time speech segment lengths (window widths). The spectral resolution is 27 Hz and the analysis range is from 200 to 3450 Hz.

Following spectrum analysis, the spectrum information is subjected to pre-emphasis, and the linear summation of spectral components for each channel is encoded with 3 bits (except for the last of the 16 channels which is only assigned 2 bits). The spectrum information is also used to obtain a measure of the pitch fundamental frequency (70 to 320 Hz) by means of a modified cepstrum-like technique; this technique is suitable when the pitch fundamental and other harmonics are missing as is typical of speech signals from loaded telephone links. After suitable scaling and windowing of the extracted pitch information, the most likely pitch frequency is selected and coded with 6 bits. The absolute amplitude of the pitch information as well as the ratio of high frequency to low frequency spectrum energy is then used to obtain a voiced/unvoiced decision.

FIGURE 5.2 FUNCTIONAL DIAGRAM OF VADAC CHANNEL VOCODER
(Courtesy of E. Systems Inc., Dallas, Texas
[130]).

At this point the 47 bits of spectrum information and 6 bits for pitch fundamental frequency (set to zero if the speech segment is unvoiced) are formatted into an output frame of 53 bits; finally, a frame bit is added, thus giving the output format shown in Figure 5.3.

On the synthesizer portion of Figure 5.2, the received frames are de-formatted to provide separate pitch and spectrum information signals. The source of voiced sounds is produced by frequency modulating a pulse generator according to the received pitch fundamental frequency, while the source of unvoiced sounds is produced by a pseudo random noise generator. In each case the source is also amplitude modulated by the received spectrum information. Thus voiced signals are synthesized by summing harmonics of the received pitch frequency weighted in accordance with the received spectrum amplitude information. Similarly, unvoiced signals are synthesized by a weighted summation of noise like components. These summations are output as time varying analogue signals representative of the original speech signal. The "modulation control" in Figure 5.2, checks the incoming pitch and spectrum information with respect to the time history of preceding information. This allows forward acting error control (based on known properties of speech parameters) to be applied in the presence of transmission errors.

FIGURE 5.3 FORMAT OF 2.4 kb/s SYNCHRONOUS OUTPUT
OF VADAC SPEECH PROCESSOR

5.1.2 External Link Considerations

5.1.2.1 Subnet Access

Before encoded voice can be transmitted within TI-NET the voice terminal must be able to gain access to the subnet. In general, data users can gain access to TI-NET with minimum standardization [4,48]; i.e., all the user provides to the network is a serial bit stream, consisting of his own data and protocol, through a RS 232-C interface (in addition a communication link - telephone line or small software package - is required for morning sign-on and evening sign-off).

Access to TI-NET with a simple serial bit stream through a RS 232-C interface is also possible for encoded voice users. However, at present, most encoded voice terminals appear to be frame synchronous devices which transmit continually whether or not anyone is talking [64,68,72-76]. Therefore, a protocol is required which takes advantage of the 60 to 65% idle time in conversations (one way) [43] and provides the capability for statistical multiplexing of data and vocoded voice.

5.1.2.2 Protocols

Basic requirements for an external link protocol for full duplex encoded voice transmission include a means of identifying the active periods in speech, and a means of providing data transparency. Error control based on retransmission is not considered to be a requirement because of the

relatively non-critical nature of voice as compared to data traffic. A suitable way to achieve the above requirements is by means of a simplified protocol based on the ISO standard HDLC [123], (or based on SDLC which is a similar protocol by IBM [124]). As shown in Figure 5.4(a), this simplified HDLC protocol involves delimiting active speech periods by HDLC flags at the start and end of each period, and uses bit stuffing for data transparency. This method accommodates active periods of variable length including a prime number of bits. Figure 5.4(b) shows a sequence of three active periods, each delimited by HDLC flags and separated by SYN (Synchronization) characters [125]. The SYN characters may be used to maintain character synchronization between a voice terminal and a TI-NET node during idle periods, but are not transmitted through TI-NET. Alternatively the link may be set to an off (MARK HOLD) state during idle periods, and character synchronization re-established at the start of each new active period by preceding each period with 1 or more (as required) HDLC flags.

At the receiving voice terminal, removal of the protocol consists of stripping off HDLC flags and performing bit deletion on the message, to restore the original encoded voice. Also, for a frame synchronous voice terminal (such as the VADAC) it is necessary to maintain a continuous frame structure for the received signal, so that the receiver does not lose frame synchronization. Therefore locally generated

idle frames must be supplied to the receiver during idle periods. However, because of variable network delays, the receiver will likely observe misalignment between the end of the last frame of idle periods, and the start of the first frame of active periods. Therefore in general, a new message must be delayed from zero to one frame in order to re-establish frame alignment.

It is expected that a protocol as just described would be implemented in hardware (i.e. by a micro-processor), perhaps as part of the voice terminal equipment (in much the same way as a modem, or encryptor plug-in option is provided). This is indicated in Figure 5.5. Also shown is a TI-NET entrance and exit node through which voice transmission is taking place (only one direction is shown for convenience).

At a TI-NET entrance node, the HDLC flags are stripped off and bit deletion may be performed by hardware in the serial synchronous interface. The remaining voice message is then packetized and transmitted to the exit node where bit insertion and flags are again applied to the messages before transmission to the receive voice terminal. Bit deletion and reinsertion in the node interfaces is not essential, and may be avoided at a small loss in transmission efficiency.

To obtain some indication of what this loss in efficiency would be for encoded voice, the VADAC 2.4 kb/s digital output was analyzed (by software means) when the VADAC voice input was excited by a 2½ minute taped session of male monologue

FIGURE 5.5 SUBNET ACCESS FOR ENCODED VOICE TRANSMISSION

speech (see Appendix B.4). The analysis revealed some 16,000 active bytes with only 91 instances of 5 consecutive '1's in the digital output. Thus 91 bit insertions of '0's would be required to ensure no imitation of the '01111110' HDLC flag characters. When averaged over the active bytes only, this represents a negligible loss in efficiency of 0.07%. Although this result is not necessarily representative of voice transmission in general, it suggests that eliminating bit deletion and re-insertion in the node interfaces will not have a significant effect on transmission efficiency.

Before ending this discussion it is mentioned that external link protocols for encoded voice transmission based on methods other than HDLC or SDLC are possible. However, a disadvantage of these methods is that they are (8 bit) byte rather than bit oriented. Bit orientation is desirable (but not essential) from the point of view that the frame length for most vocoders is not an integer number of bytes, (e.g. 1 VADAC frame = 54 bits at 44.4 frames/sec. [70]), and the smallest unit of voice message transmission would, in most cases of interest, be a frame.

5.2 External Link Protocol Implementation [48]

Normally, this protocol would be implemented in hardware as part of the speech processing equipment, but for convenience, a simplified protocol was implemented in software in the TI-NET nodes (see Appendix B.1 for documentation).

For this reason it was not necessary to implement bit stuffing or to use HDLC flags. The implementation was made possible by using the spectrum amplitude information present in each frame of the vocoder output. The vocoders used for this work were the VADAC II and VADAC V speech processors described previously. Also, the frame structure for 2.4 kb/s (synchronous) operation was shown in Figure 3.3. When the vocoder is idle, all spectrum descriptor bits are low, while some are high when the vocoder is active. The software implementation consists of receive (from VADAC) and transmit (to VADAC) routines.

The receive routine performs the following functions:

- Initial framing and continuous frame checking of incoming 54 bit frames is performed. Reframe capability is provided.
- Incoming frames are stored 4 at a time (27 bytes) to facilitate byte by byte processing. This was done for convenience since the DEC Dull serial interfaces used to interface the VADAC's to the PDP-11's are byte oriented [126], as are the present TI-NET buffering routines [50].
- An initial sequence of frames is immediately transmitted to enable the VADAC receiver to acquire initial frame synch; this avoids having the receiver synchronize on the first active frame sequence.

- Active and idle frames are identified one at a time with no speech clipping which results in a fixed one frame delay (22.5 ms) as illustrated in Figure 5.6. An alternative method, - i.e. predicting the state (active or idle) of the next frame based on the state of the present one - would require no delay, but would result in as much as one frame (22.5 ms) of speech clipping at the start of speech messages. Although this amount of speech clipping is not enough to be readily perceptible [2], the first approach was used since in principle it seemed more desirable to incur a small fixed delay, than to destroy even a small part of each message.
- A number of frames of hangover is provided at the end of each active frame sequence as indicated in Figure 5.6 (note that the two frames of hangover shown is a convenient example and is not necessarily the recommended number). Hangover is used to provide some assurance that speech messages have ended; i.e. stop consonants in speech [43] are preceded by short pauses (idle periods) that occur in the midst of words and phrases. The hangover extends active periods in speech to absorb short pauses and thereby form longer continuous speech messages. The number of hangover frames is programmable, and the factors influencing the number that should be used (i.e. transmission efficiency and flow control considerations)

FIGURE 5.6 PROCESSING OF VOCODER FRAMES BEFORE PACKETIZING

are discussed later in conjunction with the emulation of admission delays.

The transmit routine performs the following functions:

- The initial frame sequence received is output to the VADAC receiver and is followed immediately by locally stored idle frames in order to establish and sustain initial frame synch in the VADAC receiver.
- Outgoing idle frames are flagged and frames of newly arrived messages are identified. I.e. the present implementation is such that the first frame pulse in speech messages is not always located in the first bit position of a byte, but may be in one of four possible positions. Therefore a few frames (5 or 6 seems adequate) are examined at the start of each message to identify which of four possible frame sequences is being received. This involves a fixed delay which however, is absorbed in the TI-NET output buffering delay for synchronous error control. This particular implementation (although avoidable with additional processing) evolved as a result of 54 bit vocoder frames, and the byte oriented hardware and software mentioned previously.
- Once newly arrived active frames are identified and outgoing idle frames are flagged, the first of the active frames in each message is meshed with the last of the idle frames in order to maintain VADAC receiver

frame synch. This is demonstrated in Figure 5.7.

Meshing is required because of variable network delays which cause the active messages to arrive at random times with respect to the locally generated idle frames. Therefore, active frames are buffered until the present idle frame has been output. A variable meshing delay due to this buffering, of from zero to one frame is incurred.

5.3 Packetizing and Transmitting Encoded Voice within the Experimental Terrestrial Network

5.3.1 Description of the Two-Node Terrestrial Network

The transmission of data and encoded voice has been demonstrated in the experimental two-node TI-NET shown in Figure 5.8. The two nodes (i.e. PDP-11's) are joined by a 9.6 kb/s synchronous link using limited distance modems. The synchronous link protocol is implemented in hardware by the (DEC DUPl1) synchronous adapters which perform flag and abort generation, CRC 16 calculation and bit stuffing. Each node has a teletype terminal to represent asynchronous traffic, and a VADAC II speech processor operating at 2.4 kb/s to represent synchronous traffic. The VADAC interfaces shown in the figure are MIL 188 to RS 232-C conversions for the VADAC II's. One of the modem clocks serves as a "master" clock, to which the other modem, the PDP-11's, the speech processors and the synchronous adapters have been synchronized. Details related to the hardware aspects of interface conversion and clock distribution are given in Appendix B.2.

FIGURE 5.7 PROCESSING OF FRAMES BEFORE DELIVERY TO VOCODER

FIGURE 5.8 BLOCK DIAGRAM OF THE EXPERIMENTAL TI-NET

The software implementation of a protocol for encoded voice transmission described previously is incorporated in the TI-NET nodes.

5.3.2 Internal Transmission Protocol for Synchronous Voice and Data Traffic

Recall that packetizing in a TI-NET entrance node consists of the formation of alternate 5 ms synchronous and 5 ms asynchronous MP's. Therefore by the end of every 10 ms service interval, synchronous terminals load enough data into the entrance node input buffer to fill an mp, at which time the buffer is serviced as indicated in Figure 5.9 (similar to Figure 4.2(a), (b) and (c)).

Two factors related to packetizing are first, that most synchronous messages will not be an integer number of mp's in length. Therefore a method is required for "padding" those mp's (occurring at the end of messages) which are only partially filled as illustrated in Figure 5.9. Secondly, there is a possibility of synchronous messages occurring very close together (or being contiguous) and (depending on the implementation of node buffering), of becoming concatenated in the node buffers. This is not a problem for data messages using a BISYNC like protocol; however, messages using SDLC or HDLC which may have their flags (and bit stuffing) removed by the entrance node and reapplied by the exit node, will be exposed to the possibility of concatenation. (Examples of concatenation in the entrance and exit node buffers are given in Appendix B.3). For encoded voice, however, (assuming message

10 MSEC SERVICE INTERVALS

MESSAGE BEFORE PACKETIZING

AMOUNTS OF DATA
CORRESPONDING TO
FULL mp'sEND OF MESSAGE:
LAST PARTIALLY
FILLED mp WILL
REQUIRE PADDINGENTRANCE NODE
BUFFER CONTENTS
IN mp's

2

1

0

FIGURE 5.9 FORMATION OF mp's IN A TI-NET ENTRANCE NODE

lengths are strictly an integer number of frames), concatenation may normally only introduce a small amount of time scale distortion by destroying the idle time between the close messages. However, since the objective in TI-NET is to provide transparency in time, a method is required to prevent concatenation for both voice and data messages.

Also, with regard to handling close or contiguous messages, it is noted that the average amount of protocol removed (by an entrance node) from the incoming messages is expected to exceed the average amount of padding required to fill the last mp of these messages. Furthermore, in the case of encoded voice, the minimum inter-arrival time of two speech messages is one vocoder frame which is likely to always exceed even the maximum amount of padding required.

A protocol has been implemented in the experimental TI-NET which uses two bits of overhead at the end of each synchronous mp with which the padding of partially filled mp's and the prevention of concatenation is achieved as follows:

<u>Trailer Bits</u>	<u>Interpretation</u>
00	NOT LAST DATA mp
01	NOT LAST VOICE mp
10	LAST mp, UNPADDED
11	LAST mp, PADDED

Padded mp's contain pad bits which are all zeros except for the first bit following the data which is a "1" to indicate where the data ends and the padding starts. This protocol

has the advantage that it is transparent to any type of synchronous traffic. Padding is done with zero's rather than one's so as to minimize bit stuffing by synchronous adapters (e.g. DEC DUP11 serial interfaces are used in the nodes - PDP-11's - of the experiment TI-NET). The implementation resides in the experimental TI-NET synchronous multiplex and synchronous demultiplex tasks which are documented in Appendix B:1. Implicit in the implementation is end of message signalling both from the device input (e.g. receive from VADAC) routine to the multiplex task, and from the demultiplex task to the device output (e.g. transmit to VADAC) routine.

Figure 5.10 illustrates a more general protocol (not implemented) in which coding for the purposes of padding and preventing concatenation has been combined with coding required by the TI-NET flow control scheme. Recall from the TI-NET overview in Chapter 3 that TI-NET flow control is based on reservations [45,56]. The number of reservations held by a star node is controlled by its central node by means of the reservation information shown coded in Figure 5.10. Reservation information is conveyed by all synchronous mp's travelling from a central node to an exit star node, except for the last mp in each message. Hence, for flow control purposes, last mp's are interpreted as carrying no reservation information. This is consistent with the intention of allowing reservations to remain with the entrance node, to be used by subsequent messages or to be cancelled later.

FIGURE 5.10 ILLUSTRATION OF THE CODING FOR CONVEYING RESERVATION INFORMATION IN (NOT LAST) VOICE AND DATA mp's AND FOR PADDING PARTIALLY FILLED mp's

Finally with regard to Figure 5.10, it is mentioned that information pertaining to the fixed length of mp's (for a given terminal speed) is implied in the receiving node "activity field" [4], and is used to demultiplex synchronous MP's. This activity field indicates those users which are currently transmitting a message and is updated using the synchronous MP header. The philosophy behind different protocols for enabling the proper processing of mp's is discussed in more detail in a later section [49].

5.3.3 Experimental Network Delays

The experimental TI-NET is a scaled-down version of the actual TI-NET. Thus packetizing consists of the formation of alternate 20 ms synchronous and 20 ms asynchronous MP's. The service times for synchronous terminals (i.e. the VADAC II speech processors) are therefore every 40 ms (rather than every 10 ms) as shown in Figure 5.11(a).

In general, Figure 5.11 illustrates the internal delays of the experimental two-node network (and in this sense is similar to Figure 4.2 which does the same for the actual TI-NET).

Figure 5.11(b) shows a synchronous message (from the VADAC II) appearing at the entrance node at an arbitrary time with respect to the node servicing times. In general this results in an entrance or packet formation delay of 40 to 80 ms (or 60 ± 20 ms), as seen in Figure 5.11(c).

FIGURE 5.11 EXPERIMENTAL TI-NET INTERNAL DELAYS

For reasons related to implementation, multiplexing (i.e. MP formation) in the experimental network is scheduled to take place 10 ms before (rather than immediately before) MP transmission. This gives rise to the 10 ms delay indicated in Figure 5.11(d); however, it is pointed out that no such corresponding delay exists in the actual TI-NET.

Figure 5.11(e) simply shows that since both nodes are local, propagation and transmission delays are essentially zero.

Since MP lengths are 20 ms, a 20 ms store and forward delay is incurred as in Figure 5.11(f).

Finally, the implementation of error control based on retransmission for synchronous traffic involves a 120 ms output buffering delay as illustrated in Figures 5.11(f) and (g); in this regard it is emphasized that because of the relatively non-critical nature of voice as compared to data traffic, error control would not normally be applied to voice traffic in TI-NET (although it could be on an optional basis). However, in the experimental system used here (Figure 5.8), the only synchronous terminal connected to each node was a VADAC speech processor, and therefore the synchronous error control capability of the experimental system was demonstrated by using these voice terminals. Also, in the implementation of the external link protocol for encoded voice transmission described earlier, the output buffer associated with error control was also used to mesh the frames of newly arrived voice messages with previous

outgoing idle frames.

The internal delays are now summarized as follows:

Packet Formation (Entrance)	60 ± 20 ms
Multiplexing (see preceding text)	10
Store and Forward (1 hop)	20
Error Control (see preceding text)	120
	<hr/>
	210 ± 20 ms

Eliminating the multiplexing and error control delays which are peculiar to this experimental system, leaves a total internal delay of 80 ± 20 msec.

The total end-to-end delay encountered for transmitting encoded voice in the experimental system consists of the above internal network delays, plus the delays associated with the previously described protocol implementation; these were a fixed one frame (22.5 ms) delay to detect vocoder activity, and a variable zero to one frame (0 to 22.5 ms or 11.25 ± 11.25) "meshing" delay (recall Figures 5.6 and 5.7). Thus the total delay is $243.75 \pm 31.25 \approx 244 \pm 31$ ms (or 114 ± 31 ms without the multiplexing and error control delays).

Delay measurements by software means using the 9.6 kb/s system clock gave results within the range just indicated. These measurements were done using only one of the nodes in Figure 5.8 with the 9.6 kb/s link looped back. (The internal delays expected to be encountered are still given by Figure 5.11).

5.4 Emulation of Admission Delays

5.4.1 Admission Delay Model

Actual admission delays will not be encountered in the experimental network, until more nodes are introduced (with each supporting synchronous terminals), and until the flow control scheme is implemented. Since the network is not yet developed to this stage, the admission delays arising from many terminals competing for channel capacity and/or reservations, were emulated in software. The objective, from the point of view of perception, was to obtain at least a gross indication of the susceptibility of voice transmission to the presence of variable (admission) delays. For this reason the emulation is based on a model which is similar to, but basically simpler than the admission delay model described earlier (see Figure 4.21).

The simplified model is shown in Figure 5.12, in which T_1 is considered to represent principally, the mean time to find sufficient channel capacity on the first link in Figure 4.20. Similarly, T_2 is considered to be mainly T_1 plus the mean time to locate reservations. For the purposes of the emulation, and to maintain simplicity, the other relatively small mean times (to request and deliver reservations etc.) in Figure 4.21, are considered to be lumped in with T_1 and T_2 .

Thus the two service mechanisms in Figure 5.12 each have service times based on a single pdf. The pdf's are both

FIGURE 5.12 SIMPLIFIED MODEL USED
FOR THE EMULATION OF
ADMISSION DELAYS

truncated geometric. A discrete pdf is appropriate since admission delays occur in integer numbers of service times (one service time is 40 ms in the experimental network). Geometric pdf's were selected since they are discrete and (apart from the truncation) retain the memoryless behavior of message interarrival and service times assumed earlier for the model in Figure 4.21. The pdf's were truncated in order to limit the peak value of emulated delay. This was desirable in order to realize fairly large average delays and at the same time avoid overflow of the node input buffer. The other justification for truncating the pdf's is that ultimately in TI-NET, it is expected that voice traffic will be given priority over other synchronous traffic; this will tend to minimize the instances in which voice messages will be subjected to large delays.

The pdf's with means T_1 and T_2 are derived in Appendix B.4, and are respectively:

$$g_1(k) = \frac{(1-\rho_1)\rho_1^k}{1-\rho_1^{L_1+1}}, \quad (5.1)$$

and

$$g_2(k) = \frac{(1-\rho_2)\rho_2^{k-1}}{1-\rho_2^{L_2}}, \quad (5.2)$$

where k indexes the emulated delay in service times, the L 's are the number of service times at which the pdf's are truncated, and the ρ 's are parameters (between 0 and 1)

related to the means of the untruncated pdf's, given by (B.4.2) and (B.4.9) in the appendix. Also derived in the appendix are the means for $g_1(k)$ and $g_2(k)$. Thus the overall pdf for the model in Figure 5.12 is

$$g(k) = \alpha_1 g_1(k) + \alpha_2 g_2(k) \quad (5.3)$$

and the overall mean emulated delay is

$$T = \alpha_1 T_1 + \alpha_2 T_2 \quad (5.4)$$

It is noted that $g_2(t)$ is a shifted (as well as truncated) geometric. This reflects the intention of subjecting those messages which are routed through the lower path of the model in Figure 5.12, to at least one service time of admission delay; i.e. it is assumed that for such messages, it will take at least one service time to find reservations. Also, the overall proportion of messages not subjected to admission delay is from (5.1) and (5.3)

$$\alpha_1 g_1(0) = \frac{\alpha_1 (1 - \rho_1)}{1 - \rho_1^{L_1 + 1}} \quad (5.5)$$

An example of (5.1), (5.2) and (5.3) is shown in Figure 5.13. A service time of 40 ms is used to evaluate the absolute mean and truncated values of emulated delay. The truncated areas are given by the inverse of the normalizing factors in (B.4.3) and (B.4.10).

Thus in the experimental work described later, values

FIGURE 5.13 EXAMPLE OF pdf's FOR THE EMULATION OF DELAYS

of emulated delay were drawn (one per message) from a pdf, such as $g(t)$ in Figure 5.13, for which the parameters α_1 , T_1 , T_2 , L_1 and L_2 were specified at run time. However these parameters when computed from the samples drawn, only agree closely with those specified, for a large number of samples. In the experiments, a relatively small number of about 50 to 70 samples (corresponding to as many messages) were involved per test, so that based on the samples, α_1 , L_1 and L_2 tended to be somewhat smaller, and T_1 and T_2 somewhat larger than specified. In what follows only the parameter values based on the samples are referred to.

5.4.2 Emulation Implementation.

Delays were emulated in software in one of the TI-NET nodes in Figure 5.8. Input parameters specified at run time are the mean and truncated values of delay for each path of the model in Figure 5.12, and the proportion (α_1) of messages to pass through the top path of the model. Emulation of delays is accomplished in the following way. The emulated delay for a particular message takes the form of a random number (k) of service intervals (40 ms each), and is generated at the instant the previous message is completely serviced (multiplexed). The generation of k involves the following steps:

- A random number between 0 and 1 (Fortran Library Random Number Generator) is compared to α_1 to determine which path of the model in Figure 5.12 will generate the delay.

- A second similar random number then represents the CDF for the delay in the path just selected (see (B.4.6) and (B.4.12)).
- Using this CDF and the input parameters, k is then computed using either equation (B.4.7) or (B.4.13), depending on the path selected.

Thus at the first service time at which a message is in a position to be serviced, k is checked. If k is zero, the implication is that a reservation, as well as sufficient capacity on the first link in Figure 4.20 are available, and at that point servicing of the message begins. However, if k is greater than zero, the implication is that either one or both of the above resources are unavailable; therefore k is decremented and the message is ignored until the next service time when k is checked again, and the same procedure is repeated.

Eventually, k will be decremented to zero at which time resources are considered to have become available, and servicing of the message begins. It is evident that while k is being decremented, the message - as it arrives - must be stored; since the emulation only involved one terminal (VADAC II), the node input buffer for that terminal was used for this purpose. (While in TI-NET a separate common queue is used [56] - see Figure 4.5).

Thus the emulated delays may alter the message interarrival times as illustrated by the simplified example in Figure 5.14(a). In this figure the interarrival time

FIGURE 5.14(a) EFFECT OF EMULATED DELAYS: $D1 < \tau$

FIGURE 5.14(b) EFFECT OF EMULATED DELAYS: $D1 > \tau$

between the two consecutive messages is seen to be altered by the difference between the emulated delays experienced by each message ($D2-D1$); in general this may cause τ to either increase or decrease. In this example, the emulated delay for the first message $D1$, is less than the interarrival time τ , so that there is no "interference" between the two messages. Figure 5.14(b) is an example where $D1 > \tau$, in which the second message is seen to experience not only the emulated delay $D2$ but also an "interference" delay $D1 - \tau$.

In general therefore, the admission delay incurred by a particular message may be somewhat larger than the emulated delay, because of the interference mechanism just described.

The amount of delay to be emulated was based mainly on the fact that delays encountered (and dealt with by smoothing algorithms) in other research on packetized voice transmission, appear to have variations of the order of 200 to 500 ms [23,25] (99% of the time). Although delay variations as large as this are not anticipated in TI-NET, it was possible to emulate delays having variations of this order; the limitation is the node input buffer size which was extended to accommodate the equivalent of 840 ms of delay without overflowing.

In addition to the emulation of delay, a feature specifically related to reservations was implemented. This feature is analogous to the central node in Figure 4.20 anticipating the need for reservations for voice traffic at

the entrance node, and therefore occasionally providing reservations to that node. Probably the simplest strategy by which the central node decides to provide reservations, is to do so at fixed time intervals corresponding for example, to the mean voice message interarrival time. This "fixed" strategy was implemented and is suitable in terms of the controlled conditions under which the experimental work (described later) was done. Thus at every fixed (programmable) time interval, one reservation is put into a "pool" to be accessed by the voice terminal if the need arises.

It is evident that for long idle times between speech messages, an excess of reservations may build up in the pool. This is analogous to reserving an excess of channel capacity which is not being properly utilized. For this reason a (programmable) limit is set on the maximum pool contents. On the other hand, a sequence of messages with small interarrival times, may deplete the pool; however such is the nature of this simple strategy.

In an actual system, a common pool of such reservations could exist at each TI-NET entrance node, and would be shared by the voice terminals at that node. If the strategy by which the central node provided reservations required a limit to be put on maximum pool contents, this limit would correspond to a compromise between minimizing unused reserved capacity, and providing acceptable delay performance for voice traffic.

5.4.3 Experiment Description and Results

The experimental system used to examine the effects of admission delays on voice transmission is shown in Figure 5.15. The emulation described previously is implemented in the TI-NET node (see Appendix B.1 for documentation). In order to compare the effects of different amounts of admission delay etc., it was necessary to have a source of reproducible speech with which to excite the VADAC voice input. For this purpose a 2½ minute taped session of male monologue speech was used. Details related to this tape and to adjustment of the VADAC input voice level are given in Appendix B.4. The facility to monitor and record the received synthesized voice, after being subjected to admission delays, is provided as indicated in Figure 5.15.

At this point it is appropriate to examine the purpose and significance of hangover more closely. Recall that with regard to encoded voice transmission in TI-NET, channel capacity and reservations are obtained on a message basis, and different admission delays may be incurred for successive voice messages [48,56]. The perceptible effects of variable admission delays, as well as the overhead required to obtain reservations, are therefore minimized by the use of a hangover after each active period in speech, to ensure that once channel capacity and/or reservations are secured, they are not prematurely released. The hangover extends active periods of speech to absorb short pauses (e.g., especially

FIGURE 5.15 EXPERIMENTAL SYSTEM FOR EVALUATING THE EFFECTS OF ADMISSION DELAYS

those preceding stop consonants) and thereby forms longer continuous speech messages. This has the desirable effect of reducing the number of messages and thereby ensuring a more continuous flow of speech. On the other hand, the hangover frames reduce the efficiency of transmission by utilizing channel capacity during part of the idle time.

To obtain some indication of the tradeoff between the reduced number of messages and the increased utilization, the VADAC 2.4 kb/s digital output was analysed by software means, when the VADAC voice input was excited by the 2½ minute taped session mentioned earlier.

The results are shown in Figure 5.16. Thus the number of messages decreased from 444 with no hangover to 51 with 8 frames of hangover (a factor of 8.7 to 1), while correspondingly, the utilization increased by 21.8% (from 56.6 to 78.4%). Note, that the increase in utilization is for a monologue, and it is expected that the corresponding increase for a dialogue would only be half as much, or 10.9%. The reduction in the number of messages has nearly levelled off at a hangover of 6 frames; this suggests that most short pauses have been bridged, and that further significant reduction in the number of messages is only likely if the hangover is increased to the point where longer idle periods between phrases are bridged. Therefore a hangover of 6 to 8 frames is all that is warranted for this example, in order to eliminate short messages (recall Figure 5.6).

FIGURE 5.16 EFFECT OF HANGOVER

A hangover of 8 frames (180 ms) was used in some of the experiments. In addition, a hangover of 5 frames (112.5 ms) was used for which the number of messages was somewhat more than for 8 frames of hangover. This permitted an examination of the effects of admission delay for two different hangovers. For comparison it is noted that hangovers as high as 200 and 300 ms have been used in other work [27,43]. Before leaving this subject, it is mentioned that the mean active and idle period durations corresponding to Figure 5.16 may be determined directly from that figure as indicated in Appendix B.5.

Also tabulated in this appendix are numerous other parameters related to the emulation, such as: α_1 , T_1 , T_2 , T (see Figure 5.12 and (5.4)) and the percentage of messages not encountering emulated delay (see (5.5)). In addition, the following relevant information is computed from measurements and tabulated: the average message arrival rate (λ), the mean (N_q) and peak number of messages in the input buffer, and the mean time spent in the input buffer (T_q). The last parameter is of importance, since the admission delay incurred as a result of the emulation consists of not only the emulated delay, but also the delay due to "interference" as illustrated previously in Figure 5.14. Since it was not straightforward to measure the average admission delay directly, λ and N_q were used together with Little's Result ($T_q = N_q/\lambda$) to estimate the mean time spent by messages in the input buffer [114]. The results are plotted against emulated delay in Figure 5.17.

FIGURE 5.17 MEAN TIME MESSAGES SPENT IN
THE INPUT BUFFER VERSUS
EMULATED DELAY

At this point it is appropriate to recall Figure 4.5 in which the TI-NET entrance node has an input buffer for each terminal (where entrance delay is incurred), and a common queue for messages encountering admission delay. However, since only one terminal is involved in the emulation experiment being described here, the input buffer also serves as a queue. Consequently T_q in Figure 5.17 consists of admission delay plus entrance delay (actually that small average portion of entrance delay not absorbed in the admission delay).

Thus Figure 5.17 represents the total average delay incurred by messages before transmission, as a function of average emulated delay. The crest factor (peak to average ratio) for the emulated delays was maintained at about 3 or more (by suitable adjustment of L_1 and L_2). The results are similar for hangovers of 5 and 8 frames (and α_1 held nearly constant at $\approx .73$ and $.72$ respectively). However, based on peak input buffer contents (see Appendix B.5), it is expected that peak admission delays were somewhat larger for a hangover of 5 frames.

In general, peak delays incurred by messages were much higher than the averages (T_q) in Figure 5.17, as evidenced by the fact that input buffer overflow occurred for the two largest average delays indicated; i.e. the corresponding peak emulated delays were 680 μ s, but the interference mechanism increases these substantially and to the point where

overflow of the '840 ms' buffer occurred. Details of the mean and peak buffer contents versus average emulated delay are included in Appendix B.5.

Also included in Figure 5.17 is the effect of decreasing α_1 for fixed T_1 , T_2 , L_1 and L_2 , and for 8 frames of hangover. Since more messages were now routed through the lower path of the model in Figure 5.12, both the average emulated delay T , and T_q increased as shown. However comparing T_q for different α_1 at a given emulated delay, indicates that T_q reduced as α_1 reduced. This is attributed to the fact that when α_1 was reduced, L_1 and L_2 were held fixed, which in turn is expected to result in less interference delay, and thereby account for the results.

The main purpose of the emulation experiments was to obtain at least a gross indication of the susceptibility of voice transmission to the presence of variable delays. Thus the 2½ minute taped session referred to previously was played for each of the emulated delays indicated in Figure 5.17, and the results were recorded.

Before describing these results, some comments are in order regarding the subjective aspects of the experiments. A full scale detailed subjective evaluation would involve a large number of suitably trained test subjects, evaluating an extensive amount of speech material under controlled conditions; this is a major undertaking in itself and is beyond the scope of this thesis. The evaluation done here is considered to be

preliminary, and was therefore conducted on a smaller scale. One difficulty with regard to the subjective tests arose from the fact that vocoded speech (from a VADAC II speech processor [70]) was used. This was not so much a problem for the author who had experienced considerable exposure to this form of speech, was fairly familiar with its peculiarities, and therefore "adequately trained" in this regard. On the other hand, "untrained" subjects tend to be overcome to some extent by the vocoder speech peculiarities, such as its somewhat unnatural "machine" like quality, and the perceptible effects of its failure to always accurately and consistently extract pitch, and make correct voiced/unvoiced decisions. This became evident when two test subjects (the author's wife and a colleague - on separate occasions), were first briefly exposed to vocoded speech, and were then asked to listen to the prepared tape and to indicate the number of instances in which they judged delays to be perceptible. Initially there was a distinct tendency for each subject to record numerous events which were merely perceptible differences between normal and vocoded speech, rather than perceptible differences between vocoded speech with and without emulated delays. Thus further exposure to vocoded speech, as well as instructions and specific examples as to "what to look for" were necessary, and this constituted informal "training" of the test subjects.

The results obtained from each subject (including the author) showed the same general trends, although in most

cases specific numbers of perceptible delays were different, and in most of these cases, the author tended to perceive more instances than the other two subjects. Figure 5.18 indicates the overall worst case number of instances in which delay was judged to be perceptible for hangovers of 5 and 8 frames. It should be noted that some instances indicated were judged to be "definitely" perceptible while others were judged to be "questionably" perceptible, and may have very well gone unnoticed by a casual listener (although in this case it would be difficult for a listener to be "casual", unless that listener were familiar with vocoded speech). In this sense the results in Figure 5.18 are considered to be somewhat conservative.

Although the average delays encountered (T_q in Figure 5.17) were approximately the same for hangovers of 5 and 8 frames, the frequency of perceptible delays was higher for a hangover of 5 frames. This result was expected (in a qualitative sense) based on the purpose and significance of hangover discussed previously. I.e., the reduced hangover now exposes the silent periods within some words to the possibility of incurring different admission delays. For example (see tape transcript in Appendix B.4), certain two syllable words such as 'output', were observed to be separated as follows: 'ou-tput'. Similarly, the three syllable word 'adaptive' was separated as follows: 'a-da-ptive'. It is seen that separation of syllables takes place as a result

FIGURE 5.18
PERCEPTIBLE EFFECT OF
ADMISSION DELAY

of unbridged pauses preceding stop consonants like t, d and p. Depending on the difference in admission delay encountered by the syllables, the effect may be quite perceptible.

Other work - on the modification of significant within-sentence silent intervals - suggests that variations of the order $\pm 50\%$ of the silent interval are tolerable [44]. Since such intervals are of the order of 100 ms or more, this corresponds to variations of about ± 50 ms. In the work described here, specific delays were not associated with specific intervals, although in general messages were subjected to delays covering and exceeding the range indicated above. Thus even for the lowest average delay of about 80 ms in Figure 5.18, three delays were judged to be perceptible (but quite tolerable and certainly not objectionable).

It is noted that for the highest average delays shown in Figure 5.18, the number of instances of perceptible delay for 5 and 8 frames of hangover, include 3 and 2 buffer overflows respectively as indicated; these were included on the grounds that if the buffer had been bigger, (thereby preventing overflow) the large peak delays (responsible for the overflows) would probably have been perceptible. The overflows themselves were perceptible since the missing information temporarily disrupts the continuous frames structure of the digital signal presented to the VADAC receiver, thereby causing it to lose frame synchronization. (Additional processing in the external link protocol implementation - i.e. in the transmit to VADAC routine described earlier -

may prevent frame loss. However, in the implementation as it stands, the only impairment, apart from the overflowed - missing - information, is reframing of the VADAC receiver which is specified to take place in $\frac{1}{4}$ second or less). It is emphasized that the delays which produced the overflows, are considered to be substantially larger than those normally expected in TI-NET.

Also included in Figure 5.18 is the effect of decreasing α_1 for fixed T_1 and T_2 and for 8 frames of hangover. Since this causes more messages to be routed through the lower path of the model (Figure 5.12), there is a higher incidence of longer delays which are reflected perceptually as indicated. This happens even though for a given emulated delay, T_q decreased with decreased α_1 in Figure 5.17. This indicates that T_q by itself provides a poor indication as to how susceptible delays may be.

Although the results in Figure 5.18 are based on a relatively small data base, they indicate that fairly large variable delays can be accommodated provided a sufficiently long hangover is used. I.e., with 8 frames of hangover, average delays of the order of 150 ms (with peak about 3 times this amount) were judged to be imperceptible.

Another experiment was performed which relates to a TI-NET central node occasionally providing reservations for voice traffic to an entrance node. This feature was implemented in the node shown previously in Figure 5.15. In this

experiment, reservations were provided at fixed time intervals corresponding to: the mean interarrival time of voice messages, twice this time, and three times this time. Therefore the effect of varying the rate at which reservations were provided could be examined. The maximum contents of the pool which collects the reservations provided, was restricted (somewhat arbitrarily) to three reservations; since only one terminal was involved in the emulation, it was felt that this would permit a reserve of reservations which was not unreasonably high or low.

The results are shown in Figure 5.19; they are for a hangover of 5 frames, and the largest emulated delay in Figure 5.17 corresponding to this hangover. At this stage it is pointed out that reservations received from the central node are not effective in cases where messages are delayed only as a result of insufficient capacity on the first link in Figure 4.20. However, because of the simplified model (Figure 5.12) used for the emulation, it was not possible to uniquely associate a particular emulated delay with a particular cause; i.e., a delay may be due to a lack of capacity on the first link, a lack of reservations, or both. Thus for the purposes of the experiment, in order to emphasize the potential benefits of the central node reservation feature, and at the same time retain simplicity, it was assumed that all emulated delays were the result of a lack of reservations. This is analogous to the plausible situation where the entrance link

FIGURE 5.19

POTENTIAL EFFECT OF THE CENTRAL
NODE RESERVATION SCHEME

HANGOVER = 5 FRAMES

 $\alpha_1 \approx 0.74$ TOTAL MESSAGES ≈ 69 $T = 215$ ms — SEE EQN. (5.4)

in Figure 4.20 is lightly loaded, but the exit link is heavily loaded.

The results in Figure 5.19 show, that as the rate at which central node reservations are provided is increased from zero up to the message arrival rate (λ), more messages make use of these reservations. Consequently, the average admission delay is reduced as indicated. The number of instances of perceptible delay also reduces as shown. Other results included in Appendix B.5 show corresponding reductions in peak and average input buffer contents. It is also mentioned that when the reservation arrival rate was set equal to the message arrival rate, there were 13 (out of a total of about 69) instances when the reservation pool was found empty, and 13 instances when it was found full (containing 3 reservations). The interpretation is that there were potentially about 13 messages which had to request reservations, and at least 13 messages for which there were more than enough reservations. For the lower reservation arrival rates, the pool was never found full, but was often found empty. Similarly it is expected that if the reservation arrival rate was made sufficiently greater than the message arrival rate, the pool would often be found full, but never empty.

5.5 Transmission on an Experimental Satellite Link

5.5.1 Preliminaries

Aspects of the TI-NET integrated terrestrial and satellite subnet architectures have been discussed in the TI-NET

overview (Chapter 3) [4,48]. In the present section, experimental work which constitutes a first step in the development of the above systems, is presented [49]. The basic element in the above systems is the 12-14 GHz satellite link, and much of the work which follows is related to studying such a link from the point of view of transmission errors, as well as propagation delays.

The study was done in terms of TI-NET system requirements and operation. For example, satellite propagation delays are of concern for digital transmission in general because of the need to establish and maintain synchronization among terrestrial nodes [127]; this is particularly true in TI-NET which operates in an inherently synchronous manner. Also, error performance of the satellite link is interpreted in terms of the public packet structure used in TI-NET, and in terms of the error control scheme (which is based on terrestrial retransmission [136] and preserves TI-NET transparency in time as much as possible).

The approach taken in the experimental work, was from the point of view of satellite extension nodes, although most of the results were expected to be applicable to the more involved satellite subnet as well. The topology associated with satellite extension nodes was discussed previously and presented in Figure 3.4. Although topologically, the experimental arrangement described later, does not correspond

precisely to Figure 3.4, it did serve to expose the TI-NET node to the essential 'elements' of the satellite link, those being the impairments due to atmospheric attenuation, and the propagation delay and its variation. As such, the experimental system proved to be a useful vehicle for studying the feasibility of a satellite subnet and satellite extensions nodes for the TI-NET system, as well as error control based on retransmission for this system.

5.5.2 Experimental Satellite Link Description

The transmission of data and encoded voice was demonstrated in the experimental two-node terrestrial network described previously in connection with Figure 5.8. For the purposes of the experimental work described here, one of the above mentioned nodes, with appropriate software modifications (See Appendix B.1 for documentation), was used to set up a 9.6 kb/s link from Carleton University's Wired City Laboratory [55] in Ottawa, over the 12 to 14 GHz Hermes (CTS) satellite [128,129] to NASA AMES Research Center in California, where it was looped back and returned to Carleton as shown in Figure 5.20. A full duplex audio channel was used for this purpose, and a high performance "CODEX 9600" modem [130] transmitted the data and encoded voice from the TI-NET node over the 9.6 kb/s satellite link, and received it at the same node.

In the satellite terminal equipment detailed in Figure 5.21, the audio channel encoder/decoder uses sampling and

FIGURE 5.20 SATELLITE EXTENSION NODE EXPERIMENT

FIGURE 5.21 DETAILS OF CARLETON SATELLITE TERMINAL EQUIPMENT

8 bit linear quantizing at a video line rate of 15.75 KHz [131]. The resulting digitized (CODEX) modem signal is then multiplexed into a video data stream during the horizontal sync pulse time. Differential encoding/decoding is used to resolve phase ambiguity in the QPSK modem's recovered carrier [132]. Rate 1/2, constraint length 7, convolutional encoding with soft decision Viterbi decoding is used as the means of forward error correction [133,134]. The QPSK modem accepts a 16 mb/s composite digital signal of multiplexed audio and compressed video information, and outputs a modulated 70 MHz carrier into IF and RF equipment with 85 MHz bandwidth. The Carleton earth station has a 2.4 meter antennae (transmit and receive gains ≈ 48 and 47 dB) with uplink transmit power limited to about 12 watts. The NASA AMES earth station has a 3 meter antennae (transmit and receive gains ≈ 50 and 49 dB) with uplink transmit power capability of up to 125 watts and higher.

Also shown in Figure 5.20 is a "terrestrial" link and another computer (PDP-11/40) at Carleton, which together were used to simulate a terrestrial link between an extension node and a TI-NET entrance node; this "terrestrial" link in Figure 5.20 could be operated synchronously at 2.4, 4.8 or 9.6 kb/s. To put the experiment into perspective, it is pointed out that in an actual system the Carleton location would be the TI-NET extension node, while the NASA AMES

location would be the TI-NET entrance node (or vice-versa); these two nodes would be joined by a terrestrial link to give the topology shown in Figure 3.4. However, because of time and equipment limitations, only a loop back existed at NASA AMES so that the Carleton location effectively represented both the TI-NET extension and entrance nodes.

Software modifications to the existing terrestrial node (Figure 5.8) included disabling its error control features, and implementing the means to demonstrate the "terrestrial" retransmission scheme indicated in Figure 5.20. In addition, node synchronization was accomplished as follows. In the experimental system, MP's are formed and transmitted every 20 ms. In step with these transmissions is a scheduler which expects to process received MP's every 20 ms (i.e. recall Figure 5.11(e)). Hence synchronization consists of ensuring that MP's are received in step with the scheduler. In the experiment described here, the first received MP arrival (determined by the round trip transmission delay over the satellite link) was not (and in general will not be) in step with the scheduler. Thus synchronization was achieved by automatically buffering received MP's by the amount required to put them in step. In general the buffering delay will be from 0 to 1 service interval (a service interval is equivalent to 1 synchronous plus 1 asynchronous MP time and is 40 msec in this experiment). This synchronization method is capable of tracking propagation delay variations due to daily satellite motion, which are normally less than 1 msec [135],

but evidently may be as much as a few msec for the satellite being used here [129].

Before describing the experiments in more detail and presenting the results, it is mentioned that development and testing of the experimental system in Figure 5.20 was made possible by emulating both transmission errors and propagation delays in the laboratory. Details are included in Appendix B.6.

5.5.3 Significance of the Experiments

By way of summary, it is mentioned at this point that experimental results were obtained pertaining to:

- packet error rates versus uplink powers under various atmospheric conditions,
- density of packet error free interval lengths (EFIL's),
- densities of the number of bits in error and the lengths of bit error free intervals (EFI's) within packets, and
- satellite propagation delays.

The significance of the experiments can be appreciated when certain of the above results are interpreted in terms of TI-NET system operation [49]. For example, the density of MP EFIL's is of importance when considering an error control scheme for satellite links based on terrestrial retransmission. However, there are two other factors involved in determining the improvement in error performance offered by such a scheme. First, since the terrestrial link used to

carry NAK's and retransmissions would normally have a much lower speed than the satellite link, MP errors occurring closer together than the terrestrial NAK and retransmission times would create a queue of NAK's and of retransmissions. (However it is expected that the duration of NAK packets would be small enough that the queue of NAK's - if one exists - would be negligible compared to that of retransmissions).

Secondly, to maintain the continuity of a customer's synchronous message, error control based on retransmission, requires buffering (and hence delaying) such messages in the exit node to be able to supply data to the synchronous terminals during retransmissions (e.g. recall Figures 4.2(j) and (k), and 5.11(f) and (g)). The number of queued up retransmissions which can be accommodated therefore, depends directly on the amount of exit node buffering provided. For example, if the amount of time required to empty the queue exceeds the exit node buffering delay time, then a certain percentage of retransmissions will be received too late.

Figure 5.22 illustrates the terrestrial NAK plus retransmission delays encountered as a result of an MP being received in error; also shown is the associated exit node (output) buffering delay requirement. In addition, the effect of a second MP in error occurring fairly close to the previous one, is seen to introduce a queueing delay, as a result of the previous retransmission still being in progress; therefore the output buffering delay requirement is increased as indicated.

FIGURE 5.22 ILLUSTRATION OF NAK PLUS RETRANSMISSION DELAYS AND ASSOCIATED OUTPUT BUFFERING DELAYS

The number of retransmissions received too late can be reduced by increasing either the terrestrial link speed, or the amount of exit node buffering, or both. A useful ultimate objective therefore, could be to minimize terrestrial link speed (communication costs) and exit node buffering (storage delays and costs) subject to the constraint of achieving acceptable error performance. This requires a knowledge of MP EFIL's.

With regard to the number of bit errors and lengths of bit error free intervals within packets, consider briefly TI-NET packet structure and error control. It has been mentioned previously that error control based on retransmission requires exit node buffering for synchronous messages. Also, the number of possible synchronous MP retransmissions per message depends directly on the amount of exit node buffering provided. On the other hand, there is no need to maintain the continuity of an asynchronous message, and hence no need for exit node buffering. Therefore in general, more asynchronous MP retransmissions per message are possible than for synchronous MP's. In the case of synchronous traffic, if an error condition still persists after the maximum number of synchronous MP retransmissions has occurred, the policy in TI-NET is to supply customers with the contents of an MP in error. This policy is consistent with the packet structure used in TI-NET - i.e., public "multi-user" packets containing many user "minipackets" - and is adopted on the basis that perhaps only one, or a small number of users will be affected by the errors in the MP, and will have to NAK using their own protocols.

One way in which a transmission error, or errors, could potentially affect many mp's in asynchronous MP's, is if an mp header bit ("1" indicates data, "0" indicates no data in mp data field) is hit; this would prevent the proper processing of that and all subsequent mp's in the MP. The mp header is desirable for bursty asynchronous traffic and is used to prevent continual reallocation of asynchronous capacity to other users between bursts, which would be costly in terms of asynchronous MP overhead. Therefore the policy with regard to asynchronous MP's is to depend on retransmission to rectify the above situation. On the other hand, for less bursty synchronous traffic for which the number of retransmissions is more limited, it is possible to process synchronous MP's using fixed length mp information implied in the receiving node "activity field" [4,49]. This activity field indicates those users that are currently active - i.e. transmitting a message - and is updated using the synchronous MP header. Hence the information required to process a synchronous MP is only exposed to errors during update, rather than continuously as with mp headers.

Another way in which more than one user can be affected is if MP's contain a number of errors distributed among the various mp's. However, if there are very few errors per MP and/or these errors tend to be clustered, then there is a smaller probability of many users being affected. To examine these possibilities, the densities of the number of bit errors, and the lengths of bit EFI's within MP's were measured.

5.5.4 Description of Experiments

The experiments consisted of two parts. In the first, asynchronous MP's of data and synchronous MP's of encoded voice were transmitted alternately every 20 msec over the 9.6 kb/s satellite link and returned. In order to gather data related to the number of bit errors, and the lengths of bit error free intervals within MP's, a copy of the 64 most recent transmitted (as well as the 64 most recent received) packets was maintained in core (the satellite propagation delay of about 510 msec corresponds to about 26 MP's).

In the second experiment, the data and voice terminals were not used but rather, a fixed set of 64 MP's (32 asynchronous and 32 synchronous) was transmitted continuously over the satellite link and received. Node input buffers for the teletype and VADAC II speech processor were prefilled (by software means) with data characters and vocoder frames respectively, so that the contents of the MP's repeated every 64 MP's. To demonstrate error control capability based on retransmission over the "terrestrial" link, a file of 64 packets identical to the 64 above was stored in the PDP-11/40 in Figure 5.20. Hence, when an MP on the satellite link was received in error, it was NAK'd by sending a synchronous/asynchronous indicator and packet number over the "terrestrial" link as indicated in Figure 5.23 (which is analogous to Figure 5.22). The PDP-11/40 used this information to calculate the appropriate MP address, and after simulating

FIGURE 5.23 NAK + RETRANSMISSION DELAYS IN THE EXPERIMENTAL SYSTEM

a terrestrial transmission delay of about 25 msec, "retransmitted" the MP. In this way it was possible to observe the "retransmission" times and number of "retransmissions" missed as a function of "terrestrial" link speed and packet error rate. Figure 5.23 illustrates the NAK plus "retransmission" delays encountered for the various "terrestrial" link speeds tested.

Throughout the experiments just described, Carleton uplink power was held constant at about 12 watts, while the power at NASA AMES was varied from about 20 to 70 watts depending on atmospheric conditions and in order to achieve various packet error rates. The variables during the experiments were therefore the NASA AMES uplink power and atmospheric conditions (it is also mentioned that an increase in uplink power causes increased satellite downlink power in the same direction of transmission, but suppressed downlink power in the reverse direction; this is done to maintain constant total downlink power). Because of limited time available to perform the experiments, and the desire to gather data related to characterizing satellite link error performance, the NASA AMES uplink power was usually adjusted to result in fairly high packet error rates.

5.5.5 Experimental Results

5.5.5.1 Packet Error Rates, Uplink Powers and Atmospheric Conditions

Figure 5.24 summarizes MP error rate versus NASA AMES uplink power measurements made under various atmospheric

FIGURE 5.24

MP ERROR RATE VS NASA AMES UPLINK POWER
FOR VARIOUS ATMOSPHERIC CONDITIONS
(CARLETON UPLINK POWER = 12.4 WATTS)

conditions. The results shown were obtained between 6 and 8 p.m. E.S.T. on each of 5 days in April 1977 as indicated, and in addition are labeled as to percent relative humidity (R.H.), cloud cover, precipitation and wind velocity at Carleton at 6 p.m. E.S.T. It is expected that weather conditions at NASA AMES had little or no effect on the results since the looped back link was such that most if not all errors occurred on the NASA AMES to Carleton hop. Test durations - points in Figure 5.24 - varied from 3.4 to 17.4 min. with the mean 9.9 min.

The results indicate that light snow had the most significant effect on signal absorption (April 7th), whereas light rain appeared to have little effect (April 13th) when compared to results with no precipitation (April 12th). On two other days with no precipitation (April 20th and 21st), the absorption was evidently greater than on the light rain day; however, it was overcast on both of these days and it is expected that cloud cover thickness was such as to account for the results observed. Also observed on April 20th and 21st was a tendency for the MP error rate to change with time for the same uplink power, or alternatively for different uplink powers at different times to give similar MP error rates. It is expected that the overcast condition together with wind velocity could account for these results by changing cloud cover thickness, and hence signal absorption, in the reception area.

It is mentioned that MP errors were detected using a CRC 16 block check over 18 characters (144 bits including CRC 16) so that a lower bound on the bit error rate (with forward error correction and assuming 1 bit in error per MP) may be obtained by dividing the ordinate in Figure 5.24 by 144 [137]. However, this is optimistic, since as results described later will show, there are an average of about 3 bits in error per MP.

5.5.5.2 Packet Error Free Interval Lengths

In the results discussed presently, MP EFIL's are interpreted in terms of terrestrial NAK^A plus retransmission delays (which correspond to terrestrial link speeds as illustrated in Figure 5.23). The results are also interpreted for the special case where only one retransmission can be accommodated before the exit node buffering delay is exceeded; i.e. a queue of retransmissions is not considered. In addition, the effect of errors on the terrestrial link is not included in this work.

A density of MP EFIL's for a high MP error rate is shown in Figure 5.25 as an example. Typical of densities at high MP error rates (see Appendix B.6 for others), the figure indicates a tendency for MP's in error to cluster. This was not anticipated, but rather, an approximately uniform density of MP EFIL's was expected for a satellite link. The explanation is believed to lie in the forward error correction scheme mentioned previously, which is capable of eliminating

FIGURE 5.25

DENSITY OF MP ERROR FREE INTERVAL LENGTHS
(HIGH MP ERROR RATE: 1.4×10^{-2})

TOTAL MP-EFIS 521 (DENSITY INCLUDES 496)
MINIMUM MP EFIL 0 SEC
MAXIMUM MP EFIL > 4.2 SEC
MEAN MP EFIL 1.4 SEC

% OF MP ERROR FREE INTERVALS WITH LENGTHS
IN THE 0.1 SEC RANGES ON THE ABSCISSA

DISCRETIZED
EXPONENTIAL

0.1 SEC
(5 MP TIMES)

MP ERROR FREE INTERVAL LENGTHS (SEC)

(1 MP = 20 MSEC)

246

independent errors, but fails to correct all errors occurring in bursts.

The dotted curve in Figure 5.25 is based on an exponential pdf with mean equal to that of the data (1.4 sec in this case); (i.e. each dot represents the area under an exponential pdf for the .1 sec range on the abscissa that the dot covers). These results, together with others for high MP error rates included in Appendix B.6, indicate that MP EFIL's may be modelled at least approximately by a "discretized" exponential as just indicated. This in turn enables the performance of an error control scheme based on retransmission to be predicted as will be described later.

Although the clustering of MP's in error is not favourable in terms of error control based on terrestrial retransmission, Figure 5.25 shows that only 6.7% of MP EFIL's are between 0 and 0.1 sec, so that a terrestrial delay of, for example 0.1 sec, would still result in over an order of magnitude improvement in error performance. For lower MP error rates, clustering appears to be reduced as shown, for example in Figure 5.26, where the same delay would enable the retransmission of all MP's in error. However, it is pointed out that for the lower MP error rates, a lesser amount of data was gathered, and therefore the results should be interpreted as being optimistic. The exponential model just mentioned suggests that for the example in Figure 5.26 about 1% of MP EFIL's are between 0 and 0.1 sec; this corresponds to two

FIGURE 5.26

DENSITY OF MP ERROR FREE INTERVAL LENGTHS
(LOWER MP ERROR RATE: 2.0×10^{-3})

TOTAL	MP	EFIS	71	(DENSITY INCLUDES 16)
MIN	MP	EFIL	0.2	SEC
MAX	MP	EFIL	52.9	SEC
MEAN	MP	EFIL	9.82	SEC

% OF MP ERROR FREE INTERVALS WITH LENGTHS
IN THE 0.1 SEC RANGES ON THE ABSCISSA

0.1 SEC
(5 MP TIMES)

MP ERROR FREE INTERVAL LENGTHS (SEC)

(1 MP = 20 MSEC)

orders of magnitude improvement in error performance for a terrestrial delay of 0.1 sec.

All results are summarized in Figure 5.27, where the range of measured MP EFIL's is shown. The "MINIMUM" curve indicates the maximum terrestrial delay (corresponding to the minimum terrestrial link speed) which would ensure retransmission of all MP's in error. For example, at MP error rates $\leq 1 \times 10^{-3}$, a terrestrial delay as high as 0.6 sec would satisfy this condition. Hence for these MP error rates, the data indicates that acceptable error performance is easily achieved, so that a higher terrestrial link speed would be selected simply to reduce the exit node buffering delay; for higher MP error rates however, a higher terrestrial link speed would also be selected to achieve acceptable error performance.

In this regard, Figure 5.28 illustrates the performance of error control based on retransmission in terms of the percentage of "retransmissions" missed as a function of MP error rate and "terrestrial" link speed. The "terrestrial" link speeds are those discussed earlier in connection with Figure 5.20, and the NAK plus "retransmission" delays are those measured (to the nearest MP time) on the "terrestrial" link and agree with those predicted in Figure 5.23.

The results indicate that for MP error rates $\leq 1 \times 10^{-3}$, "retransmission" of all MP's in error was possible, while for

FIGURE 5.27
MP ERROR FREE INTERVAL LENGTHS
VS
MP ERROR RATE

FIGURE 5.28

PERFORMANCE OF ERROR CONTROL BY
TERRESTRIAL RETRANSMISSION FOR
SATELLITE EXTENSION NODES, BASED
ON MP ERROR FREE INTERVAL DATA

PERFORMANCE PREDICTION
BASED ON EXPONENTIAL MP BFIL'S

"TERRESTRIAL" LINK SPEED 2.4 kb/s

"TERRESTRIAL" LINK SPEED 4.8 kb/s

"TERRESTRIAL" LINK SPEED 9.6 kb/s

NAK PLUS
"RETRANSMISSION"
DELAY IN MP'S
(1 MP = 20 MS)

CORRESPONDING
"TERRESTRIAL"
LINK SPEED
(kb/s)

% OF MP "RETRANSMISSIONS" MISSED

10^{-1}

10^{-2}

10^{-3}

10^{-4}

MP ERROR RATE ON 9.6 kb/s SATELLITE LINK

2.4

4.8

9.6

6

4

3

higher MP error rates, a certain percentage of "retransmissions" were missed, and fewer misses occurred for higher "terrestrial" speeds as expected. Recall that these results correspond to the case of only 1 MP retransmission per exit node buffering delay, and in this sense are pessimistic. In other words, being able to accommodate a queue of requests for MP retransmission (corresponding to more exit node buffering) would have reduced the percentage missed. On the other hand, the results are somewhat optimistic in the sense that they do not include the effect of errors on the terrestrial link.

Also, for the lower MP error rates, the limited data indicates performance which is very likely optimistic; the dashed curves in the figure are performance predictions based on the exponential model for MP EFIL's, and suggest the extent to which the measured performance may be optimistic. However, for the higher MP error rates there is reasonable agreement between predicted and measured performance. Thus even at an MP error rate of about 2×10^{-2} and 2.4 kb/s "terrestrial" speed, nearly an order of magnitude improvement in error performance is indicated for the retransmission scheme.

5.5.5.3 Number of Bit Errors and Lengths of Bit Error Free Intervals Within Packets

Densities of the number of bit errors, and the lengths of bit EFI's within MP's were measured; Figures 5.29 and 5.30 are two respective examples, while all results are summarized in Table 5.1. (Note that Figure 5.26 is the density of MP.

FIGURE 5.29

DENSITY OF NUMBER OF BIT ERRORS
WITHIN MP's (HEADER + DATA FIELD + 128 BITS)

MP ERROR RATE *	2.0×10^{-3}
NO. OF MP's ANALYSED	62
TOTAL ERROR EVENTS	180
MIN NO. ERRORS/MP**	0
MAX NO. ERRORS/MP	17
MEAN NO. ERRORS/MP	2.90

* SEE FIGURE 5.26 FOR DENSITY OF MP ERROR
FREE INTERVAL LENGTHS

% OF MP's WITH BIT ERRORS EQUAL TO ABSISSA

30

25

20

15

10

5

0

NUMBER OF BIT ERRORS WITHIN MP's

0

2

4

6

8

10

12

14

16

18

20

** NO ERRORS IN HEADER + DATA FIELD.
≥ 1 ERROR IN CRC 16 CHARACTERS.

FIGURE 5.30

TABLE 5.1

NUMBER OF BIT ERRORS AND LENGTH OF BIT EFIS WITHIN MPs

MP ERROR RATE	NO. MP'S ANALYSED	TOTAL ERROR EVENTS	NO. BIT ERRORS WITHIN MP HEADER + DATA FIELD*			TOTAL BIT EFIS	BIT EFIS WITHIN MP HEADER + DATA FIELD*			
			MIN**	MAX	MEAN		MIN	% MIN	MAX	MEAN
1.2×10^{-3}	39	108	0	7	2.77	72	0	40.3	15	1.57
1.6×10^{-3}	69	194	0	9	2.81	133	0	36.8	9	1.48
2.0×10^{-3} ***	62	180	0	17	2.90	117	0	42.0	6	1.40
3.9×10^{-2}	945	4355	0	65	4.61	3527	0	46.7	89	1.54

* MP HEADER + DATA FIELD = 16 CHARACTERS (128 BITS)

** NO ERRORS IN MP HEADER + DATA FIELD
≥ 1 ERROR IN CRC 16 CHARACTERS

*** SEE FIGURES 5.26, 5.29 AND 5.30.

EFIL's corresponding to Figures 5.29 and 5.30). Recall (Figure 5.27) that mean MP EFIL's were of the order of many MP times; therefore, if bit errors were independent, as expected on a satellite link, the mean number of bit errors per MP in error should be about one. However, results here show an average of about three bit errors per MP which indicates a tendency for errors to occur in bursts. Again, this is believed to be caused by the forward error correction scheme which eliminates independent errors but fails to correct all errors which occur in bursts.

The results for bit EFIL's within MP's show a definite tendency for bit errors to cluster since approximately 37 to 47% of bit EFI's had zero length, and the mean bit EFIL was about 1.4 to 1.6 bits. Hence, these results are favourable in terms of the public packet structure used within TI-NET, in that the strong tendency for errors within MP's to cluster reduces the probability of many users being affected.

Figures for the 1st, 2nd and 4th entries in Table 5.1 are included in Appendix B.6. Also shown in this appendix is a means by which the bit error rate of the received signal can be estimated. This is done by taking into account the QPSK modem performance [132], the improvement in error performance provided by the forward error correction scheme when operating with this modem [133,134], and finally the average number of errors measured per MP. Thus for the example presented in Figures 5.26, 5.29 and 5.30, an MP error rate

of 2.0×10^{-3} corresponds to an estimated bit error rate of 4.2×10^{-5} , after the Viterbi decoder, and 1.2×10^{-2} before the decoder. The associated energy per (coded) bit to (one sided) noise power density ratio (E_b/N_0) is 6.4 dB. Therefore at this value of E_b/N_0 , forward error correction improved the error rate by over two orders of magnitude.

5.5.5.4 Satellite Propagation Delays

Measurements of transmission delays on the 9.6 kb/s satellite link were made (by software means) every 256 service times (about every 10 sec) using the system 9.6 KHz clock, thereby providing a resolution of about ± 0.1 msec. Tests were done on April 7, 12, 13, 14, 20, 21 and 22nd, 1977 at the times, and with the durations stated earlier in connection with Figure 5.24. During each test the current, minimum, maximum and (8 sample) average of the propagation delay was maintained in core. The results are summarized in Table 5.2. The measured absolute satellite propagation delay was found to be from 509.0 to 509.8 msec with mean 509.3 ± 0.2 ms. These results depend on the accuracy of the terrestrial delays listed in Table 5.2; in particular, the delay through the NASA AMES satellite terminal equipment audio channel was assumed the same as the mean of the delays through two similar Carleton channels (which differed by 0.3 msec). For comparison it is mentioned that based on distance measurements provided by NASA AMES, the propagation delay is from 509.4 to 510.4 msec.

The MAX-MIN measurements of propagation delay variations

TABLE 5.2

TRANSMISSION DELAY ON 9.6 kb/s SATELLITE LINK

TRANSMISSION DELAYS (MSEC)						
ABSOLUTE*			VARIABLE			
OVER ALL			PER TEST	PER SESSION ($\approx$ 1 HOUR)		OVER ALL
MIN	MAX	AVE	MAX -MIN	MAX -MIN	MAX AVE -MIN AVE	MAX -MIN
525.8	526.6	526.1	0 to 0.3	0 to 0.6	0 to 0.5	0 to 0.8
						0 to 0.7

* INCLUDES THE FOLLOWING TERRESTRIAL DELAYS (MSEC)

- PROCESSING
- CHARACTER SERIAL TRANSMISSION
- CODEX MODEM
- LOCAL CABLING
- SATELLITE TERMINAL EQUIPMENTS

$\approx$ 0 .8
 14.6
 $\approx$ 0
 1.4

TOTAL 16.8 MSEC

in Table 5.2 were likely influenced to some extent by modem residual timing jitter; however, the (8 sample) average measurements (AVE MAX-AVE MIN) eliminated fluctuations due to jitter. Therefore, the results indicate an overall variation in satellite propagation delay of 0.7 ± 0.2 msec throughout the tests.

5.5.6 A Model for Estimating Error Performance Improvement for a Satellite Link with Terrestrial Retransmission

Experimental results discussed previously suggest that for a satellite link using forward error correction, MP EFIL's may be represented by an exponential pdf, at least for high MP error rates. Based on this experimental evidence, the improvement in error performance offered by terrestrial retransmission may be modelled as indicated earlier in Figures 5.25 and 5.28. This model is now formulated more generally for the case where one retransmission at a time can be accommodated; i.e. a queue of retransmissions is not considered.

Input parameters required for the model are:

R_S = Satellite link speed (b/s)

L_M = MP size (bits)

R_T = Terrestrial link speed (b/s)

L_N = NAK Packet size (bits)

D_T = Round trip terrestrial propagation and other transmission delays (sec)

P_M = MP error rate (MP's in error/total MP's).

From these input parameters, the following parameters are computed:

$$T_E = \text{mean MP EFIL (sec)}$$

$$= \frac{L_M}{P_M R_S}$$

(5.6)

T_T = Total NAK plus retransmission delay encountered on the terrestrial link, expressed to the nearest, highest MP duration. (sec)
(e.g. see Figures 5.22 and 5.23).

$$= \left\lceil \frac{(L_N + L_M)/R_T + D_T}{L_M/R_S} \right\rceil \frac{L_M}{R_S}$$

(5.7)

where the notation $[x]$ means x , if x is an integer, or next largest integer of x , if x is not an integer.

Thus, the density of MP EFIL's may be represented by

$$h(t) = \frac{1}{T_E} e^{-t/T_E}, \quad (5.8)$$

and the proportion of retransmissions missed as a result of the NAK and retransmission delays is:

$$F(T_T) = \int_0^{T_T} h(t) dt = 1 - e^{-T_T/T_E}. \quad (5.9)$$

The interpretation is that for a very small NAK and retransmission delay, $T_T \ll T_E$ (which from (5.7) implies a correspondingly high terrestrial link speed), very few

retransmissions are missed and a large improvement in error performance is realized. On the other hand, for very low terrestrial speeds ($T_T \gg T_E$) most retransmissions would be missed, resulting in little improvement. The improved MP error rate is $P_M \cdot F(T_T)$, and the proportional improvement may be expressed as the ratio of the MP error rates without, and with retransmission; i.e. $P_M / P_M \cdot F(T_T) = 1 / F(T_T)$.

Equations (5.7) and (5.9) will be illustrated by means of the following example:

$$R_S = 56 \text{ kb/s}$$

$$L_M = 560 \text{ bits (MP duration} = 10 \text{ ms)}$$

$$R_T = \text{variable}$$

$$L_N = 6 \text{ characters (48 bits)}$$

$$D_T = 25 \text{ ms}$$

$$P_M = \text{variable}$$

Equation (5.7) is used to compute the output buffering delay incurred at an exit node as a result of the NAK plus retransmission delays, and the corresponding curve is shown in Figure 5.31 for a wide range of terrestrial link speeds. It is seen that for a speed as low as 1.2 kb/s, the output delay required is 540 ms which is of the same order as the delay that would result by NAKing and retransmitting over the satellite link itself; thus, in terms of delay alone, speeds of 1.2 kb/s and lower are clearly impractical, for this example. As the terrestrial link speed is increased, the output delay requirements reduce significantly, until the

FIGURE 5.31

OUTPUT BUFFERING DELAY

VS

TERRESTRIAL LINK SPEED

FOR A 56 kb/s SATELLITE

LINK WITH ERROR CONTROL

BY TERRESTRIAL RETRANSMISSION

higher speeds (>9.6 kb/s) are reached at which point the fixed propagation and transmission delays (25 ms assumed in this example) dominate the total NAK plus retransmission time. Although such high terrestrial link speeds require low buffering delays, they are ruled impractical for this example because of high communication costs (relative to the satellite link). A reasonable compromise appears to be 9.6 kb/s, since it results in a fairly low output delay of 90 ms, and can be supported by conditioned voice grade lines.

Figure 5.32 shows the error performance resulting from the same range of terrestrial link speeds considered in Figure 5.31, and for a wide range of MP error rates. As expected, the percentage of retransmissions missed reduces with increased terrestrial speed, although the amount of improvement is less for the higher speeds (>9.6 kb/s) because of the fixed propagation and transmission delay of 25 ms assumed here. It is pointed out that only part of Figure 5.32 is meaningful in a practical sense: For example a designer may only consider it worth while to implement error control based on retransmission, if an improvement in error performance of about one order of magnitude or better was achievable. This corresponds to missing $\leq 10\%$ of the retransmissions in Figure 5.31. Thus from this figure, for a 9.6 kb/s retransmission speed, an improvement in error performance of at least one order of magnitude is indicated for MP error rates of about 10^{-2} or lower..

4

CHAPTER 6

SUMMARY, CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

6.1 Summary and Conclusions

6.1.1 General

The objective of this thesis has been to investigate certain aspects of transmitting encoded voice with data in a shared network environment. The investigation has been primarily experimental and to some extent analytical, and the shared network, called TI-NET, uses a transparent intelligent approach. The motivation for this approach is to provide efficient sharing of network resources for both voice and data traffic, and at the same time to minimize end-to-end delays, particularly variable delays, to which voice traffic is known to be most susceptible.

The thesis begins by discussing the nature of the problem, which is related to the effects of delay on voice transmission. Large fixed delays, such as those encountered on satellite links within switched telecommunication networks, have a certain psychological effect on users engaged in normal conversations, even though users may not be aware of the presence of delay; the psychological effects are caused by delay induced increases in both the number of occurrences, and in the durations of involuntary interruptions and mutual silences during conversations. A review of previous work in packetized voice transmission has revealed the principal

disadvantage of the conventional packet switching approach, which is the existence of large variable delays. Attempts to smooth these delays at a destination node, result in various perceptible anomalies due to late packets, and re-assembly buffer underflows and/or overflows.

In spite of the above problems, the motivations for research in mixed voice and data transmission are strong; these include the advent of new technology and new voice applications, as well as the economic advantages of exploiting the silent periods in voice traffic, and of developing integrated networks to efficiently handle both voice and data.

Substantial economic benefits through savings in network bandwidth, are also made possible by statistically multiplexing data with vocoded voice. For this reason a brief review of applicable speech analysis/synthesis techniques was presented. An evaluation of these techniques revealed that the digital transmission rate and continuous frame structure of the analyser output, have certain implications in terms of eliminating the idle periods in speech, and coping with such impairments as variable network delays, missing information (due to late and lost packets, and buffer overflows), and transmission errors.

A number of approaches to integrated voice and data networks were reviewed. Although circuit switching has the advantage of fixed and minimal delay (after call setup),

sharing of transmission facilities is only possible on a call basis, and many value added services are not provided. Hybrid (circuit plus packet) switching is a viable and popular improvement over pure circuit switching; however the disadvantages just mentioned also apply to the circuit switched portion of traffic in such hybrid systems. Conventional packet switching has the potential to provide extensive sharing of network resources (apart from overhead considerations), as well as value added services; however such an approach is disadvantageous for voice traffic because of its variable throughput and delay characteristics.

What is most desirable therefore, is an integrated system which combines the useful features of both circuit and packet switching. This leads to a transparent intelligent network approach, a few examples of which, have been reviewed.

The major area of emphasis in this thesis has been (to some extent) in analysing (but primarily in), implementing and evaluating a capability for the transmission of encoded voice statistically multiplexed with data in an experimental network. The focal point for most aspects of the work has been network delays -

particularly variable delays - as they apply to encoded voice transmission.

6.1.2 Analytical Work

The analytical work first summarizes various TI-NET delays which are internal to the subnet. Then by way of introduction to the peripheral or admission delays, some aspects of parallel servicing in TI-NET are discussed. These aspects include the advantage of parallel servicing, and a description of TI-NET entrance node operation; this description also introduces a performance measure called packing efficiency, which has meaning for a non-homogeneous population, and suggests a service discipline such as SFCFS, which has also been discussed.

Following this, a beginning has been made in the analysis of TI-NET admission delays, by dividing the problem into two parts, the first dealing with single hop traffic (not involving reservations), and the second with multi-hop traffic (involving reservations). For the single-hop case, the solution is provided for the pdf of the number of bulk units in an $M/M/\infty$ system with non-homogeneous job population. The motivation for treating the infinite server case was that a closed form solution was easier to obtain, and the results are representative of TI-NET entrance node operation under light load conditions. The more difficult finite server

case is being dealt with by simulation [56,57], and is not a direct part of this thesis. The above mentioned solution is provided for two system representations; the first uses a number of independent parallel service facilities - one for each job type, and the second uses a single combined service facility.

For the parallel facility representation, two approaches - one algorithmic, and the other more classical, have been shown to give the same result. In addition, expressions have been presented for such parameters as traffic balance, load balance and overall average service time. In this regard it is shown that traffic balance may be expressed either in terms of average message arrival rates or departure rates, and that load balance may be interpreted either in terms of the average number of messages of each type in the system (expressed in bulk units), or in terms of the average number of servers required to service messages of each type in the system. Also for assumed equal mean message lengths for all message types, traffic balance and load balance are equivalent. Some useful observations were also made with respect to the finite server case. The parallel representation was also used to obtain the pdf for the number of bulk units in an $M/M/\infty/M$ (closed) system with non-homogeneous job population. The implications of open and closed queueing systems was also discussed from the point of view of voice traffic.

For the single service facility representation, a state description based on a two element vector has been presented; the elements are the number of bulk units in the system k , and a particular substate $B_j(k)$, which represents the j th way of having k bulk units in the system. Since even for moderate k , the number of substates was shown to be very large, a probabilistic approach (and the simplifying assumption of identical mean message lengths for all job types), was used to combine the dependencies on the substates. This led to a compact state transition diagram in which the states are simply the number of bulk units in the system, and in which the departure rates from each state are averaged over all applicable substates. From this state diagram, it is shown that the left hand balance equation is independent of initial conditions and of the above mentioned average departure rates, and that its solution is the same as that obtained with the parallel facility approach.

Other interesting aspects of the compact state diagram are also discussed. One of these is that the expressions for the (proportional) average departure rates used to obtain the state diagram, are not independent of the balance equations for this diagram; in fact the same expressions for these departure rates can be obtained from the balance equations, in a way which implies local balance for each message type (in addition to the global balance for all message types).

It was also shown (again, for identical mean message lengths for all message types) that for a given average number of bulk units in the system, the average proportion of servers used for each message type can be evaluated for any number of bulk units - k , in the system. Furthermore, when k equals the mean number of bulk units in the system, the above proportions of servers correspond closely with the system traffic and load balances, and the agreement improves for larger mean numbers of bulk units in the system. Finally, eliminating the dependency on k by summing over all k , gives overall average proportions of servers, which again agree more closely with the system traffic and load balances for larger average numbers of bulk units in the system.

A first step was made in the study of multi-hop traffic in TI-NET by presenting an approximate model for such traffic. In this model the dependencies of admission delay at a particular entrance node, on numerous other network nodes were lumped in order to eliminate the complexities that would be inherent in a detailed model. The model accounts approximately for the various components of admission delays associated with requesting, locating and delivering reservations to a TI-NET entrance node, as well as the queueing delay associated with finding sufficient free servers on the first (entrance) link.

The parameter of most interest in this model is the mean time to locate reservations, and this parameter was

assumed to vary in accordance with the (queueing component of the) response time of a multiple resource system. For this reason, the behavior of the multiple resource model as it applies to the TI-NET situation was reviewed, and was shown to simplify to a closed multiserver queueing system with finite (and by assumption - homogeneous) population, i.e., M/M/m/M. The probabilities of operating above and below "saturation" in the simplified multiple resource model, were established by assuming a uniform pdf for the total number of terminals at the TI-NET entrance nodes.

Using the resulting admission delay model, three examples of admission delay pdf's for multi-hop traffic were evaluated. The first example showed the separate pdf's for operation above and below saturation, as well as the combined pdf for a particular probability of operating above or below saturation. The last two examples illustrated the combined admission delay pdf's for various probabilities of operating above and below saturation. The first of these is analogous to comparing admission delay performance for different job applications with the same range of terminals, but different terminal think times; on the other hand, the last example is analogous to different ranges of terminals for the same job application (same terminal think times).

6.1.3 Experimental Work

The experimental work began by describing two voice terminals. The first is an LPC voice response system which

was used in the first demonstration of packetized data and voice transmission in TI-NET [47]. The second is a real time, pitch excited digital 16 channel vocoder (VADAC II), which was operated at 2.4 kb/s and used throughout most of the experimental work [48,49].

In order to interface voice terminals to TI-NET in such a way as to take advantage of the idle periods in speech, certain external link considerations were discussed, primarily in terms of the continuous frame structure of vocoder digital outputs. A suitable external link protocol for encoded voice transmission was then presented. Following this, the software implementation of a protocol for interfacing the VADAC channel vocoder to the experimental TI-NET was described. This description included the various functions performed at the source, such as framing, sending an initial frame sequence, active frame detection and the use of hang-over; at the destination, the functions performed are: output of the initial frame sequence, and "meshing" newly arrived messages with previous outgoing idle sequences (because of variable network delays).

The next step involved packetizing and transmitting encoded voice within the experimental network. After a brief description of the experimental system, the implementation of an internal protocol for synchronous data and voice traffic was described. This protocol identifies the end of messages, specifies whether they are voice or data messages, and "pads" any last partially filled "minipackets".

of the messages. An extended version of this protocol which conveys flow control information is also described (but not implemented). Following this, the various internal delays encountered in the experimental network are summarized and discussed.

Since the experimental TI-NET was not developed to the point where "admission" delays occurred due to the sharing of resources (channel capacity and reservations) by many users etc., these delays were emulated in the experimental system by software means. The emulation was based on a simplified version of the model for multi-hop traffic presented in Chapter 4, and was interpreted in terms of TI-NET entrance node operation.

Before the experiments with emulated delay were conducted, the purpose and significance of "hangover" was described in terms of the nature of speech messages, and in terms of the TI-NET flow control scheme. Some favourable experimental results also demonstrated a large reduction in the number of (short) speech messages with increases in hangover; the corresponding moderate penalty in lost efficiency was also illustrated. Some of the subsequent experiments with emulated delay were done with two values of hangover in order to determine the effect of varying this parameter.

One aspect of the emulation is that the actual admission delay incurred by the voice messages was somewhat larger than the emulated delay, because of delay induced interference

(or queueing) of these messages in the node input buffer; therefore the measured average buffer contents (in messages), and the average message arrival rate was used with Little's Result to estimate the actual average admission delay incurred. Other measurements included the peak input buffer contents.

The main purpose of the emulation experiments was to obtain at least a gross indication of the susceptibility of voice transmission to the presence of variable delays. Thus preliminary subjective tests indicated that fairly large average random delays of nearly 150 ms (with peaks about 3 times this amount) were imperceptible for a hangover of 8 vocoder frames (180 ms). For this hangover, most delays occurred predominantly between phrases and sentences, rather than (unnaturally) between syllables and words. On the other hand for a hangover of 5 frames (112.5 ms), the frequency of perceptible delays was considerably higher; this result was expected (in a qualitative sense) based on the purpose and significance of hangover. Another experiment was also performed which involved the emulation of a central node providing reservations to an entrance node at fixed time intervals corresponding to mean message interarrival times (and multiples thereof). The results indicated a corresponding reduction in average admission delay and in perceptible instances of delay as the reservation supply rate was increased.

The feasibility of 12-14 GHz satellite links for use with TI-NET extension nodes was demonstrated by the transmission of data and encoded voice over an experimental link, and by a detailed link error performance study of this link [48,49]. One motivation for including satellite transmission in the thesis was, among others, the fact that using such a facility to access the terrestrial TI-NET system, provides an advantage in terms of small variable delays (as compared to accessing conventional packet switched networks via satellite links); this is particularly important for voice applications.

While the experimental arrangement used did not correspond precisely to the topology of a satellite extension node, it served to expose the TI-NET node to the essential elements of the satellite link, those being the impairments due to atmospheric attenuation, and the propagation delay and its variation.

A detailed description of the experimental arrangement included the TI-NET node, the satellite terminal equipment and a capability for demonstrating error control by retransmission.

Results were obtained pertaining to the following:

- packet error rates versus uplink power under various atmospheric conditions,
- density of packet error free interval lengths (EFILS)
- densities of the number of bits in error, and the lengths of bit error free intervals (EFI's) within packets, and
- satellite propagation delays.

The packet EFIL results were interpreted in terms of the error control by retransmission scheme, or more specifically in terms of improvement in error performance versus terrestrial link speed. Furthermore it was shown that the measured density of packet EFIL's may be modelled by an exponential pdf, at least for high packet error rates. Using this model, agreement between measured and predicted improvements in error performance was demonstrated for high packet error rates (for which sufficient experimental data was available), and for various "terrestrial" link speeds. For example, at a packet error rate as high as about 2×10^{-2} , and the lowest tested terrestrial speed of 2.4 kb/s, nearly an order of magnitude improvement was indicated for the retransmission scheme.

Analysis of the number of bit errors within packets revealed a tendency for errors to occur in bursts, i.e. there were usually about three errors per packet for packet error rates of about 1 to 2×10^{-3} . These results are favourable in the sense that for a given link bit error rate, burstiness results in a lower packet error rate than independent bit errors.

In addition, the results for bit EFIL's within packets showed a tendency for bit errors to cluster, since about 37 to 47% of bit EFI's had zero length, and the mean bit EFIL was 1.4 to 1.6 bits. These results are favourable in terms of TI-NET's public packet structure, since the strong tendency for errors within packets to cluster, reduces the probability of many users being affected.

Finally, a brief discussion of TI-NET node synchronization and the results of satellite propagation delay measurements were presented. The results indicated an overall delay variation of about 0.7 ms throughout the tests, and an ability for the synchronization method to track these variations. Also, packet processing and reception was maintained in synchronization, even after such an event as a momentary outage (due to frame loss in the satellite terminal equipment).

These experiments proved to be useful for studying the feasibility of a satellite subnet and satellite extension nodes for the TI-NET system, as well as error control based on retransmission for this system. Also the exponential model for packet EFIL's was formulated more generally, and predicted improvements in error performance were illustrated by way of example for a 56 kb/s satellite link, for a wide range of packet error rates and terrestrial link speeds.

6.2 Suggestions for Further Work

6.2.1 Simulation and Analysis

A major area for extensive further study in the TI-NET system, involves the flow control scheme, and associated queueing and admission delay problems. Other members of the team of graduate students (mentioned in Chapter 1) have studied, or propose to study various aspects of these problems:

- D. Guy is completing a simulation study of two-hop traffic, involving eight nodes and a particular type of servicing called "Instantaneous Bandwidth Expansion" (IBE) [56,57].
- N. Janakiraman has done single-hop simulation studies involving IBE for both a transparent and non-transparent mode of operation, and proposes to do a three-hop simulation study, as well as emulation using micro-processors.
- J. Lyrette is performing an analysis of the basic principles of channel flow control using flow graph and matrix techniques.

From the work done thus far, it is evident that in general, even for relatively simple multi-hop configurations, closed form solutions to describe the behavior and performance of the flow control scheme are not possible. For this reason, simulation has been, and should continue to be relied on heavily to provide solutions. It is also evident however, that for more complex multi-hop configurations, a simulation approach will, in addition to being very expensive, exhaust the resources of even a large computing facility, so that

experimental and emulation techniques should then be resorted to.

In spite of this however, there are some relatively simpler subproblems, which may be amenable to a more analytical approach. In this thesis a beginning has been made in the analysis of such problems. However, specifically, for the simplest case of single-hop synchronous traffic, (not involving flow control) a solution for a finite number of servers ($M/M/m$), with a non-homogeneous population, remains to be found. Although an exact solution may be difficult to obtain, useful approximate solutions may exist. One approximate approach is to pursue further the parallel server facility representation described in Chapter 4. In that chapter it was seen that for infinite servers, the parallel facilities (one for each job type) operated in an independent manner, and the pdf for the number of "bulk units" in the system was the convolution of the corresponding pdf's for each facility. For finite servers however, an obvious dependence among the facilities arises; i.e. now the performance for a particular message type, depends on the extent to which the limited resources (servers) are occupied by other message types. This is also complicated by packing inefficiency which, depending on the service discipline used, may penalize some message types more than others. In spite of this however, simulation results for the finite server case have indicated that overall performance

with respect to such parameters as the mean number of messages in the queue, the mean waiting time and the probability of queueing, are the weighted (by the traffic balance g_i) sum of the performance for the individual message types. A similar analytical result was obtained for mean service time in the infinite server case. Hence there is some grounds for pursuing (as an approximation) independent finite server parallel facilities. In this regard, if the total number of servers was divided among the facilities say according to traffic (or possibly load) balance, (i.e. $m = \sum_{i \in S} g_i m_i$), the resulting overall performance would be pessimistic since in the true system, there is a statistical advantage as a result of server sharing among the user types. On the other hand, if the total number of servers (m) was assigned to each user type, the overall performance would be optimistic, since on the average in the true system, not that many servers are available to each user type. Therefore somewhere between these two extremes, there is likely an effective total number of servers, which when properly divided among the user types, would approximate true performance. Although this simple approach does not take into account such effects as packing efficiency, there are promising methods of minimizing these effects as discussed presently.

A disadvantage of parallel servicing in TI-NET is packing inefficiency, which because of the non-homogeneous population can result in unoccupied servers even though messages are

queued. However, there are three approaches which may be taken to minimize this inefficiency. These are: 1) to use an appropriate service discipline (FCFS, SFCFS and more sophisticated disciplines have been discussed), 2) expand the rate of service on messages which have incurred admission delay, and are either partially or fully queued, in order to utilize the unoccupied servers; i.e. use Instantaneous Bandwidth Expansion (IBE), 3) divide the channel capacity by a flexible boundary between synchronous and asynchronous traffic (with priority for synchronous traffic), to minimize the number of unoccupied servers. All of these approaches have already been introduced and discussed elsewhere [56,57], and some of them have been given preliminary study mainly using simulation. However, (for the single-hop case) much work remains to try and formulate these approaches analytically (approximately if not ²exactly) and to validate these formulations by simulation.

Another important factor requiring study is the assignment of priorities among synchronous traffic applications, and specifically giving priority to voice traffic over other synchronous data traffic.

With regard to multi-hop traffic, it is noted that the model for such traffic presented in this thesis, is simplified to the point where it only accounts for the rudimentary aspects of this traffic; these aspects are the mean times to request, locate and deliver reservations to an entrance

node, as well as to obtain sufficient channel capacity (servers) on the first link. For simplicity, exponential pdf's were assumed for each of these times, whereas in some cases this assumption is conservative. Also for simplicity, a closed network with homogeneous population was assumed whereas, the population in general is non-homogeneous, and not all job applications (e.g. voice) result in completely closed network behavior. Thus a more representative model for multi-hop traffic would eliminate at least some of these simplifications, and possibly include the effects of other factors as well.

In general, some other factors requiring further study are the effects of assigning priorities among types of synchronous traffic (e.g. voice over data), the effect of giving priority to synchronous over asynchronous traffic, the effects of various strategies for managing and distributing reservations for multi-hop traffic, and the effects of various strategies for dynamically allocating channel capacity for single and multi-hop traffic.

6.2.2 Experimental

Thus far, the experimental work has demonstrated the feasibility of packetized voice and data transmission using a transparent intelligent approach; this was done in an experimental two node terrestrial network, as well as over an experimental satellite link. In addition, emulation was used in order to subject voice traffic to the delays which would normally arise from as yet unimplemented network functions.

One such very important function upon which network performance on a large scale will depend critically, and which is likely to involve significant complexity, is the TI-NET flow control scheme. Thus in parallel with simulation and analysis studies, an experimental phase consisting of the implementation and evaluation of the flow control protocol is appropriate. A relatively simple physical network, consisting for example of four (perhaps micro-processor implemented) nodes as in Figure 4.1, together with the emulation of other nodes (and terminals), would provide the basis for initial studies of both single and multi-hop traffic, for various resource (channel capacity and reservation) allocation strategies. Such a facility would also enable a more realistic evaluation of the effects of admission delays on voice traffic, as well as the merits of giving priority to voice traffic, and of implementing various reservation supply strategies.

REFERENCES

1. McQuillan, J.M., Crowther, W.R., Cossel, B.P., Walden, D.C. and Heart, F.E., "Improvements in the design and performance of the ARPA network", 1972 Fall Joint Comp. Conf., AFIPS Proc., Vol.34, pp.139-152.
2. Frank, H., Frisch, I.; Hsieh, W., Occhiogrosso, B. and Shneider, K., "Issues in the design of networks with integrated voice and data", ICC Record, Vol.3, Chicago, June 1977, pp.38.1-36 to 38.1-43.
3. Cohen, D., "Issues in transnet packetized voice communication", Conference Record of Fifth Data Com. Symp., Snowbird, Utah, Sept.27-29, 1977, pp.6-10 to 6-13.
4. Ulug, M.E., "Systems design of a transparent intelligent network for data and voice", submitted to IEEE Trans. on Com.
5. Flanagan, J.L., Speech analysis synthesis and perception, 2nd Ed, Springer-Verlag, 1972.
6. Emling, J.W. and Mitchell, D., "The effects of time delay and echos on telephone conversations", B.S.T.J., Vol.XLII, No.6, Nov. 1963, pp.2869-2891.
7. Weinstein, S.B., "Echo cancellation in the telephone network", IEEE Com. Society Magazine, Vol.15, No.1, Jan. 1977, pp.9-15.
8. Gould, R.G. and Helder, G.K., "Transmission delays and echo suppression", IEEE Spectrum, April 1970, Vol.7, No.4, pp.47-54.
9. Riesz, R.R. and Klemmer, F.T., "Subjective evaluation of delay and echo suppressors in telephone communications", B.S.T.J., Vol.XLII, No.6, pp.2919-2941.
10. Krauss, R.M. and Bricker, P.D., "Effects of transmission delay and access delay on the efficiency of verbal communications", J.A.S.A., Sept. 1967, Vol.41, No.2, pp.286-292.
11. Klemmer, E.T., "Subjective evaluation of transmission delay in telephone conversations", B.S.T.J., Vol.XLVI, No.6, July-Aug. 1967, pp.1141-1147.

12. Brady, P.T., "Effects of transmission delay on conversational behaviour on echo-free telephone circuits", B.S.T.J., Jan.1971, pp.115-134.
13. Vartabedian, A.G., "The effects of transmission delay in four-wire teleconferencing", B.S.T.J., Vol.XLV, No.10, Dec.1966, pp.1673-1688.
14. Roberts, L.D. and Wessler, B.D., "Computer network development to achieve resource sharing", Proc. of AFIPS 1970 Spring Joint Computer Conf., Vol.36, pp.543-549.
15. Pouzin, L., "Presentation and major design aspects of the Cyclades Computer Network", Proc. of Third Data Com. Symp., St.Petersburg, Florida, Nov.1973, pp.80-85.
16. Barber, D.L.A., "A European informatics network: achievement and prospects", Proc. Third ICC, Toronto, Aug.1976, pp.44-50.
17. Solomon, R.J., "Packet networks", Mini Micro Systems, Vol.11, No.3, March 1978, pp.48-53.
18. Clipsham, W.W., Glave, F.E., and Narraway, M.L., "Datapac Network Overview", Proc. of Third ICC, Toronto, Aug.1976, pp.131-136.
19. Naylor, W.E., "A status report on the real-time speech transmission work at UCLA", NSC Note 52, Dec.7, 1974.
20. Magill, D.T., "Adaptive speech compression for packet communication systems", NTC Record, Vol.2, Nov.1973, Atlanta, Georgia, pp.29D-1 to 29D-5.
21. Coviello, G.J., Lake, O.L., and Redinbo, G.R., "System design implications of packetized voice", ICC Record, Vol.3, Chicago, June 1977, pp.38.3-49 to 38.3-53.
22. Forgie, J.W., "Speech transmission in packet-switched store and forward networks", AFIPS NCC Proc., Vol.44, Part 1, May 1975, Anaheim, Calif., pp.137-142.
23. Forgie, J.W., "Speech communications in packet-switched networks", MIT Lincoln Lab., Lexington, Mass. 02173, (91st Meeting of Ac.Soc. of Am. Washington, D.C. April 1976).
24. Forgie, J.W., "Speech communications in packet-switched networks", (Abstract RR14), J.A.S.A., Vol.59, Supp.No.1, Spring 1976, pp.S98.

25. Gold, B., "Digital speech networks", Proc. IEEE, Vol.65, No.12, Dec.1977, pp.1636-1658.
26. Huggins, A.W.F., "Effect of lost packets on speech intelligibility", NSC Note 78, Feb.24, 1976, B.B. & N., Cambridge, Mass.
27. Forgie, J.W., "Network speech: system implications of packetized speech", Annual report ESD-TR-77-178 to the Defense communications agency. Prepared by MIT Lincoln Lab., Lexington, Mass., Sept.1976.
28. Forgie, J.W., "Subjective effects of anomalies in packetized speech", MIT Lincoln Lab., Lexington, Mass. 02173, (92nd Meeting of Ac.Soc. of Am. Washington, D.C., Nov.1976).
29. Forgie, J.W., "Subjective effects of anomalies in packetized speech", (Abstract RR16), J.A.S.A., Vol.60, Supp.No.1, Fall 1976, p.S109.
30. Flanagan, J.L., Coker, C.H., Robiner, L.R., Schafer, R.W. and Umeda, N., "Synthetic voices for computers", IEEE Spectrum, Vol.7, No.10, Oct.1970, pp.22-45.
31. Flanagan, J.L., "The synthesis of speech", Scientific American, Vol.226, No.2, Feb. 1972, pp.48-58.
32. Lummis, R.C., "Speaker verification: a step toward the 'checkless' society", Bell Lab. Record, Sept.1972, pp.254-259.
33. Bayless, J.W., Campanella, S.J. and Golbert, A.J., "Voice signals bit-by-bit", IEEE Spectrum, Vol.10, No.10, Oct. 1973, pp.28-34.
34. Flanagan, J.L., "Computers that talk and listen: man-machine communication by voice", Proc. IEEE Special issue on man-machine communication by voice, Vol.64, No.4, Apr. 1976, p.413.
35. Rabiner, L.R. and Shafer, R.W., "Digital techniques for computer voice response: implementations and applications", Proc. IEEE Special issue on man-machine communication by voice, Vol.64, No.4, Apr.1976, pp.416-433.
36. Martin, T.B., "Practical applications of voice input to machines", Proc. IEEE Special issue on man-machine communication by voice, Vol.64, No.4, Apr.1976, pp.487-501.

37. Blackman, J.A., "Integrated switching of voice and non-voice traffic", ICC Record, Vol.2, Philadelphia, June 1976, pp.24-1 to 24-6.
38. Ross, M.J., Tabbot, A.C. and Waite, J.A., "Design approaches and performance criteria for integrated voice/data switching", Proc. IEEE, Vol.65, No.9 (special issue on telecommunication circuit switching) Sept.1977, pp.1283-1295.
39. "Transmission Systems for Communications", Rev.4th Ed., Bell Telephone Laboratories, 1971.
40. Suyderhoud, H.G., Janakowski, J.A. and Ridings, R.P., "Results and analysis of the speech predictive encoding communications system field trial", Comsat Tech. Rev., Vol.4, No.2, Fall 1974, pp.371-391.
41. Adoul, J.P. and Daaboul, F., "Digital TASI generalization with voiced/unvoiced discrimination for tripling T1 carrier capacity", ISC Proc. Vol.2, Chicago, Ill., June 1977, pp.32.5-310 to 32.6-314.
42. Norwine, A.C. and Murphy, O.J., "Characteristic time intervals in telephonic conversation", B.S.T.J., Vol.17, Apr. 1938, pp.281-291.
43. Brady, P.T., "A statistical analysis of on-off patterns in 16 conversations", B.S.T.J., Jan.1968, pp.73-91.
44. Webber, S.A., Harris, C.J. and Flanagan, J.L., "Use of variable-quality coding and time-interval modification in packet transmission of speech", B.S.T.J., Vol.56, No.8, Oct.1977, pp.1569-1573.
45. Hofstetter, E.M., Tierney, J. and Wheeler, O.C., "Micro-processor based realization of a linear predictive vocoder", IEEE Int. Conf. on Ac., Sp. and Sig.Proc., Hartford, Conn., May9-11, 1977, pp.421-424.
46. Ulug, M.E., "A unique flow control technique for a transparent intelligent network", to be published in the proc. of ICC 1978, Toronto, Ont., June 1978.
47. Ulug, M.E., Weir, D.F., Morris, L.R., and Gruber, J.G., "An experimental transparent intelligent network", Proc. of 3rd ICC, Toronto, Ont., Aug.3-6, 1976, pp.323-329.
48. Ulug, M.E. and Gruber, J.G., "Statistical multiplexing of data and encoded voice in a transparent intelligent network", Conference record of Fifth Data Com. Symp., Snowbird, Utah, Sept.27-29, 1977, pp.6-14 to 6-20.

49. Ulug, M.E. and Gruber, J.G., "An experimental satellite link for a transparent intelligent network", NTC Record, Vol.3, Dec.1977, Los Angeles, Calif., pp.48:4-1 to 48:4-8.
50. Weir, D.F., "Software design and implementation of a transparent intelligent network", M.Eng. thesis, Dept. of Syst.Eng. and Comp.Sci., Carleton University, Ottawa, Canada, 1977.
51. Morris, L.R., "Real-time software speech synthesis", IEEE Int. Conf. Rec. on Speech Com. & Proc., Apr.1972, pp.166-169.
52. Morris, L.R. and Gough, M.J., "An economical hardware realization of a digital linear predictive speech synthesizer", IEEE Trans. on Com., Vol.22, No.1, Jan.1974, pp.82-88.
53. Sproule, D.E., "Response time prediction and optimization for stored data in speech synthesis", M.Eng. thesis, Carleton University, Ottawa, June 1974.
54. Naylor, W.E. and Kleinrock, L., "On the effect of periodic routing updates in packet-switched networks", NTC Record, Dallas, Texas, Nov.1976, pp.16.2-1 to 16.2-7.
55. Coll, D.C., George, D.A., Strickland, L.H., Guild, P.D. and Paterson, S.A., "Multidisciplinary applications of communication systems in teleconferencing and education", IEEE Trans. on Com., (Special issue on social implications of telecommunications), Oct.1975, Vol.23, No.10, pp.1104-1118.
56. Ulug, M.E., Gruber, J.G. and Janakiraman, N., "Admission delays in a transparent intelligent network", to be published in the proc. of ICC 1978, Toronto, Ont., June 1978.
57. Ulug, M.E., Janakiraman, N. and Gruber, J.G., "Simulation of admission delays in a transparent intelligent network", accepted by Summer Computer Simulation Conf., to be held in July 1978, at Newport Beach, Calif.
58. Schroeder, M.R., "Vocoders: analysis and synthesis of speech", (a review of 30 years of applied speech research), Proc. IEEE, Vol.54, No.5, May 1966, pp.720-734.
59. Jayant, M.S., "Digital coding of speech waveforms: PCM, DPCM and DM quantizers", Proc. IEEE, Vol.62, No.5, May 1974, pp.611-632.

60. Rabiner, L.R., Schafer, R.W. and Flanagan, J.L., "Computer synthesis of speech by concatenation of formant-coded words", B.S.T.J., Vol.50, No.5, May-June 1971, pp.1541-1558.
61. Birch, J.N., "A comparison of voice coding techniques for a satellite-based air traffic control system", IEEE Int. Conf. Rec. on Speech Com. & Proc., April 1972, pp.352-355.
62. Birch, J.N., "Low bit rate voice processors", Proc. of EASCON Conf. sponsored by IEEE in Washington, 1974, pp.59-63.
63. Gold, B. and Rader, C.M., "The channel vocoder", IEEE Trans. on Audio & Electro Acoustics, Vol.AU-15, No.4, Dec.1967, pp.146-161.
64. Bially, T. and Anderson, W.M., "A digital channel vocoder", IEEE Trans. on Com., August 1970, Vol.18, No.4, pp.435-442.
65. Rabiner, L.R. and Gold, B., "Theory and application of digital signal processing", Prentice-Hall, 1975, Chap.12.
66. Oppenheim, A.V. and Schafer, R.W., "Digital signal processing", Prentice-Hall, 1975, Chap.10.
67. Rosenthal, L.H., Rabiner, L.R., Schafer, R.W., Cummiskey, R. and Flanagan, J.L., "Automatic voice response: interfacing man with machine", IEEE Spectrum, Vol.11, No.7, July 1974, pp 61-68.
68. Atal B.S. and Hanover, S.L., "Speech analysis and synthesis by linear prediction of the speech wave", J.A.S.A., Vol.50, No.2, Pt.2, 1971, pp.637-655.
69. Markel, J.D. and Grey, A.H. Jr., "Linear prediction of speech", Springer-Verlag, 1976.
70. Fulghum, D.P., "An all-digital vocoder", EASCON Record, Washington, D.C., (IEEE Pub. 74CH0883-1 AES) Oct.1974, pp.50-50G.
71. Fulghum, D.P., "Output spectrum contour scaling for an all digital channel vocoder", IEEE Int. Conf. Record on Ac., Sp. and Signal Proc., Philadelphia, Apr.1976, pp.99-102.
72. Fulghum, D.P., "Subjective effects of phase delay on synthesized speech", IEEE Int. Conf. on Ac., Sp. and Sig. Proc., Hartford, Conn., May 9-11, 1977, pp.587-590.
73. Edwards, P.G. and Clapper Jr., J., "Better vocoders are coming", IEEE Spectrum, Sept.1964, Vol.1, No.9, pp.119-129.

74. Atal, B.S. and Schroeder, M.R., "Adaptive predictive coding of speech signals", B.S.T.J., Oct.1970, pp.1973-1986
75. Goldberg, A.J., "Adaptive predictive coding at 6400 bits/second", EASCON Record, Washington, D.C., (IEEE Pub. 74CH0883-1 AES), Oct.1974, pp.64-69.
76. Kang, G.S. and Coulter, D.C., "600 bps voice digitizer", IEEE Int. Conf. on Ac., Sp. and Sig. Proc., Philadelphia, Pa., April 12-14, 1976, pp.91-94.
77. Viswanathan, R. and Makhoul, J., "Current issues in linear predictive speech compression", EASCON Conf. sponsored by IEEE in Washington, 1974, pp.577-585.
78. Viswanathan, R. and Makhoul, J., "Quantization properties of transmission parameters in linear predictive systems", IEEE Trans. Ac. Sp. & Sig. Proc., Vol. ASSP-23, June 1975, pp.309-321.
79. Markel, J.D. and Grey, A.H. Jr., "A comparison of three methods for coefficient quantization and bit allocation", Int. Conf. on Ac., Sp., and Sig. Proc., Philadelphia, Pa., April 12-14, 1976, pp.481-483.
80. Sambur, M.R., "An efficient linear-prediction vocoder", B.S.T.J., Vol.54, No.10, Dec. 1975.
81. Blackman, E., Viswanathan, R. and Makhoul, J., "Variable-to-fixed rate conversion of narrow band LPC speech", IEEE Int. Conf. on Sp., Ac. and Sig. Proc., Hartford, Conn. May 9-11, 1977, pp.409-412.
82. Cole, R.E. and Baynton, T.L., "A real-time floating point variable frame rate LPC vocoder", IEEE Int. Conf. on Sp., Ac. and Sig. Proc., Hartford, Conn., May 9-11, 1977, pp.429-432.
83. White, C.E., "Packet switching for fast and secure data", Telecommunications, Vol.12, No.1, Jan.1978, pp.33-38.
84. Rosner, R.D., "Packet switching and circuit switching: a comparison", NTC Record, Vol.2, New Orleans, Dec.1975, pp.42-1 to 42.7.
85. Esterling, R. and Hahn, P., "A comparison of digital data network alternatives", NTC Record, Vol.2, New Orleans, Dec.1975, pp.42-8 to 42-11.
86. Takehiko, Y. and Shimasaki, N., "A study of future integrated service digital networks", NTC Record, Vol.1, New Orleans, Dec.1975, pp.7-1 to 7-6.

87. Wang, J. and Ming, T.L., "Analysis and simulation of the mixed voice/data transmission system (MVD) for computer communication", NTC Record, Vol.3, Dallas, Nov.-Dec.1976, pp.42.3-1 to 42.3-5.
88. Joel, A.E. Jr., "What is telecommunication circuit switching?", Proc. IEEE, Special issue on telecommunications circuit switching, Vol.65, No.9, Sept.1977, pp.1237-1253.
89. Coviello, G.J. and Vena, P.A., "Integration of circuit/packet switching by a SENET (Slotted Envelope Network) concept", NTC Record, Vol.2, New Orleans, Dec.1975, pp.42-12 to 42-17.
90. Fisher, M.J. and Harris, T.C., "An analysis of an integrated circuit- and packet-switched telecommunications systems", NTC Record, Vol.2, New Orleans, Dec.1975, pp.42-18 to 42:23.
91. Fisher, M.J. and Harris, T.C., "A model for evaluating the performance of an integrated circuit- and packet-switched multiplex structure", IEEE Trans. on Com., Vol. Com-24, No.2, Feb.1976, pp.195-202.
92. Occhiogrosso, B., Gitman, I., Hsieh, W. and Frank, H., "Performance analysis of integrated switching communication systems", NTC Record, Vol.1, Los Angeles, Calif., Dec.1977, pp.12:4-1 to 12:4-13.
93. Forgie, J.W. and Nameth, A.G., "An efficient packetized voice/data network using statistical flow control", ICC Record, Vol.3, Chicago, June 1977, pp.38.2-44 to 38.2-48.
94. Tymes, L.R., "TYMNET - a terminal oriented communication network", AFIPS Conf. Proc., Vol.38, 1971, pp.211-216.
95. Schwartz, M., Boorstyn, R.R. and Pickholtz, R.L., "Terminal-oriented computer-communication networks", Proc. IEEE, Vol.60, No.11, Nov.1972, pp.1408-1423.
96. Schwartz, M., "Computer-communication network design and analysis", Prentice-Hall Inc., 1977.
97. Fraser, A.G., "Spider - an experimental data communications system", Proc. ICC, Minneapolis, June 1974, pp.21F-1 to 21F-10.
98. Fraser, A.G., "The present status and future trends in computer communication technology", Communications Society Magazine, Vol.14, No.5, Sept.1976, pp.10-27.
99. de Smet, J. and Sanders, R.W., "A network combining packet switching and time division circuit switching in a common system", Comp.Com.Rev., Vol.6, No.1, Jan.1976, pp.38-62.

100. de Smet, J. and Sanders, R.W., "'Pacuit' switching combines two techniques in one network", Computer Design, June, 1976, pp.83-88.
101. Klein, D.M. and Frankel, S., "A hybrid circuit and packet-switching system", NTC Record, Dallas, Texas, Nov. 1976, pp.7.3-1 to 7.3-6.
102. Forney, G.D. and Vander Mey, J.E., "The Codex 6000 series of intelligent network processors", Comp.Com.Rev., Vol.6, No.2, April 1976, pp.7-11.
103. Forney, G.D. and Stearns, R.W., "Statistical multiplexing improves link utilization", Data Communications, July/Aug. 1976, pp.37-41.
104. Vander Mey, J.E., "The architecture of a transparent intelligent network", NTC Record, Dallas, Texas, Nov.1976, pp.7.2-1 to 5.
105. Kleinrock, L., "Communication nets: Stochastic message flow and delay", Dover Publications, 1964.
106. Davies, D.W. and Barber, D.L.A., "Communications Networks for computers", Wiley, 1975.
107. Chu, W.W., "Optimum block size for computer communications with error detection and retransmission strategies", IEEE Trans. on Com., Vol.Com-22, No.10, Oct.1974, pp.1516-1525.
108. Burton, H.O. and Sullivan, D.D., "Errors and error control", Proc. IEEE, Vol.60, No.11, Nov.1972, pp.1293-1301.
109. Pouzin, L., "Flow control in data networks - methods and tools", Proc. 3rd ICC, Toronto, Ont., Canada, Aug.3-6, 1976, pp.467-474.
110. Kahn, R.E. and Crowthers, W.R., "Flow control in a resource showing computer networks", IEEE Trans. on Com., Vol.Com-20, No.3, June 1972, pp.539-546.
111. Davies, D.W., "The control of congestion in packet switched networks", IEEE Trans. on Com., Vol.Com-20, No.3, June 1972, pp.546-550.
112. Kleinrock, L., "Queueing systems, Vol.1: Theory", John Wiley & Sons, 1975.
113. Gross, D. and Harris, C.M., "Fundamentals of queueing theory", John Wiley & Sons, 1974.

114. Little, J.D.C., "A proof of the queueing formula $L=\lambda W$ ", Operations Research, 1961, Vol.9, pp.383-387.
115. Kleinrock, L., "Queueing systems, Vol.II: computer applications", John Wiley & Sons, 1975.
116. Baskett, F. and Muntz, R.R., "Networks of queues", Proc. of 7th annual Princeton Conf. on Info. Sci., Princeton University, 1973, pp.428-434.
117. E-Systems Inc., "VADAC 5 Commercial Digital Speech Processor", Technical Data Sheet 4-400200-8-75-2000-4.
118. E-Systems Inc., "VADAC 5 Information and Applications Manual", Document No.416-04146, Sept.1975.
119. VADAC Model II Speech Processor Operation and Maintenance Instructions, E. Systems Inc., Garland Division, Dallas, Texas, (Part Number 401-17400-01), Feb.12, 1973.
120. VADAC Model V Speech Processor Operation and Maintenance Instructions, E. Systems Inc., Garland Division, Dallas, Texas, (Part Number 401-24500), June 24, 1976.
121. Data Compaction System Operation and Maintenance Instructions, E. Systems Inc., Garland Division, Dallas, Texas, May 1, 1973. Section III, Theory of Operation.
122. E. Systems Inc., Description of Diagnostic Rhyme Test (DRT), Document No.416-04141.
123. High Level Data Link Control Procedures, International Standard ISO/TC 97/SC 6 (Tokyo-17), October 1974.
124. IBM Synchronous Data Link Control General Information, GA27-3090-0, March 1974.
125. Martin, J., "Teleprocessing Network Organization", Prentice-Hall Inc., 1970.
126. PDP-11 Peripherals Handbook, Digital Equipment Corporation (see most recent version).
127. Lum, Y.F., "Integration of satellite and terrestrial communication networks", Communications Satellite Systems: An Overview of the Technology, Ed. by R.G. Gould and Y.F. Yum, IEEE Press, 1976, pp.131-135.

128. Evans, W.M., Davies, N.G. and Hawersaat, W.H., "The Communications Technology Satellite (CTS) Program", Communications Satellite Systems: An Overview of the Technology, Ed. by R.G. Gould and Y.F. Lum, IEEE Press, 1976, pp.13-18.
129. Huck, R.W., Communications Experimentors' Guide (for) Communications Technology Satellite, CRC technical note no.671 (CS-03-01), Feb.1975, pp.101-106. Department of Communications, Ottawa.
130. Codex 9600 High Speed Data Modem Operation Manual, Codex Corporation, Massachusetts, Part No.0R960D4F, April 1974.
131. LT8E/LT8D Real-Time Hybrid Hadamard Transform Video Compression System Instruction Manual, Linkabit Corporation, California, 1975.
132. STI Model 8064 QPSK Modem Operation/Maintenance Manual, Stanford Telecommunications Inc., California, STI-IO-5315A, June 16, 1976.
133. LV7017MR Convolutional Encoder, Viterbi Decoder Instruction Manual, Linkabit Corporation, California, 1975.
134. Cohen, A.R., "Practical Aspects of Convolutional Encoding with Viterbi Decoding", Proc. of ICC, Vol.II, San Francisco, 1975, pp.19-10 to 19-14.
135. Nuspl, P.P., Brown, K.E., Steenaart, W. and Ghiconi, B., "Synchronization methods for TDMA", IEEE Proc., (Special Issue on Satellite Communications), Mar. 1977, Vol.65, No.3, pp.434-444.
136. Nesenbergs, M., "Error control for large and reliable satellite networks", NTC Record, Vol.1, Dallas, Nov.-Dec., 1976, pp.13.2-1 to 13.2-5.
137. Georganas, N.D., Shiva, S.G.S., Verma, P.K. and Jawanda, J.S., "Evaluation of the mean error-free interval of a noisy data channel", IEEE Trans. on Com., Corresp., Vol. COM-26, No.1, Jan.1978, pp.185-187.
138. Schwartz, M., Bennett, W.R. and Stein, S., Communication Systems and Techniques, McGraw-Hill, 1966.

APPENDIX A

A.1 Summary of Results for an M/M/m System

- pdf for the number of messages in the system:

$$P_k = P_0 \frac{(m\rho)^k}{k!} \quad k \leq m$$

$$= P_0 \frac{\rho^k m^m}{m!} \quad k \geq m$$

$$P_0 = \text{Prob (system is empty)}$$

$$= \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho} \right]^{-1}$$

$$P_q = \text{Prob (of queueing)}$$

$$= P_0 \frac{(m\rho)^m}{m!} \frac{1}{1-\rho}$$

$$\rho = \text{utilization factor}$$

$$= \frac{\lambda}{\mu m}$$

$$\lambda = \text{total arrival rate (msg's/sec)}$$

$$\mu = \text{service rate per server (msg's/sec)}$$

$$m = \text{number of servers}$$

- mean number in the system:

$$N_t = \text{number in service plus number in the queue}$$

$$= N_s + N_q = \rho m + \frac{\rho P_q}{1-\rho}$$

- pdf for the total time in the system:

$$P(t) = \mu \left[1 + \frac{P_q e^{-\mu t}}{m(1-\rho)-1} - \frac{m(1-\rho)P_q}{m(1-\rho)-1} e^{-(1-\rho)\mu t} \right]$$

- normalized mean time in the system:

$\mu m T_t$ = normalized times in service plus waiting

$$= \mu m (T_s + T_q) = m + \frac{P_q}{1-\rho}$$

- the second moment of $P(t)$ is obtained from

$$\bar{t}_t^2 = \int_0^{\infty} t^2 P(t) dt$$

$$\text{let } G = m(1-\rho) - 1$$

$$\text{then } P(t) = \left(1 + \frac{P_q}{G} \right) \mu e^{-\mu t} - \frac{P_q}{G} (1-\rho) \mu e^{-(1-\rho)\mu t}$$

$$\text{therefore } \bar{t}_t^2 = \left(1 + \frac{P_q}{G} \right) 2 \left(\frac{1}{\mu} \right)^2 - \frac{P_q}{G} \frac{2}{[(1-\rho)\mu]^2}$$

$$\bar{t}_t^2 = 2 \left(\frac{1}{\mu} \right)^2 + \frac{2}{(\mu m)^2} \frac{P_q}{G} \left(m^2 - \frac{1}{(1-\rho)^2} \right)$$

$$\text{But } m^2 - \frac{1}{(1-\rho)^2} = \frac{m^2 (1-\rho)^2 - 1}{(1-\rho)^2} = \frac{G(m(1-\rho)+1)}{(1-\rho)^2}$$

$$\therefore \bar{t}_t^2 = 2 \left(\frac{1}{\mu} \right)^2 + \frac{2}{(\mu m)^2} \frac{P_q}{(1-\rho)^2} (m(1-\rho)+1);$$

the variance of $P(t)$ is

$$\sigma_t^2 = \bar{t}_t^2 - T_t^2$$

$$\therefore (\mu m)^2 \sigma_t^2 = 2m^2 + 2 \frac{P_q}{(1-\rho)^2} (m(1-\rho)+1) - \left(m + \frac{P_q}{1-\rho} \right)^2$$

Simplifying gives

$$(\mu m)^2 \sigma_t^2 = m^2 + \frac{(2-P_q)P_q}{(1-\rho)^2}$$

$$\text{Since } \sigma_t^2 = \sigma_s^2 + \sigma_q^2,$$

$$\mu m \sigma_q = \frac{\sqrt{(2-P_q)P_q}}{1-\rho}$$

A.2 Algorithm for Determining the Ways of Having k Bulk Units in the System (Figure 4.7)

The algorithm determines in the following order the ways of having k in: facility 1, facilities 1 and 2, facilities 1, 2 and 4 and finally in facilities 1, 2, 4 and 8. Examples for k even (k=14) and k odd (k=17) are shown in Tables A.2.1 and A.2.2 respectively. In general the ways of having an odd number of bulk units in the system are identical to the ways of having the immediately preceding even number of bulk units, except that for each way in the odd case, facility 1 contains the extra - odd - entry.

The breakdown of the number of ways of having k bulk units in the system is shown in Figure A.2.1. There is only one way to have k in the system using facility 1 alone and $\lfloor k/2 \rfloor$ ways using facilities 1 and 2. (The notation $\lfloor x \rfloor$ means: x if x is an integer, or next smallest integer of x if x is not an integer). The total number of ways of having k+1 odd bulk

TABLE A.2.1
ALGORITHM OUTPUT FOR EVEN NUMBER OF BULK UNITS

RUN COMB

ENTER NO. OF BULK UNITS IN THE SYSTEM:

14

NO. OF WAYS TO HAVE THAT NO. = 26

WAY	NO. OF MESSAGES IN FACILITY				TOTAL
	1	2	4	8	
1	14	0	0	0	14
2	12	1	0	0	13
3	10	2	0	0	12
4	8	3	0	0	11
5	6	4	0	0	10
6	4	5	0	0	9
7	2	6	0	0	8
8	0	7	0	0	7
9	10	0	1	0	11
10	8	1	1	0	10
11	6	2	1	0	9
12	6	0	2	0	8
13	4	3	1	0	8
14	4	1	2	0	7
15	2	4	1	0	7
16	2	2	2	0	6
17	2	0	3	0	5
18	0	5	1	0	6
19	0	3	2	0	5
20	0	1	3	0	4
21	6	0	0	1	7
22	4	1	0	1	6
23	2	2	0	1	5
24	2	0	1	1	4
25	0	3	0	1	4
26	0	1	1	1	3

TABLE A.2.2

ALGORITHM OUTPUT FOR ODD NUMBER OF BULK UNITS

RUN COMB

ENTER NO. OF BULK UNITS IN THE SYSTEM:

17

NO. OF WAYS, TO HAVE THAT NO. = 35

WAY	NO. OF MESSAGES IN FACILITY				TOTAL
	1	2	4	8	
1	17	0	0	0	17
2	15	1	0	0	16
3	13	2	0	0	15
4	11	3	0	0	14
5	9	4	0	0	13
6	7	5	0	0	12
7	5	6	0	0	11
8	3	7	0	0	10
9	1	8	0	0	9
10	13	0	1	0	14
11	11	1	1	0	13
12	9	2	1	0	12
13	9	0	2	0	11
14	7	3	1	0	11
15	7	1	2	0	10
16	5	4	1	0	10
17	5	2	2	0	9
18	5	0	3	0	8
19	3	5	1	0	9
20	3	3	2	0	8
21	3	1	3	0	7
22	1	6	1	0	8
23	1	4	2	0	7
24	1	2	3	0	6
25	1	0	4	0	5
26	9	0	0	1	10
27	7	1	0	1	9
28	5	2	0	1	8
29	5	0	1	1	7
30	3	3	0	1	7
31	3	1	1	1	6
32	1	4	0	1	6
33	1	2	1	1	5
34	1	0	2	1	4
35	1	0	0	2	3

FIGURE A.2.1

BREAKDOWN OF NUMBER OF
WAYS OF HAVING k BULK
UNITS IN THE SYSTEM IN
FIGURE 4.7

units in the system is identical to the number of ways of having k even bulk units.

Using equations (4.6) and (4.7) and the algorithm for finding the ways of having k in the system, it is possible to compute P_k as illustrated by the first few terms in (4.8). Complete examples for P_k are also shown in Figures 4.10(a), (b) and (c).

A.3 State and Substate Representation for the Single Combined Service Facility (Figure 4.11)

It is useful to note that the ways of having k bulk units in the system are the elements of a Cartesian product in a "modified" four dimension discrete state space. The space is considered to be "modified" in the sense that the four basis vectors are indexed with an incomplete set of non-negative integers given by $0, 1, 2i, 3i, 4i, \dots$ $\forall i \in S = \{1, 2, 4, 8\}$, whereas in a "regular" space, the basis vectors would be indexed with the complete set of such integers. The elements of the Cartesian product in the modified space form a subset of the corresponding elements in the regular space. The concept is illustrated by an example in Figure A.3.1 using only two dimensions for convenience. The important point is that at any instant the "state" of the system (e.g. $k=6$) may be any one of a number of substates (ways of having k in the system; i.e. elements of the Cartesian product). However, single step transitions between the substates are not possible. For example in Figure A.3.1(b) for $k=6$, a change from element $(0,3)$ to element $(2,2)$ would

(a) "REGULAR"
2 DIMENSIONAL
DISCRETE STATE SPACE

e.g. Set consists of
7 ways to have a total
of 6.
i.e. Cartesian
product is
 $\{(6,0), (5,1), (4,2),$
 $(3,3), (2,4), (1,5), (0,6)\}$

(b) "MODIFIED"
2 DIMENSIONAL
DISCRETE STATE SPACE

e.g. Subset consists
of only 4 ways to
have a total of 6.
i.e. Cartesian
product is
 $\{(6,0), (4,1), (2,2), (0,3)\}$

FIGURE A.3.1 ILLUSTRATION OF WAYS OF HAVING
 $k=6$ IN TERMS OF CARTESIAN PRODUCTS.

involve two arrivals of type $i = 1$ (i.e. $0 + 1 + 1 = 2$) and one departure of type $i = 2$ (i.e. $3 - 1 = 2$). Therefore any single step transition will result in a complete change of state (e.g. from $k = 6$ to $k = 4$) regardless of which substate the system was in.

However, the particular new state in which the system lands as a result of a single departure will depend on which substate the system was in before the departure. For example, referring to Figure A.3.1(b), if the system was in state $k=6$ - substate $(6,0)$, then a single departure would, with certainty, land the system in state $k=5$ - substate $(5,0)$. On the other hand if the system was in state $k=6$ - substate $(2,2)$, then a single departure would (with probabilities given by equation 4.18) leave the system in either state $k=5$ - substate $(1,2)$ or in state $k=4$ - substate $(2,1)$ (see Figure A.3.2 to be described shortly).

In general therefore, it is appropriate to consider a state description based on a vector $[k, B_j(k)]$, where $B_j(k)$ represents the j th substate of k (i.e. the j th way of having k in the system). This state description is Markovian since it contains all the information relevant to the future system development. Hence, a rate transition diagram may be drawn which includes all of the substates associated with the probability of having k in the system; Figure A.3.2 is an example based on Figure A.3.1(b) for $k=6$. In this example, there are four ways to have $k=6$, so there are four

FIGURE A.3.2 - RATE TRANSITION DIAGRAM FOR $[k=6, B_j(6)]$

uncoupled rate transition diagrams; the fact that they are uncoupled indicates that there are no single step transitions between the substates for a given k , as mentioned previously.

The departure rates are given by $i\mu$ k_i . For example for substate (2,2) a departure to (1,2) takes place at rate $1 \times \mu \times 2 = 2\mu$, and a departure to (2,1) at rate $2 \times \mu \times 2 = 4\mu$.

It is possible to redraw the example in Figure A.3.2 so that the substate dependencies are incorporated in the $\bar{\gamma}_i$'s of (4.25), as shown in Figure A.3.3; the assumption of equal mean message lengths still applies. The $\bar{\gamma}_i$'s for this example are evaluated below. The notation $p(a,b)$ is used to denote the probability of being in a particular substate $a+2b$ such that there are 'a' type 1 jobs in the first facility and 'b' type 2 jobs in the second facility. Therefore using (4.25) and (4.18),

$$\begin{aligned}\bar{\gamma}_f(6) &= \frac{1}{P_6} \sum_{j=1}^3 (\gamma_1(6))_j P(A, B_j) \\ &= \frac{1}{P_6} \left[P(6,0) + \frac{2}{3} P(4,1) + \frac{1}{3} P(2,2) \right]\end{aligned}$$

Now with (4.6) (and $i = 1, 2$ for this example)

$$\begin{aligned}\bar{\gamma}_f(6) &= \frac{1}{P_6} \left[\frac{r_1^6}{6!} + \frac{2}{3} \frac{r_1^4}{4!} r_2 + \frac{1}{3} \frac{r_1^2}{2} \frac{r_2^2}{2} \right] P_0 \\ &= \frac{r_1}{6P_6} \left[\frac{r_1^5}{5!} + \frac{r_1^3}{3!} r_2 + \frac{r_1 r_2^2}{2} \right] P_0 = \frac{r_1}{6} \frac{P_5}{P_6}\end{aligned}$$

FIGURE A.3.3 RATE TRANSITION DIAGRAM FOR
FIGURE A.3.2 WITH SUBSTATE
DEPENDENCIES ABSORBED

Similarly $\bar{\gamma}_2(6) = \frac{r_2}{3} \frac{P_4}{P_6}$

$$\text{Also } \bar{\gamma}_1(6) + \bar{\gamma}_2(6) = \frac{1}{3P_6} \left[\frac{r_1}{2} P_5 + r_2 P_4 \right] = 1$$

In general for departures from state k to states $k-i$, it can be shown by induction that

$$\bar{\gamma}_i(k) = \frac{ir_i}{k} \frac{P_{k-i}}{P_k} \quad (\text{A.3.1})$$

$$\text{and } \sum_{i \in S} \bar{\gamma}_i(k) = 1. \quad (\text{A.3.2})$$

Also, redefining the events A and B_j in terms of departures from states $k+i$ to state k , and using the same approach which led to (4.25) gives

$$\bar{\gamma}_i(k+i) = \frac{1}{P_{k+i}} \sum_{j=1}^L (\gamma_i(k+i))_j P(A, B_j) \quad (\text{A.3.3})$$

Applying (A.3.3) to Figure A.3.2 gives

$$\gamma_1(7) = \frac{r_1}{7} \frac{P_6}{P_7} \quad \text{and} \quad \gamma_2(8) = \frac{r_2}{4} \frac{P_6}{P_8}$$

In general, by induction

$$\bar{\gamma}_i(k+i) = \frac{ir_i}{k+i} \frac{P_k}{P_{k+i}} \quad (\text{A.3.4})$$

The right hand balance equation in (4.30) is

$$\lambda P_k = \mu \sum_{i \in S} (k+i) \bar{\gamma}_i (k+i) P_{k+i}, \quad (A.3.5)$$

and is a function of the $\bar{\gamma}_i (k+i)$'s since the departures take place from the four different states $k+i$, whereas with the left hand equation they took place from the same state k so that $\sum_{i \in S} \bar{\gamma}_i (k) = 1$.

Substituting (A.3.4) for $\bar{\gamma}_i (k+i)$ into (A.3.5) and using equation (4.4) shows that (A.3.5) is satisfied. It is interesting to examine in more detail the way in which (A.3.5) is satisfied. This can be done by using (4.4) in (A.3.5) to give

$$\sum_{i \in S} i r_i P_k = \sum_{i \in S} (k+i) \bar{\gamma}_i (k+i) P_{k+i} \quad (A.3.6)$$

and comparing it to a re-arranged version of (A.3.4)

$$\text{i.e.} \quad i r_i P_k = (k+i) \bar{\gamma}_i (k+i) P_{k+i} \quad (A.3.7)$$

Comparing (A.3.6) and (A.3.7) suggests that the $\bar{\gamma}_i (k+i)$'s can be determined directly from (A.3.6) by expanding the left and right hand sides, equating corresponding terms in i and solving for each $\bar{\gamma}_i (k+i)$. This implies that there exists forms of right hand local balance for each i , the equations for which are given by using (4.4) in (A.3.7)

$$\text{i.e.} \quad \lambda g_i P_k = \mu(k+i) \bar{\gamma}_i(k+i) P_{k+i} \quad (\text{A.3.8})$$

Similar forms of left hand local balance can be recognized with equations (4.4) and (4.28) in (4.31) which gives

$$\sum_{i \in S} k \bar{\gamma}_i(k) P_k = \sum_{i \in S} i r_i P_{k-i} \quad (\text{A.3.9})$$

and comparing it to a re-arranged version of (4.27)

$$\text{i.e.} \quad k \bar{\gamma}_i(k) P_k = i r_i P_{k-i} \quad (\text{A.3.10})$$

Finally using equation (4.4) in (A.3.10) gives

$$\mu k \bar{\gamma}_i(k) P_k = \lambda g_i P_{k-i} \quad (\text{A.3.11})$$

Thus from (A.3.8) and (A.3.11) the localized rate transition diagram in Figure 4.13 may be drawn for each i . By equating the flows into and out of state k , the overall local balance equation is

$$(\lambda g_i + \mu k \bar{\gamma}_i(k)) P_k = \lambda g_i P_{k-i} + \mu(k+i) \bar{\gamma}_i(k+i) P_{k+i}$$

A.4 Standard Deviation for the Number of Messages in an M/M/ ∞ /M System

The pdf for the number of type i messages in an M/M/ ∞ /M system is given by (4.38), and the second moment is

$$\bar{N}_i^2 = \sum_{k=0}^{M_i} k^2 P_k$$

For simplicity, the subscript i will be dropped, then

$$\bar{N}^2 = \frac{1}{(1+\rho)^M} \sum_{k=0}^M k^2 \rho^k \binom{M}{k} \quad (\text{A.4.1})$$

Using the differentiation "trick" twice [112] gives

$$\frac{\partial}{\partial \rho} \rho^k \binom{M}{k} = k \rho^{k-1} \binom{M}{k}$$

$$\therefore \rho \frac{\partial}{\partial \rho} \rho^k \binom{M}{k} = k \rho^k \binom{M}{k} \quad (\text{A.4.2})$$

and $\frac{\partial}{\partial \rho} k \rho^k \binom{M}{k} = k^2 \rho^{k-1} \binom{M}{k}$

$$\therefore \rho \frac{\partial}{\partial \rho} k \rho^k \binom{M}{k} = k^2 \rho^k \binom{M}{k} \quad (\text{A.4.3})$$

From (A.4.2) and (A.4.3)

$$k^2 \rho^k \binom{M}{k} = \rho \frac{\partial}{\partial \rho} \left(\rho \frac{\partial}{\partial \rho} \rho^k \binom{M}{k} \right) \quad (\text{A.4.4})$$

Substituting (A.4.4) into (A.4.1) and interchanging the sum and derivatives gives:

$$\bar{N}^2 = \rho \frac{\partial}{\partial \rho} \left(\rho \frac{\partial}{\partial \rho} \sum_{k=0}^M \rho^k \binom{M}{k} \right) \quad (\text{A.4.5})$$

But it is well known that

$$\sum_{k=0}^M \rho^k \binom{M}{k} = \frac{1}{P_0} = (1+\rho)^M \quad (\text{A.4.6})$$

where $P_0 = P(\text{system empty})$.

Now substituting (A.4.6) into (A.4.5), taking the derivatives and simplifying leaves

$$\bar{N}^2 = \frac{\rho M (\rho M + 1)}{(1+\rho)^2}$$

Since $\sigma^2 = \bar{N}^2 - \bar{N}^2$, then from (4.41)

$$\sigma = \frac{\sqrt{\rho M}}{(1+\rho)}$$

A.5 Summary of Results for an M/M/m/M System

- pdf for the number of messages in the system:

$$P_k = P_0 \left(\frac{\lambda}{\mu} \right)^k \binom{M}{k}, \quad 0 \leq k \leq m-1$$

$$= P_0 \left(\frac{\lambda}{\mu} \right)^k \binom{M}{k} \frac{k!}{m!} m^{m-k}, \quad m \leq k \leq M$$

$P_0 = \text{Prob (System is empty)}$

$$= \left[\sum_{k=0}^{m-1} \left(\frac{\lambda}{\mu} \right)^k \binom{M}{k} + \sum_{k=0}^M \left(\frac{\lambda}{\mu} \right)^k \binom{M}{k} \frac{k!}{m!} m^{m-k} \right]^{-1}$$

$\lambda = \text{inverse of the mean think time per terminal}$
(msg's/sec)

$\mu = \text{service rate per server (msg's/sec)}$.

$$\binom{M}{k} = \frac{M!}{(M-k)!k!}$$

M = number of terminals.

m = number of servers.

- In Figure A.5.1, the normalized mean time spent in the queue, given by (4.53), has been plotted against the utilization factor (U.F.) defined to be the average throughput over the maximum service rate, i.e.

$$\text{U.F.} = \frac{\lambda (M-N)}{m\mu}$$

Note that as $M \rightarrow \infty$ and $\lambda \rightarrow 0$

$$\text{U.F.} \rightarrow \frac{\lambda M}{m\mu}$$

as in the open $M/M/m$ case.

Another way to interpret Figure A.5.1 is in terms of self regulation. For example with $\lambda/\mu \geq 1$ (for which the range of the number of terminals involved is relatively small) the queue is strongly regulated: i.e. for U.F.'s as high as 0.99 and 0.999, the normalized delays are only of the order of 7 and 13 respectively. This means the resource may be fully utilized without incurring prohibitive delays. However for $\lambda/\mu \rightarrow 0$, the number of terminals becomes very large and ultimately all self regulation disappears leaving an open system for which the delays are exceedingly high for the U.F.'s cited above.

FIGURE A.5.1
NORMALIZED MEAN TIME
SPENT IN THE QUEUE
FOR M/M/m/M SYSTEM

$m=32$

A.6 Convolution of Four Exponential pdf's with Different Means

$$\left. \begin{aligned} \text{Let: } g_1(t) &= ae^{-at} \\ g_2(t) &= be^{-bt} \\ g_3(t) &= ce^{-ct} \\ g_4(t) &= de^{-dt} \end{aligned} \right\} a \neq b \neq c \neq d$$

$$\text{Find: } g(t) = g_1(t) * g_2(t) * g_3(t) * g_4(t).$$

$$g_1(t) * g_2(t) = E(e^{-at} - e^{-bt})$$

$$\text{where } E = \frac{ab}{b-a} \quad (\text{A.6.1})$$

$$g_1(t) * g_2(t) * g_3(t)$$

$$= F(e^{-at} - e^{-ct}) - G(e^{-bt} - e^{-ct})$$

$$= Fe^{-at} - Ge^{-bt} + He^{-ct}$$

$$\text{where } F = \frac{Ec}{c-a}, G = \frac{Ec}{c-b}, \text{ and } H = G - F \quad (\text{A.6.2})$$

$$g_1(t) * g_2(t) * g_3(t) * g_4(t) = A(e^{-at} - e^{-dt})$$

$$+ B(e^{-bt} - e^{-dt}) + C(e^{-ct} - e^{-dt})$$

$$\therefore g(t) = Ae^{-at} + Be^{-bt} + Ce^{-ct} + De^{-dt} \quad (\text{A.6.3})$$

$$\text{where } A = \frac{Fd}{d-a}, B = -\frac{Gd}{d-b}, C = \frac{Hd}{d-c} \quad (\text{A.6.4})$$

$$\text{and } D = -(A + B + C) \quad (\text{A.6.5})$$

Using (A.6.4) and (A.6.5) and the definitions for E, F, G and H in (A.6.1) and (A.6.2) gives:

$$\left. \begin{aligned} A &= \frac{abcd}{(b-a)(c-a)(d-a)} \\ B &= \frac{abcd}{(a-b)(c-b)(d-b)} \\ C &= \frac{abcd}{(c-a)(c-b)(d-c)} \\ D &= \frac{ab(a-b)+ac(c-a)+bc(b-c)}{(b-a)(c-a)(c-b)(d-a)(d-b)(d-c)} \cdot abcd \end{aligned} \right\} \quad (\text{A.6.6})$$

$$\text{and } g(o) = A + B + C + D = 0.$$

In the main text it is to be understood that A_j , B_j , C_j and D_j are equivalent to (A.6.6) with $b = b_j$. Also, in the main text $f_j(t)$ is equivalent to (A.6.3) with (A.6.6) replaced by A_j , B_j , C_j and D_j . Finally recalling the definitions of a , b , c and d in (4.59), the mean, variance, second moment and coefficient of variation for $f_j(t)$ are:

$$\bar{t}_j = T_1 + T_{2j} + T_3 + T_4 \quad (\text{A.6.7})$$

$$\sigma_j^2 = T_1^2 + T_{2j}^2 + T_3^2 + T_4^2 \quad (\text{A.6.8})$$

$$\bar{t}_j^2 = \bar{t}_j^2 + \sigma_j^2 \quad (\text{A.6.9})$$

$$cv_j = \frac{\sigma_j}{\bar{t}_j} \quad (\text{A.6.10})$$

APPENDIX BB.1 Software Documentation

As mentioned in Chapter 1, the software development associated with the experimental work in this thesis, uses the software package developed by D. Weir as a base; thus reference [50] should be referred to in conjunction with the following documentation.

The software consists of 4 packages:

- I. A terrestrial (T) two node package consisting of the following 9 different source programs (see Figure 5.8):
 - 1) DUMMY.MAC - calling routine as in [50]
 - 2) DEMOT.MAC - control routine as in DEMO in [50], except for the calling of separate asynchronous and synchronous, multiplex and demultiplex routines.
 - 3) AMUX.MAC - asynchronous multiplex routine (see Figure B.1.1)
 - 4) ADEMUX.MAC - asynchronous demultiplex routine (see Figure B.1.2)
 - 5) SMUX.MAC - synchronous multiplex routine (see Figures B.1.3 to B.1.5)
 - 6) SDEMUX.MAC - synchronous demultiplex routine (see Figures B.1.6 and B.1.7)
 - 7) LIST.MAC - buffering routine as in [50]
 - 8) DCB45T.MAC - device control blocks as in DCB45 in [50], except for external synchronous timing, and statistical multiplexing capability for the voice terminal. (see Figures B.1.8 to B.1.13)

The load module DEM45T.SAV links the on line debugging technique - ODT, and assembled (.OBJ) versions of the above 8 routines, and runs in the PDP-11/45 as one node.

- 9) DCB40T.MAC - Same as DCB45T, except for PDP-11/40 (or 11/60) device addresses etc.

The load module DEM40T.SAV links ODT and .OBJ versions of routines 1) to 7) and 9), and runs in the PDP-11/40 (or 11/60) as another node.

- II. A single node terrestrial package involving the following 5 routines which are specialized in order to emulate admission

(A) delays (see Figure 5.15):

- 10) ADMCAL.FOR - calling routine and computation of emulated delays (see Figure B.1.14)
- 11) DEMOA.MAC - control routine as in 2), plus setup and logging of emulated delays (see Figure B.1.15)
- 12) SMUXA.MAC - synchronous multiplex routine as in 5) plus implementation of emulated delay (see Figure B.1.16)
- 13) SDMUXA.MAC - synchronous demultiplex routine as in 6) plus implementation of certain functions related to 'reservations' (see Figure B.1.17)
- 14) DCB45A.MAC - device control blocks as in 8) plus measurement and logging of message statistics.

The load module DEM45A.SAV links ODT and .OBJ versions of routines 10), 11), 3), 4), 12), 13), 7) and 14), and runs in the PDP-11/45.

- 15) ADMTST.FOR - test routine based on 10), for computing various parameters related to the emulation of delays.

- III. A single node satellite (S) package involving the following two routines (see Figure 5.20):

- 16) DEMOS.MAC - control routine as in 2), with terrestrial error control disabled, synchronization of MP processing and reception, and measurement and logging of error statistics (see Figures B.1.18 and B.1.19)

- 17) DCB45S.MAC - device control blocks as in 8), with terrestrial error control disabled, synchronization of MP processing and reception, error detection, and measurement and logging of delay and timing statistics (see Figures B.1.20 to B.1.22)

The load module DEM45S.SAV links ODT and .OBJ versions of routines 1), 16), 3) to 7) and 17), and runs in the PDP-11/45.

- 18) DLYEKO.MAC - test routine for emulating satellite propagation delay in the PDP-11/40 (or 11/60) (see Figure B.6.1)

- IV. A single node satellite package with capability of demonstrating error control by retransmission over an in house "terrestrial" link to another computer (see Figure 5.20). MP's received in error are NAK'd (N) and retransmitted (R) over the "terrestrial" link. The following 4 source routines are involved:

- 19) DEMON.MAC - control routine as in 16), plus 64 MP's repetitively transmitted, initiation of NAKing, logging of NAK and retransmit time, and number of retransmissions missed.
- 20) DCB45N.MAC - device control blocks as in 17), plus terminals disabled, 64 MP's repetitively transmitted, and capability for sending NAK's and receiving retransmissions (see Figures B.1.23 and B.1.24)

The load module DEM45N.SAV links ODT with .OBJ versions of routines 1), 19), 3) to 7) and 20), and runs in the PDP-11/45.

- 21) DEMOR.MAC - control routine as in 19), except after formation and storing of first 64 MP's, it only computes addresses of MP's to be retransmitted, simulated a terrestrial transmission delay, and initiates retransmissions.

- 22) DCB40R.MAC - device control blocks as in 20), except that after formation and storing of first 64 MP's, it only receives NAK's and sends retransmissions (see Figures B.1.23 and B.1.24)

The load module DEM40R.SAV links ODT with .OBJ versions of routines 1), 21), 3) to 7) and 22), and runs in the PDP-11/40 (or 11/60).

The following flow graphs cover the most significant additions and modifications made to the software package developed by D. Weir [50], in order to produce the 4 packages just summarized. Each flow graph is labelled by routine number and name as just summarized, and as to the specific function or functions it performs.

FIGURE B.1.1 PART OF ROUTINE 3) - AMUX:
mp HEADER (BYTE COUNT) PROCESSING

FIGURE B.1.2 PART OF ROUTINE 4) - ADEMUX:
mp HEADER (BYTE COUNT) PROCESSING

FIGURE B.1.3 ROUTINE 5) - SMUX: (1ST OF 3)
MESSAGE STATUS AND mp HEADER BIT PROCESSING

FIGURE B.1.4 ROUTINE 5) - SMUX: (2ND OF 3)
MULTIPLEXING AND mp TRAILER PROCESSING

FIGURE B.1.5 ROUTINE 5)-SMUX (LAST OF 3)

MESSAGE MANAGEMENT AND
mp PADDING CALCULATIONS

FIGURE B.1.6

ROUTINE 6) - SDEMUX:
(1ST OF 2)
MP HEADER AND TRAILER
PROCESSING

FIGURE B.1.7 ROUTINE 6) - SDEMUX:
(LAST OF 2)
DEMULTIPLEXING AND
MESSAGE MANAGEMENT

FIGURE B.1.8 PART OF ROUTINE 7) -
DCB45T: RECEIVE FROM
VADAC (1ST OF 3);
FRAMING, FRAME CHECKING
AND REFRAMING

FIGURE B.1.9
 PART OF ROUTINE 7)
 - DCB45T: RECEIVE
 FROM VADAC (2ND OF 3)
 FRAME ALIGNMENT,
 INITIAL SEQUENCE TRANSMISSION, END OF
 MESSAGE MANAGEMENT AND SIGNALING

FIGURE B.1.11
PART OF ROUTINE 7) - DCB45T:
TRANSMIT TO VADAC (1ST OF 3);
MESSAGE STATUS, ACTIVE FRAME
SEQUENCE IDENTIFICATION, MESHING
CHECK, IDLE TO ACTIVE FRAME
TRANSFER

FIGURE B.1.12
 PART OF ROUTINE 7) - DCB45T:
 TRANSMIT TO VADAC (2ND OF 3)
 END OF MESSAGE SIGNALLING CHECK,
 END OF MESSAGE FRAME SEQUENCE
 IDENTIFICATION, MESSAGE
 ALIGNMENT BEFORE OUTPUT

FIGURE B.1.13

PART OF ROUTINE 7) - DCB45T:
TRANSMIT TO VADAC (LAST OF 3)
ACTIVE TO IDLE FRAME TRANSFER,
FRAME CHECKING, CONTINUOUS
FRAME OUTPUT

FIGURE B.1.14 ROUTINE 10) - ADMCAL:
CALLING ROUTINE AND COMPUTATION
OF EMULATED DELAYS

FIGURE B.1.15 PART OF ROUTINE 11) - DEMOA:
SETUP AND LOGGING OF EMULATED DELAYS

FIGURE B.1.16 PART OF ROUTINE 12) - SMUXA: IMPLEMENTATION OF EMULATED DELAY

FIGURE B.1.17 PART OF ROUTINE 13) - SDMUXA:
"CENTRAL NODE RESERVATION" SUPPLY

FIGURE B.1.18

PART OF ROUTINE 16) - DEMOS:
 (1ST OF 2)
 TRANSMISSION ERROR ANALYSIS AND
 LOGGING OF ERROR STATISTICS

FIGURE B.1.19
PART OF ROUTINE 16) - DEMOS:
(LAST OF 2)
SYNCHRONIZATION OF SYNCHRONOUS MP
PROCESSING AND RECEPTION (LOGICALLY
IDENTICAL FOR ASYNCHRONOUS MP'S)

FIGURE B.1.20 PART OF ROUTINE 17) - DCB45S (1ST OF 3)
CLOCK SERVICE ROUTINE: TASK TIMING,
PROPAGATION AND MESSAGE DELAY ACCUMULATORS,
MP ERROR FREE INTERVAL TIMING

FIGURE B.1.21 PART OF ROUTINE 17) - DCB45S (2ND OF 3)
MP TRANSMIT SERVICE ROUTINE: MP START
AND END TIMING, INITIATION OF PROPAGATION
DELAY MEASUREMENTS

FIGURE B.1.22 PART OF ROUTINE 17) - DCB45S (LAST OF 3) MP RECEIVE SERVICE ROUTINE: MP START AND END TIMING, TERMINATION OF PROPAGATION DELAY MEASUREMENTS, ERROR DETECTION, BUFFERING FOR SYNCHRONIZATION OF MP PROCESSING & RECEPTION

FIGURE B.1.23 PART OF ROUTINE 20) - DCB45N (1ST OF 2)
 NAK TRANSMIT ROUTINE (RETRANSMIT SEND
 ROUTINE IN ROUTINE 22) - DCB40R IS NEARLY
 IDENTICAL)

FIGURE B.1.24 PART OF ROUTINE 20) - DCB45N (LAST OF 2)
 RETRANSMISSION RECEIVE ROUTINE (NAK RECEIVE ROUTINE
 IN ROUTINE 22) - DCB40R IS NEARLY IDENTICAL)

B.2 Hardware Documentation

Figure B.2.1 shows the wiring to convert the MIL-188 standard for the VADAC II, to the EIA RS-232C standard for the DEC Dull synchronous serial adapter. The VADAC V is RS-232C compatible. (see Figure 5.8).

Figure B.2.2 shows how the modem 9.6 KHz clock is used to supply a 9.6 KHz external processor clock, and 2.4 KHz clocks for the VADAC and Dull in Figure B.2.1. (see Figure 5.8).

Figure B.2.3 shows schematics of the audio adapters used in the emulation of admission delay experiments (Figure 5.15), and in the satellite experiments (Figure 5.21).

Figure B.2.4 shows the schematic for emulating transmission errors on the 9.6 kb/s link in order to test satellite extension node software (Figure B.6.1).

PIN NO. MIL-188

RS 232-C

FIGURE B.2.1 VADAC INTERFACE TO PDP-11

FIGURE B.2.2 MODEM INTERFACE AND CLOCK DISTRIBUTION

(a) BALANCED/SINGLE ENDED

(b) SINGLE ENDED/BALANCED

FIGURE B.2.3 AUDIO FREQUENCY ADAPTERS

FIGURE B.2.4 SIGNAL/NOISE COMBINER

B.3. Concatenation of Messages in Node Buffers

This problem is more likely to occur in the experimental TI-NET where the duration of a minipacket (40 ms) is longer than as proposed in the TI-NET system design (10 ms). Normally when occupied, node buffers contain 1 to 2 minipackets for the entrance node and 4 to 5 for the exit node. During retransmissions, the entrance node could instantaneously contain as many as 6 minipackets, and the exit node could be empty. Therefore concatenation is most likely to occur in the entrance node during retransmissions (a rare event), and in the exit node under normal conditions. Figures B.3.1 and B.3.2 show examples of concatenation in the entrance and exit nodes respectively. (The timing and exit node buffer contents correspond to the experimental TI-NET). Figure B.3.2 illustrates how the prevention of concatenation also tends to preserve the idle time between messages.

FIGURE B.3.1 PREVENTING CONCATENATION OF TWO MESSAGES IN THE ENTRANCE NODE BUFFER

40 MSEC SERVICE INTERVALS

FIGURE B.3.2 PREVENTING CONCATENATION OF TWO MESSAGES IN THE EXIT NODE BUFFER

B.4 Emulation of Admission Delays: Model and Speech Source

pdf's for the Emulation of Admission Delays

The pdf for the first service mechanism in Figure 5.12 is based on the Geometric pdf

$$P_1(k) = (1-\rho_1)\rho_1^k, \quad k=0,1,2,3, \dots \quad (\text{B.4.1})$$

where k indexes the number of service times of admission delay, and ρ_1 ($0 \leq \rho_1 < 1$) is a parameter related to the mean of $P_1(k)$ by

$$N_1 = \frac{\rho_1}{1-\rho_1} \quad (\text{B.4.2})$$

Let the pdf for $P_1(k)$ truncated at $k=L_1$ be $g_1(k)$, where $g_1(k)$ is properly normalized as follows:

$$\begin{aligned} \sum_{k=0}^{L_1} g_1(k) &= 1 = H_1 \sum_{k=0}^{L_1} P_1(k) \\ &= H_1 (1-\rho_1) \sum_{k=0}^{L_1} \rho_1^k \end{aligned}$$

$$= H_1 (1-\rho_1^{L_1+1})$$

$$\therefore H_1 = \frac{1}{1-\rho_1^{L_1+1}} \quad (\text{B.4.3})$$

$$\text{and } g_1(k) = \frac{(1-\rho_1)\rho_1^k}{1-\rho_1^{L_1+1}}, \quad 0 \leq k \leq L_1 \quad (\text{B.4.4})$$

Note, as $L_1 \rightarrow \infty$, $g_1(k) \rightarrow P_1(k)$.

The mean value of $g_1(k)$ is

$$N_1 = \sum_{k=0}^{L_1} k g_1(k) = \frac{1-\rho_1}{1-\rho_1^{L_1+1}} \sum_{k=0}^{L_1} k \rho_1^k$$

Using the differentiation trick [112]

$$\begin{aligned} N_1 &= \frac{1-\rho_1}{1-\rho_1^{L_1+1}} \rho_1 \frac{\partial}{\partial \rho_1} \sum_{k=0}^{L_1} \rho_1^k \\ &= \frac{(1-\rho_1)}{1-\rho_1^{L_1+1}} \rho_1 \frac{\partial}{\partial \rho_1} \frac{1-\rho_1^{L_1+1}}{1-\rho_1} \end{aligned}$$

After differentiating and simplifying,

$$N_1 = \frac{\rho_1}{1-\rho_1} \frac{1-\rho_1^{L_1} (L_1(1-\rho_1)+1)}{1-\rho_1^{L_1+1}} \quad (\text{B.4.5})$$

N_1 is in service intervals. The actual mean time T_1 in Figure 5.12, is N_1 multiplied by the service time (e.g. 40 ms in the experimental network).

The CDF for $g_1(k)$ is

$$\begin{aligned} G_1(k) &= \sum_{j=0}^k g_1(j) = \frac{1-\rho_1}{1-\rho_1^{L_1+1}} \sum_{j=0}^k \rho_1^j \\ &= \frac{1-\rho_1^{k+1}}{1-\rho_1^{L_1+1}} \quad 0 \leq k \leq L_1 \quad (\text{B.4.6}) \end{aligned}$$

Solving for k gives

$$k = \left\lceil \frac{\ln(1 - (1 - \rho_1)^{L_1 + 1}) G_1(k)}{\ln \rho_1} - 1 \right\rceil, \quad (\text{B.4.7})$$

where $\lfloor x \rfloor$ means x if x is an integer, or next smallest integer of x if x is not an integer.

The pdf for the second service mechanism in Figure 5.12 is based on the shifted Geometric pdf

$$P_2(k) = (1 - \rho_2) \rho_2^{k-1} \quad k=1, 2, 3, \dots \quad (\text{B.4.8})$$

where ρ is now related to the mean of $P_2(k)$ by

$$N_2 = \frac{1}{1 - \rho_2} \quad (\text{B.4.9})$$

Letting the truncated version be $g_2(k)$, and following the same procedure as before gives the normalizing factor

$$H_2 = \frac{1}{1 - \rho_2^{L_2}} \quad (\text{B.4.10})$$

Therefore
$$g_2(k) = \frac{(1 - \rho_2) \rho_2^{k-1}}{1 - \rho_2^{L_2}} \quad k=1, 2, 3, \dots, L_2.$$

Similarly as before the mean of $g_2(k)$ is

$$N_2 = \frac{1}{1 - \rho_2} \frac{1 - \rho_2^{L_2 - 1} (L_2 (1 - \rho_2) - \rho_2)}{1 - \rho_2^{L_2}} \quad (\text{B.4.11})$$

where $T_2 = N_2 \times$ service time. Also, the CDF for $g_2(k)$ is

$$G_2(k) = \frac{1 - \rho_2^k}{1 - \rho_2^{L_2}}, \quad 1 \leq k \leq L_2. \quad (\text{B.4.12})$$

Solving for k gives

$$k = \left\lceil \frac{\ln(1 - (1 - \rho_2^{L_2}) G_2(k))}{\ln \rho_2} \right\rceil \quad (\text{B.4.13})$$

where the notation $\lceil x \rceil$ was defined earlier for (B.4.7).

- Vocoder Analogue Input Level Considerations

The voice level into the VADAC analyzer (TX) is an important parameter. Too low a level causes the analyzer to encode the speech in the form of many segments each having a short duration; also, the speech sounds choppy as a result of the analyzer failing to detect some low level speech segments. On the other hand too high a level will tend to generate fewer segments, but with longer durations; there is also a tendency for analyzer input circuitry to saturate on high level peaks and cause perceptible distortion. The analyzer input automatic gain control (VOGAD) was not used because of uncertainty as to its operating point at the beginning of a session, and its very long (12 sec) time constant; also use of the VOGAD may give rise to spurious durations of analyzer activity during idle periods. Thus the

analyzer input voice level was adjusted to give what was judged to be the best quality, and the same setting was used throughout the tests.

- Transcript of 2½ Minute Taped Session.

"The VADAC 5 Speech Processor is a solid state, all digital, adaptive voice analyzer data converter which provides digitized speech output at switch selectable data rates of twenty-four hundred or forty-eight hundred bits per second. Twenty-four hundred bits per second permits narrowband system operation on standard voice quality circuits. The VADAC 5 Speech Processor can be used to provide a single digitized speech circuit or the output can be multiplexed with other * data bit streams to allow simultaneous voice and data transmission.

"Synchronization from an out of sync * start is obtained in a quarter of a second or less. The VADAC 5 Speech Processor can operate from a standard telephone input. In-band ringing is provided. It comes as a desk unit or is rack mounted.

"The VADAC 5 Speech Processor may be operated locally or remotely. An auxiliary connector provides the capability for remote operation. A second connector is an EIA modem connector that may be used for connection to an external multiplexer or modem.

"VADAC has a low susceptibility to high bit error rates making the VADAC 5 Speech Processor ideal for use over degraded digital communication links. Synchronization is firmly acquired with a bit error rate of one percent or better.

* Perceptible "unnatural" pauses in the clear text.

Operation at this bit error rate or less is essentially equal to operation at zero bit error rate. The VADAC 5 Speech Processor provides speaker recognition and very acceptable word intelligibility."

- Taped Session Statistics

Session Duration:

2.25 Minutes = 135 sec

= 6000 VADAC Frames

= 3375 TI-NET mp's

= 40500 bytes at 2.4 kb/s

Active Time = 76.41 sec (56.6%)

Idle Time = 58.59 sec (43.4%)

The densities of active and idle periods for the taped session were measured and the results are shown in Figures B.4.1 and B.4.2 respectively.

- Effect of Hangover on Mean Active and Idle Period Duration

The 2½ minute session involves 6000 vocoder frames, so that the mean message length can be computed from Figure 5.16 and is seen to increase with hangover. I.e. for no hangover the mean length is:

$$\frac{56.6\%}{100} \times 6000 \text{ Frames} \times 22.5 \text{ ms/Frame} \approx 170 \text{ ms.}$$

Similarly for 8 frames of hangover, the mean length is about 2 sec. The mean duration of idle periods can be found in the same way, and all results are shown in Figure B.4.3.

FIGURE B.4.1
 FREQUENCY OF ACTIVE PERIODS HAVING VARIOUS DURATIONS
 FOR
 2½ MINUTE TAPED SESSION

TOTAL NUMBER OF ACTIVE PERIODS = 447
 MEAN DURATION = 7.488 FRAMES = 168.5 ms

FIGURE B.4.2

FREQUENCY OF IDLE PERIODS HAVING VARIOUS DURATIONS
FOR
2½ MINUTE TAPED SESSION

TOTAL NUMBER OF IDLE PERIODS = 447

MEAN DURATION = 5.935 FRAMES = 133.5 ms

NO. OF IDLE PERIODS WITH DURATION EQUAL TO ABSCISSA

360

DURATION IN VOCODER FRAMES (1 FRAME = 22.5 ms)

FIGURE B.4.3

EFFECT OF HANGOVER ON MEAN ACTIVE
AND IDLE PERIOD DURATION

55

OF/DE

U.S. GOVERNMENT PRINTING OFFICE

1964 O - 700-000-000

B.5 Emulation of Admission Delays: Prepared Tape and Results

A tape was prepared in order to record the effects of various admission delays. The contents of the tape are summarized below and in Tables B.5.1 and B.5.2. Various other results were obtained as the tapes were being recorded and these are shown in Tables B.5.3 and B.5.4 and in Figures B.5.1, B.5.2 and B.5.3. The results include, average node input buffer utilization, and average and peak contents of the input buffer.

TAPE CONTENTS - SIDE A

TEST 1 Clear text (Male monologue)

TEST 2 VADAC II back to back operation

TEST 3 VADAC II through one TI-NET node on loop-back;

STDM mode

TEST 4 VADAC II through one TI-NET node on loop-back;

ATDM mode, eight frames hangover, no admission delay.

The following cases 1 to 9 are for the conditions in Test 4 above except for emulated (admission) delay and other parameters as per Table B.5.1, and Case 9 is for 5 frames of hangover.

Admission delay Case 1

Admission delay Case 2

Admission delay Case 3

Admission delay Case 4

Admission delay Case 5

Admission delay Case 6

Admission delay Case 7

Admission delay Case 8

Admission delay Case 9 (five frames hangover)

TAPE CONTENTS - SIDE B

TEST 5 VADAC II through one TI-NET node on loop-back;

ATDM mode, five frames hangover, no admission delay.

The following cases 10 to 17 are for the conditions in

Test 5 above, except for emulated (admission) delay and other parameters as per Table B.5.2.

Admission delay Case 10
Admission delay Case 11
Admission delay Case 12
Admission delay Case 13
Admission delay Case 14
Admission delay Case 15
Admission delay Case 16
Admission delay Case 17

TABLE B.5.1

EMULATED DELAY PARAMETERS

CASE	NO. HANGOVER FRAMES	RESERVATIONS /SEC FROM CENTRAL NODE	NO. OF MESSAGES	α_1	% ZERO EMULATED DELAY		AVERAGE EMULATED *DELAY			MAXIMUM EMULATED DELAY	
					PATH 1	OVERALL	PATH 1	PATH 2	OVERALL	PATH 1	PATH 2
1	8	0	56	7143	47.50	33.93	.7250	2.125	1.125	2	4
2	8	0	57	7193	29.27	21.05	1.488	4.125	2.228	4	7
3	8	0	57	7193	21.95	15.79	2.342	6.000	3.368	6	11
4	8	0	56	7143	20.00	14.27	3.100	7.813	4.446	8	14
5	8	0	54	7222	20.51	14.82	3.949	10.07	5.648	10	17
6	8	.4	54	7222	20.51	14.82	3.949	10.07	5.648	10	17
7	8	0	51	4706	29.17	13.73	2.208	5.926	4.177	6	12
8	8	0	52	1923	50.00	9.615	1.400	5.929	5.058	5	12
9	5	0	74	7432	20.00	14.87	2.327	5.947	3.257	6	11

NOTE: All times and delays are in service times (1 service time = 40 ms).

TABLE B.5.2

EMULATED DELAY PARAMETERS

CASE	NO. HANGOVER FRAMES	RESERVATIONS /SEC FROM CENTRAL NODE	NO. MESSAGES	α_1	% ZERO EMULATED DELAY		AVERAGE EMULATED DELAY			MAXIMUM EMULATED DELAY	
					PATH 1	OVERALL	PATH 1	PATH 2	OVERALL	PATH 1	PATH 2
10	5	0	70	.7286	49.02	35.70	.7059	2.053	1.071	2	4
11	5	0	70	.7286	27.45	20.00	1.471	4.000	2.157	4	7
12	5	0	68	.7353	20.00	14.71	2.300	6.056	3.294	6	11
13	5	0	69	.7391	19.61	14.49	3.059	7.889	4.319	8	14
14	5	0	72	.7361	18.87	13.89	3.943	9.421	5.389	10	17
15	5	.5319	68	.7353	18.00	13.24	3.880	9.611	5.397	10	17
16	5	.2660	71	.7324	19.23	14.09	3.846	9.421	5.338	10	17
17	5	.1773	65	.7385	18.75	13.85	3.938	9.588	5.415	10	17

NOTE: All times and delays are in service times (1 service time = 40 ms).

TABLE B.5.3

EMULATION DELAY RESULTS

CASE	MSG'S/SEC λ	CHAR/MSG μ	% ACTIVITY	% BUFFER UTILIZATION	AVERAGE BUFFER CONTENTS		MAXIMUM BUFFER CONTENTS mp's	T _q ms
					mp's	MESSAGES		
TEST 4	.3852	613.0	78.70	80.83	1.106	.02165	1.917	56.21
1	.4148	572.2	79.12	82.52	1.665	.03492	4.833	84.18
2	.4222	568.4	80.00	85.04	2.737	.05778	9.417	136.9
3	.4222	562.2	79.13	85.75	4.028	.08597	12.00	203.6
4	.4148	572.3	79.13	88.33	5.374	.1127	18.83	271.7
5	.4000	592.7	79.02	88.53	7.392	.1497	>21	374.2
6	.4000	593.5	79.13	81.27	1.319	.03282	9.167	82.08
7	.3778	628.4	79.13	85.60	4.183	.07988	16.92	211.4
8	.3852	613.7	78.80	87.41	5.319	.1040	18.42	270.0
9	.5481	414.7	75.78	84.33	3.803	.1100	15.50	200.7

TABLE B.5.4

EMULATION DELAY RESULTS

CASE	MSG'S/SEC λ	CHAR/MSG ρ	% ACTIVITY	% BUFFER UTILIZATION	AVERAGE BUFFER CONTENTS		MAXIMUM- BUFFER CONTENTS mp's	T _q ms
					mp's	MESSAGES		
TEST 5	4963	451.8	74.75	77.39	1.134	.03012	2.417	60.68
10	5185	435.1	75.20	79.02	1.626	.04484	5.917	86.49
11	5185	443.0	76.57	82.43	2.549	.06905	10.92	133.2
12	5037	449.3	75.43	83.23	3.413	.09116	12.83	181.0
13	5111	442.8	75.43	85.66	5.361	.1453	20.50	284.3
14	5333	427.1	75.93	88.53	6.382	.1793	>21	336.2
15	5037	452.4	75.97	79.26	1.415	.03848	10.75	76.40
16	5259	430.0	75.38	82.19	3.228	.09008	18.33	171.3
17	4815	468.6	75.20	83.38	4.632	.1186	>21	246.4

FIGURE B.5.1 MEAN INPUT BUFFER CONTENTS
VS
EMULATED DELAY

B.6 Satellite Link Emulation, and Supplementary Results of Satellite Link Error Measurements

Figure B.6.1 shows how transmission errors and satellite propagation delay were emulated, in order to test the satellite extension node software, before tests over the satellite link were conducted.

Figures B.6.2 to B.6.4 are MP EFIL results for fairly high MP error rates, and show the extent to which these results agree with a discretized exponential pdf. Figures B.6.5 to B.6.7 are similar results for lower MP error rates.

Figures B.6.8 to B.6.10 are results for the number of bit errors within MP's, while Figures B.6.11 to B.6.13 are for bit EFIL's within MP's.

Finally Figure B.6.14 was used to estimate the improvement in error performance provided by the forward error correction scheme shown in Figure 5.21. The performance curves without error control (uncoded) were obtained from references [132, 138], while the ideal BPSK curve with error control (coded) was obtained from [133] or [134]. The two remaining curves were obtained graphically using a procedure given in [134] whereby the coded performance of modems other than that of BPSK can be determined. The procedure is to select a value of E_b/N_0 and error rate on the ideal BPSK coded curve (e.g., $E_b/N_0 = 2\text{dB}$ at 1×10^{-2}), project up to the ideal BPSK uncoded curve (3.7×10^{-2}), across to the uncoded curve for the modem in question (e.g., $E_b/N_0 = 4.4\text{ dB}$ for the measured QPSK curve),

and down again to the original error rate (1×10^{-2}); this gives one point for the coded performance of the modem in question, as illustrated in Figure B.6.14. By repeating this procedure for other values of E_b/N_0 , the entire coded performance curve can be found.

Thus at an MP error rate of 2×10^{-3} , 144 bits per MP, and approximately 3 bits in error per damaged MP on the average, the estimated bit error rate after the Viterbi decoder is $2 \times 10^{-3} \times 3/144 = 4.2 \times 10^{-5}$. Therefore from Figure B.6.14 at 4.2×10^{-5} , $E_b/N_0 = 6.4$ dB, and the error rate before the Viterbi decoder is about 1.2×10^{-2} . This indicates an improvement in error performance of well over 2 orders of magnitude.

FIGURE B.6.1 EMULATED SATELLITE LINK

FIGURE B.6.2

DENSITY OF MP ERROR FREE INTERVAL LENGTHS

MP ERROR RATE: 2.1×10^{-2}

(DENSITY INCLUDES 324)

TOTAL MP EFIL 429

MINIMUM MP EFIL

MAXIMUM MP EFIL

MEAN MP EFIL

20 ms

6.14 SEC

0.97 SEC

DISCRETIZED EXPONENTIAL

0.1 SEC
(5 MP TIMES)

% OF MP ERROR FREE INTERVALS WITH LENGTHS
IN THE 0.1 SEC RANGES ON THE ABSCISSA

MP ERROR FREE INTERVAL LENGTHS (SEC)

(1 MP = 20 MSEC)

FIGURE B.6.3

DENSITY OF MP ERROR FREE INTERVAL LENGTHS

MP ERROR RATE: 1.4×10^{-2}

TOTAL MP EFIS 380

MINIMUM MP EFIL

< 0.1 SEC

MAXIMUM MP EFIL

10.4 SEC

MEAN MP EFIL

1.48 SEC

% OF MP ERROR FREE INTERVALS WITH LENGTHS IN THE 0.5 SEC RANGES ON THE ABSCISSA

DISCRETIZED
EXPONENTIAL

0.5 SEC
(25 MP TIMES)

MP ERROR FREE INTERVAL LENGTHS (SEC)
(1 MP = 20 MSEC)

FIGURE B.6.4

DENSITY OF MP ERROR FREE INTERVAL LENGTHS

MP ERROR RATE: 8.8×10^{-3}

TOTAL MP EFIS 202 (DENSITY INCLUDES 177)

MINIMUM MP EFIL < 0.1 SEC

MAXIMUM MP EFIL 13.8 SEC

MEAN MP EFIL 2.29 SEC

8 OF MP ERROR FREE INTERVALS WITH LENGTHS
IN THE 0.5 SEC RANGES ON THE ABSCISSADISCRETIZED
EXPONENTIAL.5 SEC
(25 MP TIMES)MP ERROR FREE INTERVAL LENGTHS (SEC)
(1 MP = 20 MSEC)

FIGURE B.6.5

DENSITY OF MP ERROR FREE INTERVAL LENGTHS

MP ERROR RATE: 3.3×10^{-3}

TOTAL MP EFIS	99	DENSITY INCLUDES 31)
MINIMUM MP EFIL		0.2 to 0.3 SEC
MAXIMUM MP EFIL		36.9 SEC
MEAN MP EFIL		6.04 SEC

OF MP ERROR FREE INTERVALS WITH LENGTHS
IN THE 0.1 SEC RANGES ON THE ABSCISSA

0.1 SEC
(5 MP TIMES)

MP ERROR FREE INTERVAL LENGTHS (SEC)
(1 MP = 20 MSEC)

FIGURE B.6.6

DENSITY OF MP ERROR FREE INTERVAL LENGTHS

MP ERROR RATE: 1.2×10^{-3}

TOTAL MP BFIS	38	(DENSITY INCLUDES 11)
MINIMUM MP BFIL	0.2 SEC	
MAXIMUM MP BFIL	94.4 SEC	
MEAN MP BFIL	16.7 SEC	

% OF MP ERROR FREE INTERVALS WITH LENGTHS
IN THE 0.1 SEC RANGES IN THE ABSCISSA

← 0.1 SEC
(5 MP TIMES)

MP ERROR FREE INTERVAL LENGTHS (SEC)
(1 MP = 20 MSEC)

1.7

1.6

1.5

1.4

1.3

1.2

1.1

1

.9

.8

.7

.6

.5

.4

.3

.2

.1

0

1

2

3

4

5

6

7

8

9

FIGURE B.6.7

DENSITY OF MP ERROR FREE INTERVAL LENGTHS

MP ERROR RATE: 9.8×10^{-4}

TOTAL MP EFIS	(DENSITY INCLUDES 4)
38	

MINIMUM	MP	EFIL	0.5 to	0.6 SEC

MAXIMUM MP	EFIL	78.2 SEC

[illegible]

FIGURE B.6.8

DENSITY OF NUMBER OF BIT ERRORS WITHIN MPs (HEADER + DATA FIELD + 128 BITS)

MP ERROR RATE 3.9×10^{-2}
 NO. OF MPs ANALYSED 945 (DENSITY INCLUDES 893)
 TOTAL ERROR EVENTS 4355
 MIN NO. ERRORS/MP* 0
 MAX NO. ERRORS/MP 65
 MEAN NO. ERRORS/MP 4.61

% OF MPs WITH BIT ERRORS EQUAL TO ABSCISSE

* NO ERRORS IN HEADER + DATA FIELD
 > 1 ERROR IN CRC 16 CHARACTERS

FIGURE B.6.9

FIGURE B.6.10

DENSITY OF NUMBER OF BIT ERRORS WITHIN
MPs (HEADER + DATA FIELD + 128 BITS)

MP ERROR RATE 1.2×10^{-3}

NO OF MPs ANALYSED	39
TOTAL ERROR EVENTS	108
MIN NO. ERROR/MP*	0
MAX NO. ERROR/MP	7
MEAN NO. ERROR/MP	2.77

% OF MPs WITH BIT ERRORS EQUAL TO ABSCISSA

0

1

2

3

4

5

6

7

NUMBER OF BIT ERRORS WITHIN MPs

* NO ERRORS IN HEADER+DATA FIELD
≥ 1 ERROR IN CRC16 CHARACTERS

FIGURE B.6.11

FIGURE B.6.12

DENSITY OF BIT ERROR FREE INTERVAL LENGTHS
WITHIN MPs (HEADER + DATA FIELD + 128 BITS)

MP ERROR RATE	1.6×10^{-3}
NO. OF MPs ANALYSED	69
TOTAL BIT EFIs	133
MIN EFIL	0 BITS
MAX EFIL	9 BITS
MEAN EFIL	1.48 BITS

% OF BIT ERROR FREE INTERVALS WITH LENGTHS EQUAL TO ABSCISSA

BIT ERROR FREE INTERVAL LENGTHS WITHIN MPs
IN BITS (1 BIT = 9600⁻¹ SEC)

FIGURE B.6.13

DENSITY OF BIT ERROR FREE INTERVAL LENGTHS
WITHIN MPs (HEADER + DATA FIELD + 128 BITS)

MP ERROR RATE	1.2×10^{-3}
NO. MPs ANALYSED	39
TOTAL BIT EFIS	72
MIN EFIL	0 BITS
MAX EFIL	15 BITS
MEAN EFIL	1.57 BITS

FIGURE B.6.14

CODED AND UNCODED HIGH SPEED MODEM PERFORMANCE

END

170879

FIN