NAVAL POSTGRADUATE SCHOOL
Monterey, California

19980803 050

DISSERTATION

CHANNEL ALLOCATION IN WIRELESS INTEGRATED SERVICES NETWORKS FOR LOW-BIT-RATE APPLICATIONS

by

Amir Uziel

June 1998

Dissertation Adviser: Murali Tummala

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 1

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1998	Ph.D. Dissertation

4. TITLE AND SUBTITLE
CHANNEL ALLOCATION IN WIRELESS INTEGRATED SERVICES NETWORKS FOR LOW-BIT-RATE APPLICATIONS

5. FUNDING NUMBERS

6. AUTHOR(S)
Uziel, Amir

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

This work addresses issues related to the design and performance of a wireless integrated services network with emphasis on a tactical framework. We propose an asynchronous transfer mode (ATM)-like protocol architecture for the mobile network, which is an extension of schemes proposed in the literature. A medium-access-control (MAC) scheme, based on slot reservation by the remotes, is proposed for the network. Traffic models for low-bit-rate applications, suitable for low-capacity channels, such as a multiple-access (macrocell) wireless network, are presented. New bi-directional speech-conversation and bursty data models are proposed.

The issue of scheduling in wireline integrated services networks is thoroughly addressed and new algorithms are proposed. An analytical scheme to obtain the required (static) capacity for homogeneous sources based on their Markov-chain characterization is provided. A necessary condition for optimality of a scheduling algorithm is the balance of cell-loss-probability (CLP) ratios to values approaching 1 from below, on the boundary of the admissible region. The balanced-CLP-ratio (BCLPR) algorithm satisfies this condition but ignores the deadlines of the cells. The shortest time to extinction (STE) with BCLPR (STEBR) algorithm, proposed here for the first time, utilizes the earliest-deadline-first concept while satisfying the necessary condition. A proof is provided to show that the STEBR decisions are optimal at each service slot given that no information about future traffic arrivals is available. Simulation results indicate that STEBR admits more sources and yields larger normalized channel throughput (by up to 4%) than STE.

14. SUBJECT TERMS	15. NUMBER OF PAGES
B-ISDN, ATM, MAC, Scheduling, Channel Allocation, Mobile Networks, Low-Bit-Rates Source Models	347
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev.2-89)
Prescribed by ANSI Std. 239-18

13. The wireless network presents a case of distributed queues at the command post (CP) and in the remotes, making channel allocation more involved compared to scheduling in wireline systems. Based on the schedulers discussed for the wireline queue, corresponding algorithms for operation in the wireless network are developed. The cases of partial and complete status reports of the remotes are investigated as a function of the network load in five representative scenarios. The following (descending) order of performance under both partial and complete status reports is maintained in all scenarios: STEBR, STE, BCLPR, and static allocation. Performance of the schedulers using partial or complete status reports depends on the value of the normalized throughput. The complete-status mechanism is preferred whenever the normalized throughput is smaller than 0.70-0.75; partial status reports are sufficient for normalized throughput larger than 0.70-0.75. A hybrid approach that makes use of this outcome is proposed to best utilize the available channel capacity under all possible levels of network load.

Approved for public release; distribution is unlimited

CHANNEL ALLOCATION IN WIRELESS INTEGRATED SERVICES NETWORKS FOR LOW-BIT-RATE APPLICATIONS

Amir Uziel
Major, Israeli Army
B.Sc., Tel-Aviv University, 1988

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

June 1998

Author: _____
Amir Uziel

Approved by: _____ _____
Herschel H. Loomis, Jr.
Professor of Electrical and Computer Engineering

Gus K. Lott, Jr.
Assistant Professor of Electrical and Computer Engineering

_____ _____
Gilbert M. Lundy
Associate Professor of Computer Science

Craig Rasmussen
Associate Professor of Mathematics

_____
Murali Tummala
Professor of Electrical and Computer Engineering
Dissertation Supervisor

Approved by: _____
Herschel H. Loomis, Jr., Chairman
Department of Electrical and Computer Engineering

Approved by: _____
Maurice D. Weir, Associate Provost for Instruction

ABSTRACT

This work addresses issues related to the design and performance of a wireless integrated services network with emphasis on a tactical framework. We propose an asynchronous transfer mode (ATM)-like protocol architecture for the mobile network, which is an extension of schemes proposed in the literature. A medium-access-control (MAC) scheme, based on slot reservation by the remotes, is proposed for the network. Traffic models for low-bit-rate applications, suitable for low-capacity channels, such as a multiple-access (macrocell) wireless network, are presented. New bi-directional speech-conversation and bursty data models are proposed.

The issue of scheduling in wireline integrated services networks is thoroughly addressed and new algorithms are proposed. An analytical scheme to obtain the required (static) capacity for homogeneous sources based on their Markov-chain characterization is provided. A necessary condition for optimality of a scheduling algorithm is the balance of cell-loss-probability (CLP) ratios to values approaching 1 from below, on the boundary of the admissible region. The balanced-CLP-ratio (BCLPR) algorithm satisfies this condition but ignores the deadlines of the cells. The shortest time to extinction (STE) with BCLPR (STEBR) algorithm, proposed here for the first time, utilizes the earliest-deadline-first concept while satisfying the necessary condition. A proof is provided to show that the STEBR decisions are optimal at each service slot given that no information about future traffic arrivals is available. Simulation results indicate that STEBR admits more sources and yields larger normalized channel throughput (by up to 4%) than STE.

The wireless network presents a case of distributed queues at the command post (CP) and in the remotes, making channel allocation more involved compared to scheduling in wireline systems. Based on the schedulers discussed for the wireline queue, corresponding algorithms for operation in the wireless network are developed. The cases of partial and complete status reports of the remotes are investigated as a function of the network load in five representative scenarios. The following (descending) order of performance under both partial and complete status reports is maintained in all scenarios:

STEBR, STE, BCLPR, and static allocation. Performance of the schedulers using partial or complete status reports depends on the value of the normalized throughput. The complete-status mechanism is preferred whenever the normalized throughput is smaller than 0.70-0.75; partial status reports are sufficient for normalized throughput larger than 0.70-0.75. A hybrid approach that makes use of this outcome is proposed to best utilize the available channel capacity under all possible levels of network load.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND NOTATIONS

$\lfloor \bullet \rfloor$	The integer value smaller than/equal to •
$\lceil \bullet \rceil$	The integer value larger than/equal to •
$\lvert\{\bullet\}\rvert$	Number of elements within the set {•}
$\bullet^A$	Parameter • of an aggregate source
δ	Slope of the burst region in a multiplexer
$\delta(i)$	Delta function at state i
λ_i	Cell arrival rate (of a Markov chain) at state i
$E\{\lambda\}$	Mean rate of cells, successfully received in the mobile network (in cells/sec)
π_i	Steady-state probability (of a Markov chain) of state i
ρ_M	Mobile-channel utilization
ρ_W	Wireline-link utilization
A	Coefficient of loss probability in the burst region
$A[i]$	Number of cells arrived thus far from source i
$AS[i]$	Number of slots allocated thus far to source i
B_T	Number of bins (arrival levels) of source of type T
C_M	Mobile-channel capacity (in cells/sec)
C_T	Server capacity of a homogeneous-class buffer (in cells/sec)
C_W	Wireline-link capacity of a multiplexer (in cells/sec)
$CLP[i]$	Cell loss probability of source i
$CLPR[i]$	Cell-loss-probability ratio of source i
$CRC[i]$	Static capacity required by source i (in cells/sec)
CRC_T	Total static capacity required by all sources
$CRS[i]$	Static capacity required by source i (in information slots/MAC frame)
$DC[i]$	Number of cells discarded thus far at the CP from source i
$DR[i]$	Number of cells discarded thus far at the remote from source i
$DS[i]$	Number of cells discarded thus far from source i (end to end)
$EC[j]$	Number of cells expired thus far from node j (as known at the CP)
f	Objective function in scheduling in integrated services networks
F	Number of MAC frames per second
G	Mobile-channel load
$\{G_k\}$	Set of cells having deadline to beginning of service within $[T_k, T_{k+1})$
H	Hurst parameter
K	Finite buffer size
K_0	Maximum buffer size at which slopes of cell and burst regions equate
L	Latest slot at which the queued cell having the largest deadline to beginning of service can be serviced
$MCLPR[j]$	Mean cell-loss-probability ratio of station j
N	Occupancy of a wireline queue
N_C	Number of distinct classes of traffic
N_{CD}	Number of control slots on downlink subchannel

N_{CU}	Number of control slots on uplink subchannel
N_{ID}	Number of information slots on downlink subchannel
N_{IS}	Total number of information slots in the MAC frame
N_{IU}	Number of information slots on uplink subchannel
N_L	Number of cells left in the queue after expired cells have been discarded
N_{SS}	Number of service slots that can be used for allocation
N_T	Number of sources of type T
P	State-transition probability matrix
Q	Infinitesimal generating matrix (of a continuos-time Markov chain)
R_D	Arrival rate of a single active data source
R_S	Arrival rate of a single active speech source
$RS[j]$	Number of cells requiring service on the next frame by node j
RS_T	Number of cells requiring service on the next frame by all nodes
$\{S_D\}$	A subset of $\{S_S\}$ from which a cell should be discarded
$\{S_{ES}\}$	Set of eligible sources in a queue
S_M	Mobile-channel throughput
$\overline{S}_M$	Normalized mobile-channel throughput
$\{S_N\}$	Set of active nodes in the network
$\{S_S\}$	Set of active sources in the network
S_W	Wireline-channel throughput
$\overline{S}_W$	Normalized wireline-channel throughput
T	Source type (S - speech, V - video, or D - data)
T_i	Service slot i (T_0 is current slot)
T_L	Deadline to beginning of service of the last cell in the queue
$W[i]$	Number of cells waiting from source i.
x	Variable buffer length
Y	A scheduling algorithm assumed to outperform STEBR algorithm
$Z[i]$	Counter of static allocation for source i (in cells/sec)

ACRONYMS AND ABBREVIATIONS

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACK	ACKnowledgement
ACLP	Allowed Cell Loss Probability
AGC	Automatic Gain Controller
ANSI	American National Standards Institute
AP	Access Point
APT	ATM Payload Type
ARQ	Automatic Repeat reQuest
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
BCLPR	Balanced Cell-Loss-Probability Ratio
C-MAC	Core-Medium Access Control
CAC	Connection Admission Control
CBR	Constant Bit Rate
CDMA	Code Division Multiple Access
CDV	Cell Delay Variation
CI	Channel Indicator
CLP	Cell Loss Priority or Cell Loss Probability
CLPR	Cell-Loss-Probability Ratio
CLR	Cell Loss Ratio
CP	Command Post
CRC-16	Cyclic Redundancy Code (16-bit)
DBS	Deadline to Beginning of Service
DL	Data Link
DLC	Data Link Control
EMUI	Extended Mobile User Identifier
FCFS	First Come First Serve
FECC	Forward Error Correction Code
IEEE	Institute of the Electrical and Electronic Engineers
ICLP	Instantaneous Cell Loss Probability
IDVC	IDentity within the Virtual Channel
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ITU	International Telecommunication Union
LAN	Local Area Network
LSB	Least Significant Bit
LOS	Line Of Sight
MAC	Medium Access Control
maxCTD	maximum Cell Transfer Delay
MATM	Mobile ATM

MCC	Mobile Communication Controller
MCLPR	Mean Cell-Loss-Probability Ratio
MMPP	Markov Modulated Poisson Process
MNC	Mobile Network Coordinator
MPT	Mobile Payload Type
MSB	Most Significant Bit
MSI	Mobile Signaling Identifier
MSN	Message Sequential Number
MST	Mobile Signaling Type
MSVCI	Mobile Signaling Virtual Channel Identifier
MUI	Mobile User Identifier
MVC	Mobile Virtual Channel
MVCI	Mobile Virtual Channel Identifier
NACK	Negative ACKnowledgement
NRT	Non-Real Time
OSI	Open System Interconnection
PAR	Piggyback Allocation Request
PCM	Pulse Code Modulation
PCN	Personal Communication Network
PCS	Personal Communication Services
PDU	Protocol Data Unit
PMAC	Primary Medium Access Control
PN	Pseudo Noise
PRMA	Packet Reservation Multiple Access
P2P DLC	Point-To-Point Data Link Control
QoS	Quality of Service
RF	Radio Frequency
RT	Real Time
S-MAC	Supervisory-Medium Access Control
SC	Segment Counter
SMAC	Secondary Medium Access Control
SPS	Static Priority Scheme
SSMI	Standard ATM Signaling Mobile Identifier
SSMVCI	Standard ATM Signaling Mobile Virtual Channel Identifier
STE	Shortest Time to Extinction
STEBR	Shortest Time to Extinction with Balanced cell-loss-probability Ratio
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
ToE	Time of Expiry
UBR	Unspecified Bit Rate
UNI	User Network Interface
UPC	User Parameter Control
VBR	Variable Bit Rate
VC	Virtual Channel

VCI	Virtual Channel Identifier
VP	Virtual Path
VPI	Virtual Path Identifier
WATM	Wireless ATM

ACKNOWLEDGEMENT

I devote this work to my parents, Hedva and Mordechai Uziel. Their love, support, and encouragement to pursue high education have been an inspiration to me throughout the course of my entire life.

Being approved to study at NPS was not a trivial process. My great appreciation to senior personnel in the Israeli Defense Forces Signal Corps who have placed faith in my ability to conduct advanced research and lead technical staffs in years to come.

My heartiest thanks to my adviser, Prof. Murali Tummala, for being there for me whenever needed. My gratefulness for his guidance, advice, and listening, supplying the newest equipment possible, endless careful revisions of this work, and friendship.

The dissertation committee members, Professors Hershel Loomis, Gus Lott, Bert Lundy, and Craig Rasmussen – not only have I been honored to attend their classes and acquire the highest-education possible, but their professionalism and thoroughness made me work harder and become more professional.

The proof of the STEBR algorithm was a high hurdle to overcome. I was fortunate to have many fruitful discussions with Craig Rasmussen regarding various properties of the algorithm. My special thanks to my colleauge and friend, Major Eitan Israeli from the Israeli Air Force, currently a Ph.D. candidate in the OR dept. at NPS, for sparing the time to discuss scheduling algorithms and proofs with me.

Throughout my stay at NPS and during the research stage specifically, I have received outstanding technical support from the ECE-Dept. technical staff. The help I got from Colin Cooper, Elaine Kodres, Gary Rodeske, Janice Sheldon, and David Schaeffer in installing and maintaining numerous versions of OPNET, the first Solaris workstation in the dept., various printers, hard drives, backups, plus the countless replies to other requests that I had, have really been exceptional. I wish to thank them for their time.

Finally, I would like to acknowledge the omnipresent support and understanding from the most important one in my life, my wife Ronit. Words cannot express the gratitude I have for her patience, support, and love.

I. INTRODUCTION

A. BACKGROUND

The field of modern telecommunications is being rapidly redefined by events surrounding two emerging concepts: broadband networking and personal communication services (PCS). Broadband networks are characterized by packet-based transport, bandwidth upon demand, and multimedia-traffic integration. All types of telecommunication traffic (voice, data, image, video) are carried by a broadband network using a common fixed-length packet format, and the only distinction between low-rate and high-rate connections is the rate at which such fixed-length packets are generated. Network resources are statistically shared among users and are allocated only when packets are actually available for transfer.

The PCS networks are developed with a goal to provide voice and moderate-rate (a few kbps to several hundred kbps) data services through a lightweight, portable, personal communication system that has a battery life of several hours. Some practical limitations are imposed on such networks: low quality of radio channels and a limited number of channels (and hence users) available within a given coverage area.

The principal purpose of a wireline or wireless network is to allow the users to exchange information using the network nodes. A network node generally includes a single server with one or more outgoing links, servicing a buffer with multiple entries from several local traffic sources or other network streams. Given a node's outgoing links and their capacities, the server needs to schedule the transmission of packets from the queue onto one or more of its outputs. This process is referred to in the literature as *channel allocation* or *scheduling*.

1. Broadband Integrated Networks and Channel Allocation

Three generations of network technologies have been recognized in the literature. The dedicated voice networks were the first generation. The packet-switched or the

combined voice and data networks are considered the second generation. The third-generation networks, aimed at providing multiple services, are a result of digitization of the public telephone networks worldwide as well as the concurrent deployment of the optical fiber. The current technology potentially provides much-higher channel data rates than ever before. With the availability of high bit rates, it is natural to consider new user services with high-bandwidth requirements. Primary candidates include applications involving high-resolution images and video. The next step is to merge the different services currently provided by different networks (e.g., public telephone networks for voice, Internet, X.25 networks for packet data, cable TV, etc.) into one common network. Activity has thus begun in earnest, worldwide, to develop broadband integrated services digital networks (B-ISDN) capable of accommodating the multiple services just described.

Asynchronous transfer mode (ATM) technology is currently viewed as the next high-speed integrated-network paradigm. ATM supports different classes of traffic and can be deployed in both local- and wide-area-network environments. At this time, ATM is widely considered the standard technology for the network backbone infrastructure, implementing B-ISDN such that all communication services are integrated into a single universal system.

In wireline ATM networks, a node is a switching element while wireless ATM channels include nodes of type store and forward. A node in a wireline network establishes and releases connections between its inputs and outputs and handles traffic transfer between source and destination end-user entities. The fixed-size information unit or *cell* results in equally-spaced instants for cell service.

The concept of quality of service (QoS) in B-ISDN forces allocation of the channel in a manner that guarantees the agreed-upon grade of service to all active streams of information. If a connection cannot be guaranteed with the desired QoS level throughout its lifetime, the call is not admitted into the network at all. Channel-allocation schemes differ from each other by their decision policies that determine the node's

efficiency, i.e., the number of information streams it can handle simultaneously, given a fixed outgoing channel capacity.

2. Wireless Integrated Services Networks

The convergence of mobile communications, computing, and ATM gives rise to the wireless integrated services networks. While ATM helps to bring multimedia to the desktop, wireless ATM provides similar services to mobile computers and devices while integrating seamlessly into the B-ISDN. Although bandwidth provided by the existing mobile phone systems is sufficient for voice and data traffic, it is insufficient to support real-time multimedia traffic. Wireless integrated services networks, on the other hand, aim to support higher bit rates and fast handovers. The ATM notion of virtual channels (VCs) with specified QoS can also be applied to the wireless medium by requiring the medium-access-control (MAC) layer to allocate the channel among the active traffic streams. (Nevertheless, one must keep in mind that this notion is applied in a qualitative sense due to the higher cell loss probability in wireless channels.)

In summary, the need for seamless integration into B-ISDN, the presence of a form of QoS specification, the limitations associated with existing mobile phone and wireless local-area-network (LAN) systems, and the need to provide mobile multimedia services, all explain the desire for wireless integrated services networks.

3. Military Wireless Networks

Similar to the civilian market, the military environment is characterized by a multiplicity of networks at all echelons. Traditional voice networks for operations, supply, fire control, intelligence, etc. exist together with recently-developed, packet-switched, data networks for database retrieval, command assignments, and mail. Dedicated networks for transmission of still video for a look over the horizon or geographical location data (using global positioning system) can also be found in some tactical systems. The large number of such existing networks not only requires the use of a lot of different equipment (radio transceivers, modems, multiplexers, etc.), but also results in logistical difficulties. Installation, maintenance, and training (of users and

technicians) of these disparate networks to provide a satisfactory level of services are daunting tasks. These factors, for various reasons, lead to a relatively low quality of information transfer. As a result, there is a need for developing an overall effective solution to carry out reliable information transfer and smooth network operation.

B. PROBLEM STATEMENT AND DISSERTATION OBJECTIVES

The design of a mobile ATM (MATM) network, which expands the emerging, stationary, B-ISDN technology to the wireless medium, imposes several limitations. The existence of wireline ATM networks, predicted to be ubiquitous, makes it desirable to have a mobile architecture that is ATM-like. Second, the system is required to operate in a wireless network environment, where impairments, such as multipath and noise, limit the available channel bit rate and constrain the network scalability and performance. The implications of these phenomena are that user applications may be limited to low bit rates and the number of active connections is restricted. Third, allocation of the radio channel among the mobile nodes cannot be optimal[1] due to their physically-distributed locations and their distinct QoS requirements. At a given node, instantaneous queue status of other stations is not available at all times, and transmission of status information can occupy a large portion of the channel bandwidth. Additionally, the QoS requirements of the traffic streams, which the network must satisfy, impose constraints on the mean capacity that can be utilized. The problem of allocation in the wireless network is not to achieve maximum throughput or minimum delay as traditionally considered in multiple-access channels. Wireless integrated services networks aim at maximizing the number of user services such that the QoS requirements of all of them are met; this implies that a portion of the channel capacity must be left unused to account for possible instantaneous congestion. Achieving this goal implicitly means that maximum channel throughput is obtained under the service constraints while keeping the loss and delay for each source below the allowed bounds.

---

[1] Unless *all* traffic streams are of constant bit rate.

The goal of this work is to design and develop a portion of a wireless integrated services network, suitable for military use. The first step is to characterize the environment in which the system is required to operate. It is desirable to extend wireline ATM networks to the wireless medium in a seamless fashion with minimal changes in the architecture. The wireless implementation should endeavor to make use of the existing ATM-layer system architecture. An essential portion of this architecture is the MAC protocol, which determines the guidelines that every transmission in the wireless channel must follow. It is desirable to obtain a mechanism that achieves a high level of utilization for a given channel capacity. In addition, a small portion of the channel capacity, for in-band system signaling, is required for proper operation of the network. A detailed description of a MAC protocol that supports integrated services is not available in the literature while the issue of signaling support is often ignored. Following the development of the system architecture, the objective is to develop a MAC that achieves efficiency and supports signaling as described above.

Efficient use of the limited-available channel bit rate while maintaining the QoS for all the sources is desired. The scheduling function within the MAC, allocating the channel among the nodes in the mobile integrated services network, is responsible for achieving this goal. The nodes in the network operate in a distributed manner, making the scheduling problem difficult to solve since status information regarding the queue occupancy and/or cell deadlines is not always readily available to the scheduler. There are several objectives in relation to channel allocation in MATM networks. First, we aim to obtain an optimal scheduling algorithm for a single-queue single-server system. This case, representing a situation in which all active sources are located at one station, allows clear descriptions of various channel-allocation algorithms and their properties. The second objective is to develop an optimal scheduling algorithm for operation over a mobile network. This includes design and implementation of the algorithm that fits the structure of the proposed MAC as well as adaptation of the MAC to include status reports by remote stations. Finally, in order to compare the performance of several channel-allocation algorithms, representative scenarios must be defined. Most existing traffic

models for integrated services networks were designed for high-speed wireline links, making them unsuitable for low-capacity wireless channels. We thus aim at developing low-bit-rate traffic models for three major applications generating information: bi-directional speech conversation, low-bit-rate video, and bursty data.

C. DISSERTATION OVERVIEW

The dissertation is organized as follows. We begin with a general description of the mobile station and the wireless network architectures, continue with the design of the MAC layer in the mobile node, and proceed to develop and implement several channel-allocation algorithms for wireless integrated services channels.

Chapter II begins with an overview of existing ATM environment and MAC protocols in wireless integrated services networks, followed by a description of the proposed system environment. The chapter concludes with a proposed architecture for mobile ATM networks. Chapter III details design requirements of the MAC in mobile integrated services networks, followed by description of MAC structure and protocol data units, signaling procedures in the mobile network supported by the MAC, and performance-related issues.

Chapter IV proposes models for low-bit-rate user services (applications) with their QoS requirements that are essential for system simulation in the following chapters.

Chapter V thoroughly addresses the issue of scheduling in a representative wireline ATM network node, such as a multiplexer. Several channel-allocation algorithms are described and their properties explored. Simulation results comparing the performance of different algorithms are presented and discussed.

In Chapter VI, we discuss the design and implementation of the scheduling schemes, presented in Chapter V, for the wireless channel. Chapter VII presents empirical performance results of the channel-allocation techniques for the wireless network.

Chapter VIII summarizes the significant contributions made in the dissertation and provides concluding remarks along with a discussion on possible, future, research topics in the area of wireline and mobile integrated services networks.

Appendices A through D include examples and technical information related to this work. Appendix A demonstrates a representative database of a mobile station and communication processes within a node and between nodes. Appendix B discusses self-similar random processes. Appendix C lists Matlab™[2] code used to produce graphs in this work, and Appendix D details inputs and outputs of the wireless-channel simulation program.

[2] Matlab is a registered trademark of MathWorks, Inc.

II. MOBILE SYSTEM CONFIGURATION

Recently, the topic of "mobile ATM" has received considerable interest. Researchers as well as service providers have explored the feasibility of extending ATM-like virtual connectivity from the wireline to the wireless domain in a seamless fashion [2]. The ATM technology can also be extended to meet the needs of tactical networks, especially by combining separate networks for voice and data into one integrated network, which leads to simplification of network management.

In wireline ATM, links between network nodes are of point-to-point type. Error control performed on these links is minimal due to the high reliability of the underlying, fiber-optic, physical medium [25]. In mobile ATM, on the other hand, it is vital to add some error-control capabilities as harsh radio environment causes a large rate of channel errors. In a layered network architecture, the data-link (DL) layer performs two important functions that impact the performance of a multiple-access radio network [86]. The first function (generally known as medium access control) consists of allocating transmission time slots and sending the data frames to the physical layer for actual transmission in the shared channel. The second function (data link control) ensures successful transmission of data over the noisy radio channel. This can involve segmentation of higher-layer messages into smaller frames, application of forward error correction code (FECC) on the data, transmission of frame acknowledgment messages for successfully-received frames, and retransmission in case an acknowledgment is not received.

In addition to the data link control and medium access control, the protocol architecture in mobile ATM includes also a wireless control plane [78], which performs three major functions. The connection admission controller decides whether to accept or reject new calls, taking into consideration the ATM traffic descriptors and the QoS required for all connections, including the new one. The cell handover controller allows mobile users to roam between cells in a cellular-based network. The ID-assignment function handles station mobility as remotes join or leave the cell.

In this chapter, we propose a scheme for a data-link-control layer for central, (single cell), wireless, integrated services networks. The proposed scheme attempts to provide a comprehensive mechanism that performs several functions. It efficiently allocates transmission time as a function of the active ATM-oriented services, ensures reliable transmission over impaired radio links, determines whether to accept or reject a request for a new connection such that the QoS of the existing connections will be maintained, and handles ID assignment. Node priority at the time of admission is provided by the network according to the various levels of priority in the military hierarchy. Cell handoff capability is beyond the scope of this work.

Section A includes a brief overview of wireline ATM concepts and terminology. Then in Section B, the essence of MAC in mobile integrated services networks is addressed with emphasis on packet-radio networks. The section also includes a survey on related work in the area of mobile ATM. The environment in which the system is to operate in (i.e., its topology, supported services, etc.) is presented in Section C. System architecture and protocol layering pertinent to mobile integrated services networks are discussed in Sections D and E, respectively. In Section F, a scheme for data-link-control (DLC) layer in tactical mobile integrated services networks is proposed.

A. ATM ENVIRONMENT

Asynchronous transfer mode is an asynchronous, time-division, multiplexed scheme, which is designed to utilize the high-bandwidth low-noise (thus low-loss) characteristics of the fiber-optic transmission medium [98]. ATM makes use of short, fixed-length packets or *cells* (53 octets each) that are asynchronously multiplexed into slots, thus allowing the bandwidth to be allocated in fixed or variable-size frames. The small cell size combined with large bandwidths (hundreds of megabits per second) used by ATM results in a large number of cells being in transit at any given time between end users. The ATM cell traffic from many existing (low- and high-speed) user applications can be merged together at the nearest user to network interface (UNI) to facilitate multimedia or integrated services networking. Numerous standards bodies (e.g., ANSI

T1, ITU-T SG13, and The ATM Forum) have chosen ATM as the underlying transport technology for broadband integrated services digital networks (B-ISDNs) [13].

1. Architecture

The B-ISDN protocol-reference model defined in ITU-T Recommendation I.121 is shown in Figure II.1 [40] [8]. It is divided into multiple planes: user plane (transfer of user information), control plane (establishment and termination of calls), and management plane (management functions).

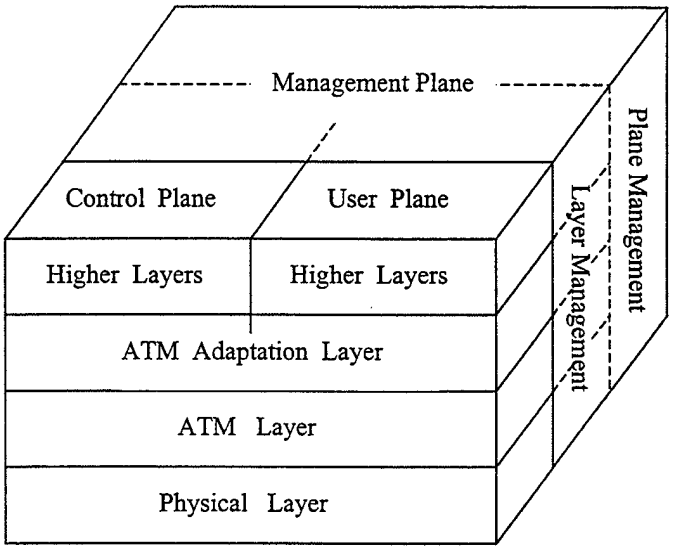


Figure II.1: B-ISDN Protocol Reference Model [40]

The physical layer, used by all planes to carry out digital information, is mostly based on the synchronous optical network (SONET) standard [8]. It is currently specified for rates starting at 51.84 Mbps and 155.52 Mbps, and up to 9953.28 Mbps with plans to expand to higher bit rates in the future. The physical layer includes two sublayers: physical-media *dependent* sublayer and transmission-convergence *independent* sublayer. The former deals with aspects that are dependent on the transmission medium selected (e.g., bit timing and line coding). The latter handles issues that are independent of the

transmission medium characteristics, such as error control or determination of cell boundaries in the physical-layer payload.

The ATM layer provides transparent transfer of cells between (higher-layer) communicating entities. This transfer occurs over a pre-established VC according to a traffic contract agreed upon by the user and the network. The ATM layer supports multiplexing of connections with different QoS requirements [8]. A cell-loss-priority (CLP) bit is used in the header of ATM cells for selective cell discarding in network equipment.

The ATM adaptation layer (AAL) enhances the service provided by the ATM layer to a level required by the next higher layer. The functions performed by the AAL depend on the higher-layer requirements: segmentation and reassembly, message identification, time/clock recovery, etc. Four classes of adaptation layers are defined, differing from each other by one or more of the following connection characteristics: timing relation between source and destination, bit rate, and connection mode [25].

The control plane mainly contains traffic control and signaling. It aims to protect the network and the user in order to achieve guaranteed QoS for all active VCs and optimize the usage of network resources [8]. The functions performed by the control plane include connection admission control (CAC) (whether to accept or reject a call), usage parameter control (UPC) (traffic monitoring and control), cell-loss-priority control (selective discarding of cells with low priority), traffic shaping, etc.

The management plane is used to maintain the network and perform operational functions [25]. It also provides the mechanisms for information exchange between the user and the control planes. The management plane includes both layer and plane management [64]. The former handles management of layer-related functions (e.g., protocol irregularities) while the latter coordinates system functions required for proper system operation.

2. Service Classes

As B-ISDN is intended to support a wide variety of services, it is convenient to separate services into classes and apply specific control mechanisms (e.g., congestion

control) and traffic-control procedures to each class. Basically, five service categories were defined by The ATM Forum at the ATM layer [9]:

- Constant bit rate (CBR) provides a virtual fixed-bandwidth circuit. Real time applications with fixed bandwidth, such as telephony, may use CBR [9].

- Variable bit rate (VBR) is intended for bursty traffic with precisely defined throughput requirements [29]. It exploits statistical multiplexing to improve the efficiency of the network. VBR is further divided into two subclasses, real time (RT) and non-real time (NRT), depending on whether or not the application is sensitive to cell delay variation. An example of a RT-VBR source is interactive compressed video whereas multimedia email is an example of a NRT-VBR source.

- Available bit rate (ABR) makes use of any available bandwidth, but the user can specify a minimum cell rate (for example, the minimum bandwidth required to keep an application running [29]). File transfer may characterize this class.

- In unspecified bit rate (UBR), any leftover capacity is used by delay tolerant applications but without any guarantee in terms of QoS. An example is a file transfer submitted as a background job.

The AAL may enhance the service provided by the ATM layer to the requirements of a specific source (ITU-T Recommendation I.362) [25]. These can be user applications as well as control and management functions. Three criteria have been defined for adaptation of higher layers: timing between source and destination (required or not), bit rate (constant or variable), and connection mode (connection oriented or connectionless oriented). Four classes out of the eight adaptation possibilities have been recognized to cover most of the existing and future service requirements by applications.

3. Quality of Service (QoS)

A typical ATM user connection consists of three stages: call setup, information exchange or transfer, and call release. At the connection-setup stage, a contract is established between the user and the network, where the user specifies the traffic descriptors of the source and the desired quality of service [9]. Commonly-used traffic descriptors are the peak cell rate, maximum burst size of cells, and minimum cell rate. The QoS is evaluated in terms of one or more of cell loss ratio (CLR) (or cell loss

probability (CLP)), maximum cell transfer delay (maxCTD), and cell delay variation (CDV). For example, the contract can include a maximum delay of 150 milliseconds, no more than 0.1% (10^{-3}) of the time. Based on these parameters, the CAC function has to decide whether to accept or reject the connection.

4. Congestion Control

Congestion control is needed to fairly allocate network resources (e.g., bandwidth, buffers and processing power) to the various and mixed types of services [24]. The objective is to implement a congestion control mechanism that is simple, effective, fair, and not optimized for any specific service type. In ATM networks, three congestion control techniques are usually implemented: admission control, access control, and traffic shaping.

a. Admission Control

Admission control determines the amount of traffic the network can handle given the QoS requirements for each traffic class. As a new call with pre-defined service parameters arrives, the admission controller decides to accept it if the network has sufficient resources to satisfy the service requirements of all connections (including the new one); otherwise, the call is rejected. Several admission-control policies based on traffic occupancy have been proposed in the literature [80]. Of these, most utilize only the instantaneous knowledge of traffic at the access point (AP) as shown in Figure II.2. (Virtual path (VP) is a group of virtual connections sharing a segment of the network.)

An admission-control policy usually goes hand-in-hand with the *bandwidth-allocation* (or *scheduling*) scheme used in the multiplexing buffer. The location of this buffer in an ATM network is indicated in Figure II.2. In Figure II.3, a representation of a typical access node (also called processor sharing node [47]) is depicted; each arriving cell enters one of the N_C queues representing N_C distinct classes of service. A scheduler (having a fixed capacity of C_W cells/sec) services one cell at a time, following a pre-defined scheduling algorithm. The order of service in the buffer affects the utilization of the system and influences the number of calls that may be admitted.

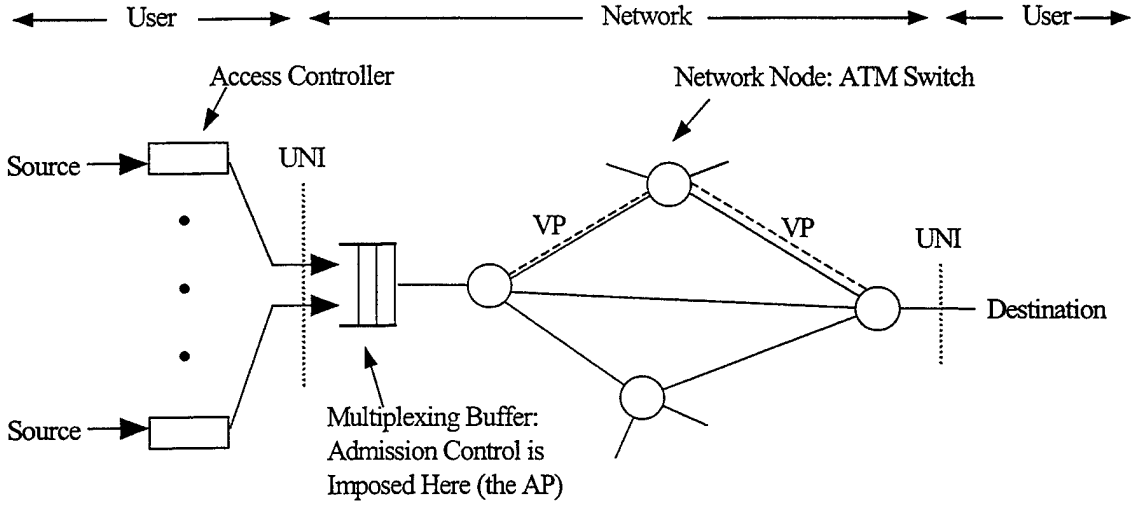


Figure II.2: General Structure of an ATM Network [80]

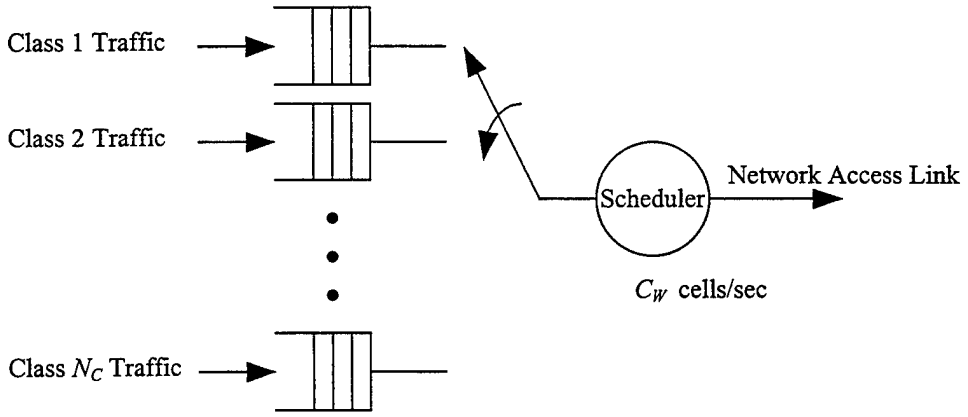


Figure II.3: Scheduling at the Access Node [47]

A simple scheduling solution would be to allocate a specific bandwidth to each VP along its complete path, end to end, based on the peak-rate requirement [23]. Such a solution is quite simple but nullifies the advantage of statistical multiplexing, thus requiring possibly much higher bandwidth than may otherwise be the case. This scheme provides only a lower bound to the number of possible calls in the system.

A first-come-first-serve (FCFS) scheme services the incoming traffic in a buffer according to arrival time. The maximum number of connections possible here depends on the available channel capacity, the number of traffic classes in the buffer, and the QoS required for each. The set of combinations of the maximum numbers of traffic-class users determines the boundary of the so-called *admissible region* of the system. Any combination of sources that lies within the admissible region guarantees that the desired QoS for *all* sources is met. On the other hand, any combination outside the region would cause violation of agreed QoS for at least one source. A call is admitted into the network if the system, with the new call, still operates in the admissible region; otherwise, the call is blocked. The FCFS scheme, where cells from several service classes are multiplexed at the AP, results in a less than optimal admissible region as can be seen in the example shown in Figure II.4 for a two-class case [80].

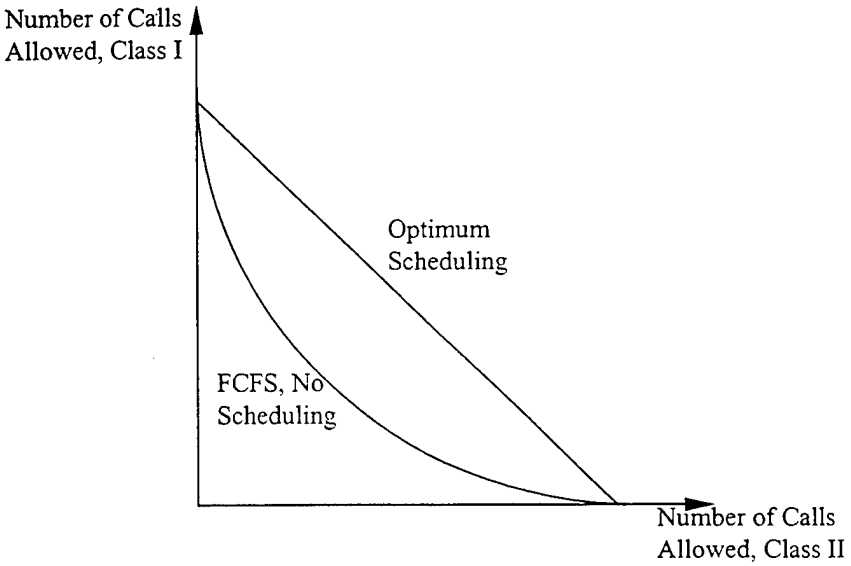


Figure II.4: Two-Class Admissible Region [80]

Admission-control policy based on the average rate of transmission provides a maximum multiplexing gain. However, it assumes a perfect utilization of the link and does not consider the statistical fluctuations of the sources, thus resulting in high cell loss probability. This scheme is useful to obtain an upper bound on the number of

calls that can be admitted. Admission-control policies are then expected to operate within the bounds provided by peak-rate and mean-rate bandwidth allocations as shown in Figure II.5; for example, n_2 represents the number of active sources that could be admitted or the width of the admissible region when a hypothetical, realistic, admission-control policy is used.

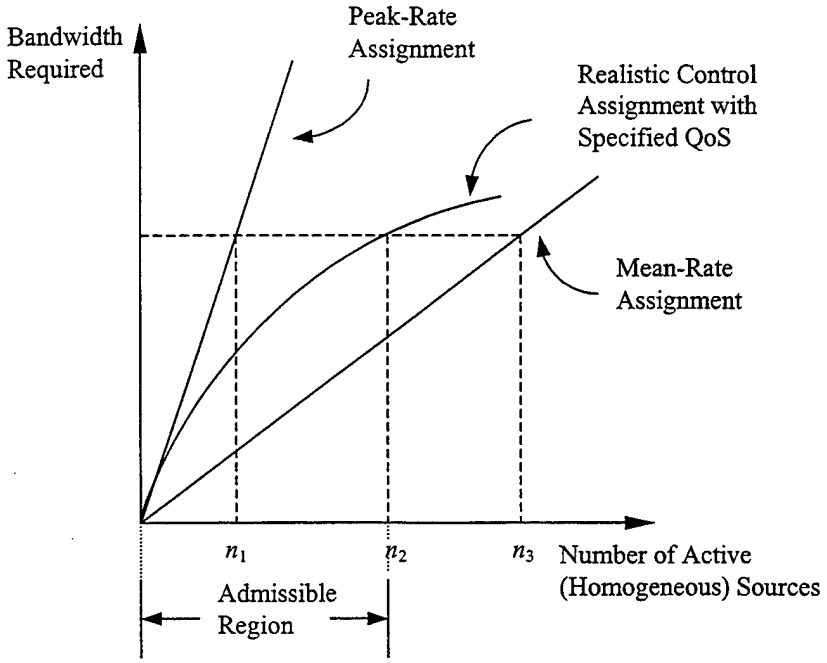


Figure II.5: Admission Control for Homogeneous ON-OFF Sources [80]

b. Access Control

Access control is used to ensure that users do not violate their traffic descriptors negotiated during connection admission. Given a decision to admit a new call into the network, the traffic generated by this call must be monitored to ensure that it does not start congesting the network [80]. Due to the random nature of the practical sources, congestion may develop despite a good admission policy. Intentional misuse by some users is another issue of concern. To prevent congestion from occurring, control must be employed at the AP (or the UNI) as well as within the network.

c. Traffic Shaping

Traffic shaping is a mechanism that modifies the traffic characteristics of a stream of cells on a connection to achieve better network efficiency while still meeting the QoS objectives [25]. The technique may significantly increase capacity savings for highly-bursty sources while increasing the transfer delay [64]. Examples of traffic shaping are selective cell discarding, peak-cell-rate reduction, and cell spacing (in time) [25].

B. DLC AND MAC IN MOBILE INTEGRATED SERVICES NETWORKS

The DL layer provides reliable transfer of information across the physical link, sends blocks of data (*frames*) with necessary synchronization, and performs error control and flow control [86]. In the case of a multiple-access medium (e.g., a radio channel), the DL layer also performs the task of sharing of the medium among the users, in some manner. These functions are performed on a point-to-point or point-to-multipoint (single-hop) basis only, not end to end.

In this section, the functions performed by the DL layer are discussed in detail. Also, the concepts of mobile networks in general and tactical and packet-radio networks using ATM in particular are briefly described. Because of its importance in this work, emphasis is placed on the MAC sublayer. Physical-layer aspects of the mobile network are discussed as the physical layer directly influences the performance of the MAC.

1. Access-Control Functionality

The MAC sublayer manages allocation of the multiaccess medium among the various nodes. It has been added to the open-system-interconnection (OSI) layered model [39] that has been basically designed for wireline computer communication networks. Conceptually, we can view a wireless, multiaccess, communication system in queueing terms. Each node in the network has a queue of packets to be transmitted, and the multiaccess channel is a common server. Ideally, the server should view all the waiting packets as one combined queue to be served according to an appropriate queueing

discipline. Unfortunately, the server does not know which nodes contain packets; similarly, nodes are unaware of packets at other nodes, thus creating a "distributed-" queue scenario. The problem, which is not trivial due to the geographically distributed location of the users, is one of conflict resolution in accessing the channel among them.

Multiple-access protocols for conflict resolution that have been proposed and studied can be generally grouped into three general categories: polling, contention, and reservation [52]. Under *polling* protocols, users are passive, i.e., they remain quiet whether or not they desire access of the channel [86]. They are queried from time to time by a central controller; a user can transmit data only when queried. On the other hand, both contention and reservation protocols require users having data for transmission ("ready" users) to actively seek channel access. A sample list of polling protocols proposed in the literature are the exhausted and gated policies [57], random access polling [17], and protocols for non-central networks [22]. In *contention* protocols, no attempt is made to coordinate the ready users to avoid collisions entirely [12]. Instead, each user monitors the shared channel and tries to transmit the data packets without incurring a conflict as much as possible. Collided packets are retransmitted by users based on a control algorithm using local information as well as observable outcomes in the channel. Representative contention algorithms are the pure and slotted ALOHA [12], carrier-sense multiple-access family [86], and the IEEE 802.11 for wireless LANs. The objective of *reservation* protocols is to avoid collisions of data completely. To do so, a queue global to all nodes needs to be maintained for channel access. Each user, having data to send, generates a request to reserve a place in the global queue [10]. A fraction of the channel capacity is used to accommodate the reservation-request traffic. Since users are geographically distributed, the multiple-access problem has not disappeared. It now exists in the access of the reservation channel. Examples of reservation protocols are the slotted ALOHA with reservation [52], packet reservation multiple access (PRMA) [32], and non-collision PRMA [96].

Other classes of multiple-access protocols, such as adaptive or mixed modes, are designed to handle specific combinations of traffic; however, they do not significantly improve the overall throughput-delay performance of the network [50].

2. Error-Control Functionality

Error control refers to mechanisms to detect and correct errors that occur in the transmission of frames. Data frames are sent in a sequence and arrive in the same order in which they were sent. Each transmitted frame is subject to a variable amount of delay before reception and errors due to impairments in the channel. Two types of errors are possible: a loss of the complete frame due to a noise burst or a failure of the receiver to recognize the beginning of the frame (preamble sequence), and a damaged frame in which the frame is recognizable but contains errors. The most common techniques for error control are based on a hybrid of automatic repeat request (ARQ), FECC, and error detection.

3. Mobile ATM Networks

Since adequate radio spectrum is now available in the mobile cellular and personal communication bands, researchers and service providers have explored the feasibility of extending ATM-like virtual connectivity from the wireline to the wireless domain. The ATM Forum has formed a Wireless ATM Working Group to study these issues and develop standards for wireless ATM (WATM) [89] [77]. The charter of this group is to develop a set of specifications in order to facilitate the use of ATM technology for a broad range of wireless network scenarios, both public and private.

In addition to the control- and user-plane functions required for ATM, MATM requires mobility-related functions in order to support mobile connection establishment, connection handovers, etc. [89]. Mobile connection management protocol (MCMP) handles the initialization of a mobile call, specification of the desired QoS, and the support of this QoS by the network throughout the call lifetime. Supervision of allocation and de-allocation of virtual channel identifiers (VCIs) over the wireline and wireless portions of the connection are also performed by the MCMP. Mobile handover

management protocol supports remote handovers in cellular-type networks. It reroutes all active connections of the remote to the new access point, possibly reallocating VCs after a handover, and ensures sequential cell arrivals. Mobile location management protocol maintains the physical location of the remote in relation to its current wireless access point and associated network. These are done via registrations of remotes on power-up and handoffs later. Mobile routing protocol supports (dynamic) routing in both the wireline and the wireless segments of the network to accommodate the dynamic nature of the network topology. Mobile medium-access-control protocol advocates multiple-access operation in the radio channel and is usually controlled by a central controller (known as the *base station* in cellular systems) that has knowledge about wireless resource utilization. Mobile data-link-control protocol performs error control and flow control. The mobile MAC and DLC protocols (together) must be able to support multiple VCs having different QoS requirements.

Designing high-speed, wireless, network architecture requires a careful consideration of many communication, control, and management aspects. Issues that must be taken into account include the types of service to be supported, mobility profiles of users, availability and limitations of wireless technologies, high network spatial efficiency (in bps/Hz/km^2), low remote-station cost, complexity, and power consumption [89] [76].

4. Tactical Mobile ATM Networks

Wireline and mobile ATM technologies can also benefit tactical networks. Simplification of network management due to integration of separate networks for voice and data into one is especially attractive. In order to satisfy military needs, however, additional constraints must be imposed and some new features added to commercial networks. First, lower transmission rates are available at tactical levels, mainly due to stricter requirements about the operational environment of the equipment [35]. For example, a transceiver called near-term digital radio, which is designed to operate at data rates of hundreds of kbps, is currently being developed by the US Army for Force XXI, to provide seamless communication for the digital battlefield in the 21st Century (platoon

to brigade tactical echelons) [7]. Low transmission rates mandate low-bit-rate applications that require special attention to guarantee specific end-to-end delays (for example, overcome large segmentation delays by reducing cell size or multiplexing several sources into a single cell) [92]. Second, several levels of user priorities and precedence are needed in order to be able to provide non-fair resource allocation according to the military hierarchy [35]. Third, the military networks usually operate in strong radio-frequency-interference environment, especially at the headquarters echelons. Consequently, the channel links are expected to be noisier compared to their counterparts in a civilian environment. For efficient transport of data and signaling frames over noisy radio links, ways to enhance the bit-error-rate performance of the associated links need to be explored. These may include compression of the frame header, strengthening the frame synchronization, use of smaller frames (mini frames) at the transmission level, or addition of FECC [92]. Security, interconnection with existing systems, and survivability are additional issues required to be addressed as well [73].

5. Physical Layer

The design of a DL layer is highly dependent upon the characteristics of the physical medium in use. Here we describe the physical-layer aspects of mobile communications with an introduction to *packet-radio* networks that are typical in a military environment.

Packet radio is a technology that extends the original packet-switching concepts, which evolved for networks of point-to-point communication links, to the domain of broadcast radio networks [45]. The original purpose for packet-radio development was based on tactical, military, computer communication requirements, though we are currently witnessing an increased use of this technology in commercial applications (e.g., wireless LAN). Packet-radio technology is applicable to ground-based, airborne, seaborne, and space environments.

The radio links, particularly when mobile remote stations are involved, are subject to severe variations in the received signal strength due to the terrain, man-made structures, and foliage [45]. Even under desirable conditions, where a line-of-sight (LOS)

radio path between end users exists, reflections, diffraction, and multipath can greatly reduce the signal strength [4]. These phenomena give rise to multiple signal paths leading to distortion and fading. Radio-frequency (RF) connectivity is thus difficult to predict and may powerfully change in unexpected ways as remotes move about. The existence of multipath signal components affects the reliability with which symbol decisions can be made in the receiver by causing symbol distortion. Typical multipath signals are experienced as intersymbol interference that occurs when a symbol is overlapped by delayed components of adjacent symbols. Adaptive equalization can improve the performance by suppressing the multipath components, but it must rapidly obtain good estimates of the channel impulse response [45]. An estimator-correlator type of receiver, such as typified by the RAKE structure, may exhibit a possibility for work in the range of megabit rates.

A distinction is made between sited and non-sited remotes. A *sited* remote is one that has been located to avoid surrounding obstacles, and its antenna has been elevated to the maximum extent possible. A remote operating from a moving vehicle would generally be considered *non-sited.* The mixture of users of the two types further complicates the prediction of RF connectivity in a large, mobile, user community. When the communicating users are in motion, fading is observed as a function of time proportional to the velocity of the user motion.

Packet-radio nodes share a single wideband channel; each node consists of a radio, an antenna, and a digital controller [55]. Linear modulation techniques, such as QPSK, DPSK, and QAM are possible candidates for the radio, but due to the high level of noise and the difficulty of equalization, multilevel modulation is difficult to achieve [10]. The antennas used by the stations are usually omnidirectional. Good connectivity can be maintained between mobile remotes as long as a LOS path exists between the antennas. The digital controller is in charge of information transfer between the end users over the wireless medium. Error control and multiple access control (DLC and MAC functions, respectively) are part of the utilities performed by the digital controller. The controller uses packets for information transfer; a typical structure of a transmitted packet

is illustrated in Figure II.6. A packet preamble, which usually contains a few dozens of bits, is used by the radio section of the receiver for several purposes. The first few bits allow carrier detection and automatic-gain-control (AGC) setup. The rest of the bits are used to acquire bit and byte timing (synchronization). The user data follows the preamble. A checksum field may be provided in the packet (or may be left for the application to handle). The preamble sequence must be sent before any data transmission, which may consist of a payload that includes more than one data unit.

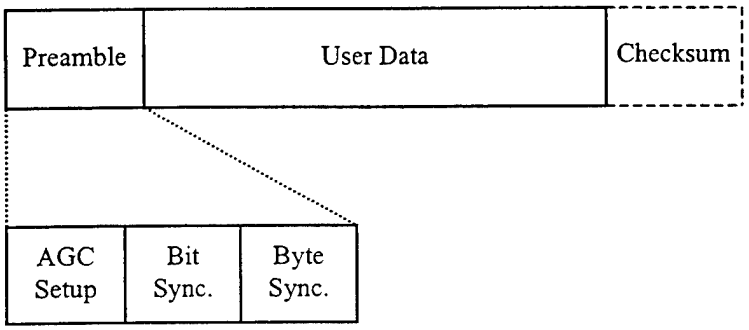


Figure II.6: Structure of a Typical Packet-Radio Frame [45]

6. Current Research on Data-Link Layer for Mobile ATM Networks

Three generations of wireless cellular networks are recognized in the literature. First-generation, analog, voice networks have spread worldwide. Second-generation, digital, voice/data networks are currently under deployment. Proposed third-generation networks, designed to carry integrated-services traffic (voice, video, image, text, or a combination of them), are under study by the Commission of the European Union as well as the International Telecommunications Union (ITU-R) [79].

Here we briefly report the past as well as present effort on various issues related to data-link layer in mobile integrated services networks that are relevant to this work. This includes network configuration, MAC protocols, and system architecture. Much of the literature, however, concerns with cellular-type networks in which a base station manages the operation of the cell.

a. Network Configuration

The Olivetti Radio ATM architecture, which extends existing wireline ATM networks to the radio medium, is based on a *remote-station representative* [89]. A software agent called mobile representative (MR) is located at the cell's interfacing switches with the wireless segments, and supports the mobility of the remotes. Each MR has a set of remotes associates with it, as has been defined by the network manager; the MR is referred to as the 'home' of these remotes. A user wishing to set a connection with a remote, is required to contact the remote's MR (since the remote may have roamed to another cell). Hence, the MR must always maintain up-to-date location information regarding its home remotes. Information transfer involving a remote passes through the interfacing switch at the current location of the remote and the remote's MR, making it a non-optimal route. During remote handovers, the segment from one end user to the MR remains fixed while the segment from the MR to the remote may be rerouted.

The mobile broadband system, sponsored by the Commission of the European Union, contains *three hierarchical levels* of mobile elements and aims to provide ATM compatibility for wireless services [89]. Remotes in a cellular system get network service from a fixed broadband termination unit (FBTU) that serves as a base station. Several FBTUs are connected to one fixed broadband termination control unit (FBTCU) that controls the connection management and channel allocation of calls within its area. Two or more FBTCUs are then connected to a mobile switching unit, which connects the wireless segment to the fixed one. The mobile system is seen by the stationary network as a standard ATM UNI.

An architecture for (cellular-type) WATM networks, based on *distributed control*, has been proposed by AT&T [93]. Each ATM switch and base station has a channel server (CSR), which supports connection and handover management. Connectivity between ATM switches, base stations, and CSRs in a given area is achieved by cluster-based wireless-LAN communications. The CSRs within the cluster are connected in a star topology to a connection server. During connection setup, the CSR at the source sends a request to the connection server to reserve resources along the source-

destination path. The latter spreads this request to all the CSRs along the path, thus admission decision is obtained faster than in the hop-by-hop traditional case. Handovers within a cluster are managed by the connection server of the cluster and the corresponding CSRs; otherwise, interaction between connection servers of the old and new clusters is required to support cell rerouting along the new source-destination path.

A *hierarchical, ATM-based, transport* architecture for the next-generation, multiservices, wireless personal communication network (PCN) has been proposed by NEC USA [76]. Each node in the PCN network includes a hierarchy (from bottom up) of base stations, multiplexers, small ATM switches, and large ATM switches. Connection, handover, and location controls in the PCN are performed by a mobile service unit located within the large ATM switches. The system is based on cellular microcells, each of which is serviced by high-speed, shared-access, radio links based on ATM-compatible cell-relay principles. The ATM cell serves as the basic unit for protocol processing and switching in both wireline and wireless portions of the network. The wireless segment of the network requires additional medium-access-control and data-link-control layers for channel sharing and error control in the radio links.

b. MAC Protocols

A multiservice, dynamic-reservation, time-division-multiple-access (TDMA) channel in which a protocol frame is divided into request slots and message slots is proposed in [76]. Each frame slot provides for transmission of an ATM-like cell with data payload of 48 octets, together with a PCN protocol header. Request slots are comparatively short and used for initial access in slotted-ALOHA (contention) mode. Of the frame slots, some are assigned to CBR voice traffic. The rest of the slots are dynamically allocated (based on a suitable statistical multiplexing scheme, given the UPC values declared during call establishment) to VBR and ABR cells. Earliest-deadline-first service approach is expected to improve cell loss rate of real-time connections in mixed, RT and NRT, traffic scenarios.

The access of the physical medium by multiple sources should be controlled in such a way that not only the delay requirements of a specific source are satisfied, but also the synchronization between multiple sources composing a node is met [6]. The MAC proposed in [76] does not take such requirements into account. In [6], a TDMA-based multiple-access scheme is used in which transmissions from remotes to the base station occur on the uplink frequency band while the base station broadcasts every transmitted packet on the downlink frequency band. The uplink frame is divided into uplink slots, each allowing the transmission of one ATM cell followed by a guard period; the length of the latter is determined by the coverage area of the base station. The source or destination remotes are identified by the virtual-path-identifier (VPI) field value in the ATM cell header. In the uplink frequency, the possible messages to occur include:

- Successful transmission in a non-contending mode.
- Registration message in a contention mode to declare the presence of a remote, followed by VPI assignment.
- Request for resource allocation, issued by registered remotes (in the cell controlled by the base station) without (currently) established connections that would like to access the control channel in a contention mode.
- Request-back message indicating transition of a source between inactive and active states.

In order to determine how to reallocate the uplink time slots, the base station executes an "allocation algorithm," which assigns different priorities to different types of services and uses a scheduling method, such as round robin. The scheduling algorithm divides the set of established connections into time-slot owners and renters. The time-slot owners acquire their time slots as soon as possible after having reported transition from inactive to active. The time-slot renters are just taken into account by the algorithm when an owned time slot is available (owner is inactive). The scheme presents some delays to the sources, but these are always less than the uplink frame duration such that bandwidth is preserved.

Polling systems offer the ability to allocate "bandwidth on demand" to users, suitable for ATM networks. In a regular polling scheme, a user is polled every

cycle even if no cells are present in its buffer most of the time, i.e., part of the bandwidth is wasted. A non-uniform, polling-based, MAC protocol, proposed in [61] and [62] for indoor mobile ATM networks, polls each remote as frequently as its traffic descriptors and requirements necessitate. For each type of traffic (both RT and NRT are considered) a certain cycle limit is determined. Low-bit-rate sources with less-stringent delay requirements have larger polling cycles, i.e., they are polled less frequently. This way, the polling overhead is reduced, especially for systems containing many low-bit-rate and a few high-bit-rate sources. The protocol assumes that an ARQ mechanism is present at the ATM cell level to overcome failures in the channel for all kinds of traffic.

In order to provide satisfactory transmission performance for delay-sensitive and error-sensitive sources, a protocol that combines an extended packet-reservation multiple-access protocol with reservation-ALOHA is proposed in [102]. For delay-sensitive voice sources, the protocol works in a way similar to PRMA proposed by [32] except that the slots are over *multiple* carriers instead of a single carrier. Once a remote reserves a slot, it continues to use the same slot over each frame on the same carrier until the transmission is completed. For high-rate sources that need multiple slots over one frame period, the reservation process contains the following stages:

- A remote first listens to the broadcasting channel to see if there are enough slots available. If not, it sends one packet that tells the base station the number of slots it needs over one frame period.

- The base station then checks for all available slots over all carriers. If enough slots are available, the base station reserves them for the remote by broadcasting the information through the downlink. If not, the base station sends a negative acknowledgment (NACK) to the remote.

- If the remote receives a positive acknowledgement (ACK), it starts transmission using the slots reserved for it. Otherwise, it listens to the broadcasting channel until there are enough slots available, and the process repeats itself.

To best utilize the slots available in each frame, all the slots on different carriers must be polled together for reservation [102]. For error-sensitive sources with various data rates, an extended reservation-ALOHA protocol over multiple carriers with ARQ error control is proposed. One carrier is used for reservation and the rest for packet

transmissions. The reservation channel is divided into a number of small reservation subslots used to reserve packet slots (after appropriate ACK). ARQ is used to overcome the harsh radio-transmission environment. The type of ARQ chosen depends on radio channel fading statistics, modulation scheme, propagation delay, transmission rate, and packet size. Analysis of the protocol indicates that the throughput increases only when the number of remotes is small compared to the number of slots. Then, as more collisions occur, the throughput decreases, and the packet loss probability increases dramatically due to finite buffers.

Time-slotted frame-based protocols using multiple transmissions per slot, or code division multiple access (CDMA), are presented by [21] and [72]. In the former, the users are divided into three classes of traffic, namely, CBR, VBR, and ABR. CBR and VBR sources cannot be queued and must have bandwidth reserved while the non-delay-sensitive ABR traffic has the lowest priority. CBR and VBR traffic are divided into voice and video groups only, each with Poisson arrival characteristics. Both schemes assume that synchronization in the network is ensured by the base station.

A modified, polling-based, MAC protocol is presented in [2]. The approach requires the MAC protocol to provide time-ordered ATM cells between the base station and each remote, focusing on one representative cell. A token, including a mobile addressee identification, is sent to each remote according to some order, ensuring the holder the right to send a limited number of ATM cells. The specified remote responds by sending data it has ready for transmission or by transmitting a short pilot tone to let the base station know of its current existence within the cell. Once all the remotes have been polled, the base station can then sequentially transmit its waiting cells. The overall protocol frame is, thus, dynamically changing, depending on the backlogged data of the network users. The efficiency of the protocol decreases significantly whenever the frame length decreases to values less than several tens of milliseconds; hence, it is suitable only for slow-moving remotes (e.g., pedestrians).

An alternative protocol suggested by [2], suitable for vehicular velocities, segments the channel bandwidth into three intervals (see Figure II.7, where n represents

the number of remotes). The first is a polling interval, with one polling slot for each remote within the cell, which is divided into base-send and remote-send subsegments. The second is used primarily for remote-to-base communications (and, under certain circumstances, for base-to-remote communications as well). The third is used only for base-to-remote communications. A remote, having queued ATM cells, sends a request message in the remote-to-base subsegment of its assigned polling slot. The base station acknowledges this segment during the following base-to-remote subsegment of that polling slot. Corresponding to each polling slot is a larger time slot in the remote-to-base interval. Reception of the remote's request at the base station reserves the corresponding time slot in the remote-to-base interval for use by that remote. At the beginning of the reserved time slot, the remote sends a brief RF tone, followed by a sequence of ATM cells. If the remote has not sent a request, the corresponding time slot in the remote-to-base interval may be assigned by the base station for other purposes. To send information to a given remote, the base station must first stimulate the remote to reply with a pilot tone. Overall utilization of the channel is not driven by the need to poll each remote within some prescribed timeout interval as in the previously proposed protocol; thus, the utilization can be high. A disadvantage of this scheme is the need to compute the antenna element weights in real time, whenever the pilot tone is received.

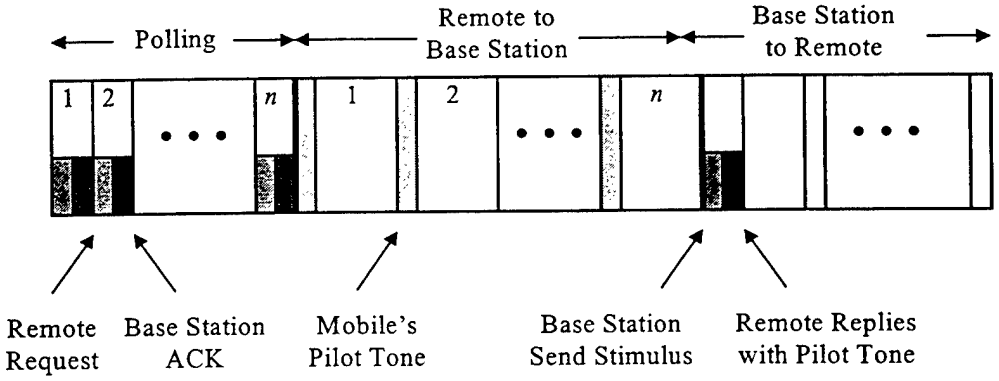


Figure II.7: Media Access Protocol Permitting Rapid Array Adaptation [2]

c. System Architecture

A study dealing with the overall functionality of the DL layer in a mobile ATM environment is presented in [100]. The architecture, shown in Figure II.8, includes DLC and MAC protocol sublayers. Standard ATM cells arrive at the DLC sublayer corresponding to a particular VC. The DLC first replaces the standard ATM header with a WATM header and requires the supervisory-MAC (S-MAC) to allocate slots for the transmission of these cells. The S-MAC schedules transmissions/receptions for a particular frame and passes this schedule table to the core-MAC (C-MAC), which receives the cells from the corresponding DLC for transmission. At the receiving end, the DLC transfers received cells (after error recovery, if required) to the ATM layer after replacing the WATM header with the standard ATM header.

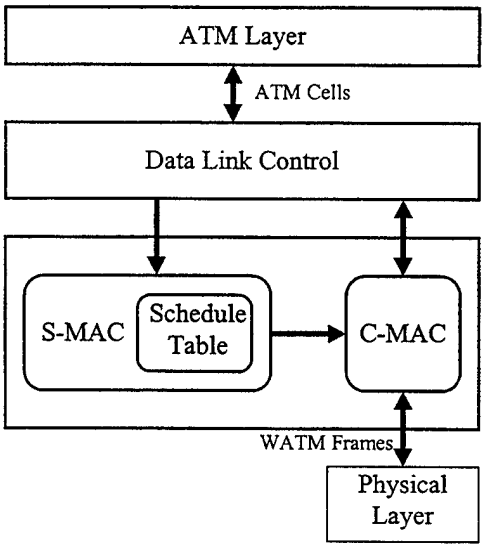


Figure II.8: Wireless ATM System Architecture [100]

The S-MAC at the base station is responsible for channel allocation, both uplink and downlink, for all VCs in the system. Subject to the fulfillment of the QoS requirements of each VC, the S-MAC arrives at a policy of scheduling data transmissions and acknowledgments to enable error recovery by the DLC. Call-admission-control

functions for the wireless link are also part of the S-MAC at the base station. At a remote, the S-MAC processes the control information received in each frame from the base station and builds the schedule table for its C-MAC. The C-MAC, based on the schedule table entries, multiplexes/demultiplexes transmissions and receptions for the corresponding VCs. A typical entry contains service type, message type, VCI, position, and duration of the service. Frame structure of the WATM contains TDMA-based downlink transmissions and dynamic TDMA transmissions for the uplink, with varied boundaries between them to support changing traffic needs. Control message slots are also reserved in the frame for contention by remotes, in a slotted-ALOHA fashion.

DLC functions vary in relation to the traffic type [100]. For CBR traffic, the DLC attempts to correct cell errors within a specified time window, indicated at VC setup. Using additional on-demand bandwidth for retransmissions of lost cells maintains a nominal CBR stream's bandwidth seen at the ATM interface. A FCFS buffer is used at the DLC to ensure that the cell delay variation is maintained within acceptable limits and provides a window for error recovery while introducing a fixed delay over the wireless link. ABR services can tolerate longer delays but require extremely-low cell loss rate; thus, the strategy of the DLC layer, in this case, is to set the time limit for error recovery to be as large as possible.

C. SYSTEM ENVIRONMENT

In this section, the (operational) environment of the system under study is presented in detail.

1. Operational Environment

The network under study is a (single macrocell) central mobile integrated services digital network (ISDN). Its topology is shown in Figure II.9. It includes a command post (CP) and several forces under command; together we call them a *unit*. The CP, when stationary, usually opens a radio-telephone LOS link or a satellite link along its commanding echelon (the wireline backbone). In a battle environment, the basic structure of units can change due to operational reasons or destruction. As a result, a unit may get

control of other forces (temporarily or for an extended period) or can attach some of its forces to other units for fighting efforts in other battle zones. Nevertheless, it is assumed that the central configuration remains throughout the operation and that there is a CP in the network to manage the unit operationally.

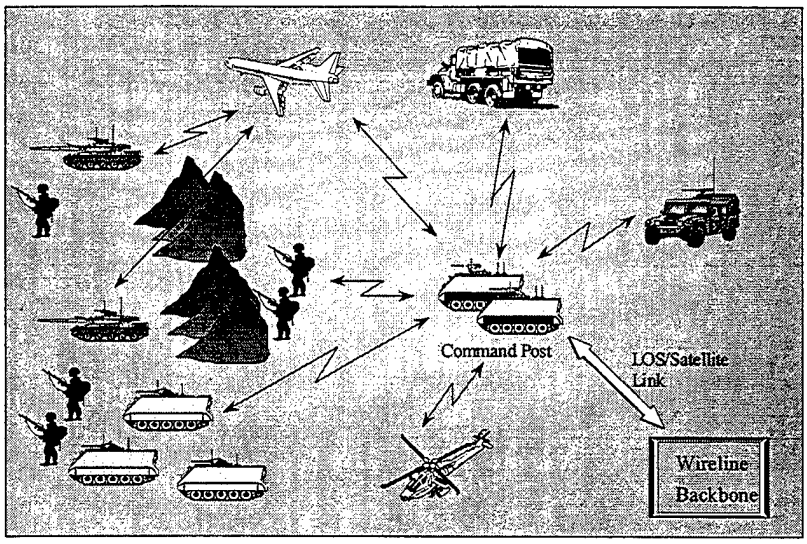


Figure II.9: Typical Tactical System Topology

All forces are mobile by definition, thus communication in the network takes place via mobile radio transceivers. It is assumed that there is connectivity between the CP and its forces under command at all times (or most of the time). Relays (installed in an aircraft, a helicopter, or some other means) may be employed to extend the communication range. Connectivity between the forces other than via the CP varies as the operation progresses. It depends on many factors, such as distance between stations, type of terrain, transmitters power, and elevation of antennas. Transparent roaming of users between radio cells is not allowed as in a cellular system. The proposed system occupies, in comparison with a cellular system, one macrocell (coverage area of about 25-50 squared miles), which reflects a fighting cell in the battlefield. The radio system under study is of type packet radio, operating in the UHF band (typically in the range of 900 to 1900 MHz). Each node is equipped with a mobile transceiver and an omnidirectional antenna.

Our concern here is a mobile integrated services digital network, employed by a given unit for command, control, communication, and intelligence purposes. This network is expected to serve many sources of information, such as slow motion video, images, database files, electronic mail, location reports, and (point-to-point) voice conversations. The backbone channel from the CP, if installed, provides a link to a gateway that connects the unit to the external world. This enables the unit to exchange traffic with other units in general or with its commanding echelon in particular.

2. Technology

Much effort has been devoted to achieve faster speeds of communication over the wireline links, to allow higher-bit-rate applications. Rates of gigabits per second already exist with potential for higher rates in the upcoming years. Nevertheless, for outdoor (macrocell) mobile networks, where the LOS propagation delay is of the order of a few microseconds [45], multipath and other impairments place a limit on the (single-band) channel bit rate (typically 1.0 to 1.5 Mbps).

Emerging, wireless, integrated-services technologies (such as mobile ATM) that make use of a macrocell environment can only support low-bit-rate sources because of the limited channel capacity available. In fact, the multiplexing provided by ATM is well suited to low-speed wireless links since it leads to lower delay jitters and queueing delays [89]. Constraints imposed on ATM cell processing time due to high-speed transmission in wireline ATM networks are absent for MATM networks. Moreover, applying the notion of VCs with specified QoS over wireless links allows the MAC layer to allocate and schedule shared wireless-channel resources more efficiently. (Nevertheless, one must keep in mind that this notion is applied in a qualitative sense due to the higher cell loss probability in the wireless case.)

3. Classes of Service

Taking the low-bit-rate radio channel into account, we propose definitions for three classes of service in the system [90]:

- Class A: Low-bit-rate (32 kbps), digital, voice conversations (involving bi-directional flow of traffic).

- Class B: Variable, low-bit-rate, compressed video. A representative, real-time, video source generates one frame per second of size 256×256 pixels and 256 gray levels per pixel. Picture frames (8×256×256) are compressed (8:1) to obtain a source with mean arrival rate[3] of 64 kbps.

- Class C: Bursty data. This includes queries and responses from textual and imagery databases, command assignments, position reports, and mail transfers. The size of each type of information source varies between 200 bits and 1 megabits after appropriate compression. Existence of an access-control mechanism is assumed to limit the peak rate of the data source to a specified rate (e.g., using a traffic shaper).

The definition of the classes of service must be supported by the QoS requirements [9] to be fulfilled by the network. The QoS issues are addressed later in Section IV.B.

D. SYSTEM ARCHITECTURE

Here we describe the architecture of the network and the network stations in more details. The packet-radio network under study comprises up to 64 nodes (including the CP). A single radio-carrier frequency is used for all transmissions, *downlink* and *uplink*, using the time-division-duplex (TDD) approach. The proposed architecture has two key modules: the CP and a mobile network coordinator (MNC). The CP is concerned with network management from operational aspects while the MNC is responsible for managing and maintaining the network communication entities. Each remote in the network has a bi-directional radio connection with the MNC, although connections with other remotes are possible as well, thereby realizing a (logical) star topology for the network. Each station is equipped with a single radio transceiver that serves *all* the users and services of that station. In this section, we propose system architectures for the remote, the CP and the MNC; two functional models are presented for configuring the CP and the MNC.

---

[3] The picture itself may be generated at a higher resolution and/or rate; however, we assume that it is digitized and sent at the indicated rate.

1. Mobile-Station Architecture

Generally, in a mobile station, traffic is generated by several user applications (speech, video, and data) within the station as illustrated in Figure II.10. Each station may have its own LAN such that one or more sources can communicate simultaneously [91]. All user sources/applications are interfaced to single radio equipment via at least three modules. The first is a *LAN-interface* module that handles the transmission and reception of information across the LAN. An application (e.g., transfer control protocol/Internet protocol) may be present to capture the information from the LAN and forward it toward the radio channel and vice versa. For example, an ATM LAN would not require an application module since the information on the LAN is already formatted as standard ATM cells while an Ethernet LAN would require it. The second module contains the AAL and ATM layers, which convert user information into ATM cells and vice versa. The mobile-communication-controller (MCC) module is required to handle the transmission and reception of the ATM cells in the wireless network. Essentially, it contains the functionality of the DLC, the MAC, and related protocols that are required for the wireless operation. It also handles management and control functions necessary to support flows of user traffic to and from the local sources at the station. In summary, the information originated by the various users on the LAN is translated into ATM cells and enqueued by the MCC for transmission over the wireless channel.

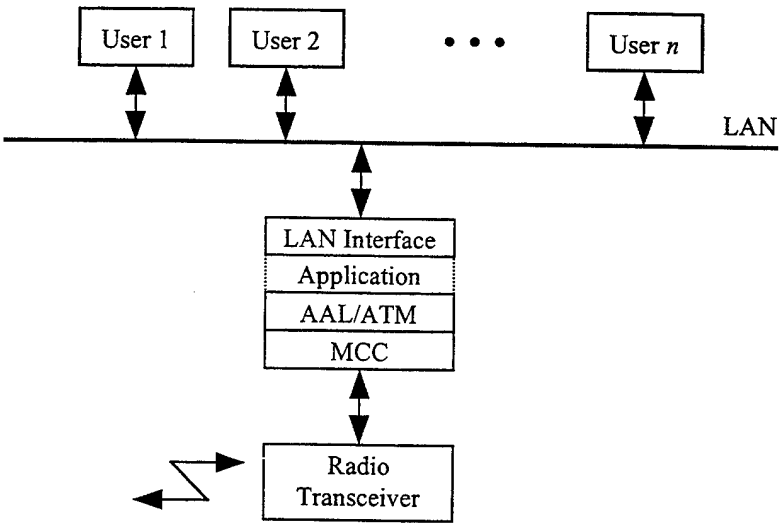
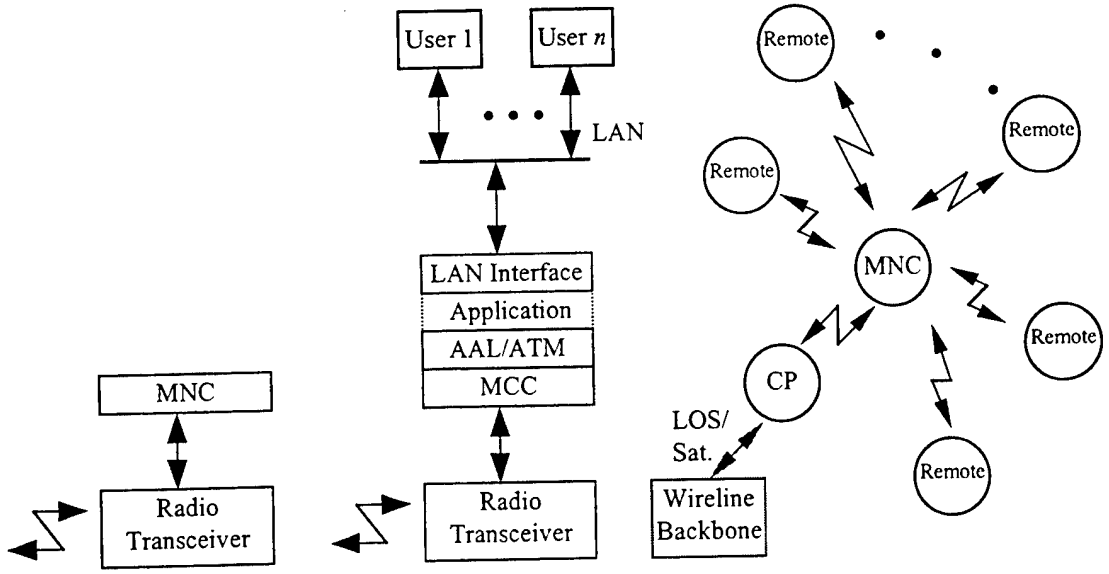


Figure II.10: General Architecture of a Mobile Station

2. CP Architecture

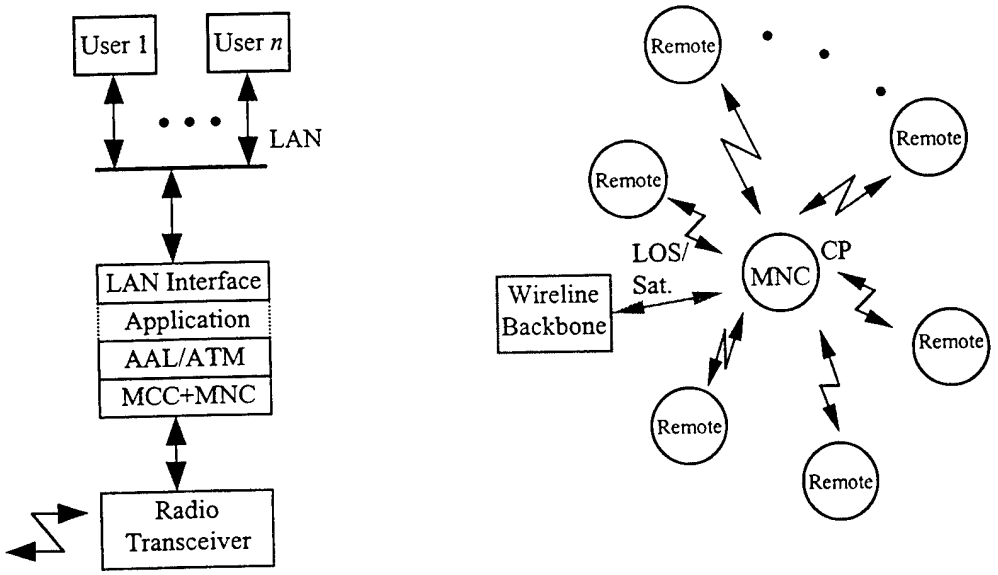
In cellular systems, the base station does not generate any user information other than control/signaling messages. It basically coordinates the remote transmissions within the cell (the MNC function in our case). Unlike the cellular base station, the CP contains user services that *do* generate traffic.

Here, two schemes are considered as possible architecture candidates for the CP (and the MNC). The first scheme, shown in Figure II.11, separates the function of network coordination from that of handling user traffic at the CP. The CP then functions like a regular remote (albeit with heavier traffic load) in the network running the MCC, while an independent node is set up for the coordination functionality. (A radio connection between the MNC node and all other nodes in the network is required.) The second scheme, illustrated in Figure II.12, combines the MNC and MCC functions at the CP. The combined module performs all MCC functions mentioned above and additionally coordinates the traffic flow in the network, using a single radio transceiver. Notice that in both cases the link to the wireline ATM backbone is located at the CP.



(a) MNC Architecture **(b) CP Architecture** **(c) Logical Topology**

Figure II.11: CP Architecture with Separate MCC and MNC



(a) CP Architecture **(b) Logical Topology**

Figure II.12: CP Architecture with Combined MCC and MNC

The two schemes have distinct advantages and disadvantages. In the first approach (Figure II.11), an additional transceiver and a computational element (for the MNC) are required. On the other hand, this scheme allows the network to be much more flexible. The MNC may be physically located within the CP area or at any other location, as the operation requires. This implies a greater survivability: the CP is no longer a single point of failure for both wireless operation and backbone connection. Such configuration can operate even if the backbone link does not exist or the CP is unavailable for any reason. A backup MNC station may be launched at any time in case of malfunction or destruction in battle.

The main weakness of the separate CP and MNC approach arises from the requirement that a link to the wireline backbone be at the CP. Signaling and information traffic from all sources that involve connections with outside world must pass through the CP. Since the MNC coordinates the traffic in the network, first the traffic has to flow on the uplink channel to the MNC and then be relayed on the downlink channel to the CP. The radio-network efficiency and throughput are thus significantly reduced. In addition, the scheme results in a complex network control. For example, consider the case of a call-admission decision of a new call between a remote user and an external user. In the combined approach (Figure II.12), the admission decision is determined at a single point (the CP); in the separate approach, it is done at *both* the MNC (the only element that has a complete knowledge of all the existing virtual channels in the mobile network) and the CP (that links the stationary backbone).

Possible solution for the external connection overhead in the separate CP and MNC approach is to provide a (fast) point-to-point link between the CP and the MNC; it is, however, undesirable for several reasons. First, it requires additional equipment, installation, maintenance, and user training. Second, the independent MNC becomes not as mobile as before. Third, it requires more professional manpower, usually located at the CP (as part of a larger technical crew), at the MNC. Fourth, it is desirable to locate the network manager at the CP, mainly for operational reasons (operational developments that affect technical decisions, e.g., registration of a new attached force in the network).

The additional point-to-point link increases the probability of failure and results in lower quality of the network service.

From the above, we can see that neither solution is perfect. The low bit rate of the radio channel requires an efficient architecture. On the other hand, in a military environment the survivability of the system is a key design issue. We choose to adopt the combined approach as the preferred scheme in the network because of its superior efficiency. We remark that the survivability and the option to operate the mobile network even without a CP are issues beyond the scope of this work and left as topics for future research.

3. Radio Channel

We assume that the radio channel has a capacity of 1 Mbps. Every transmission in the radio medium starts with a preamble sequence of length 40 bits (40 microseconds). This is required for bit timing and frame acquisition at the receiver, as well as for equalization purposes. A guard time of 6 bits (6 microseconds) ends every transmission to allow synchronization of receivers between transmissions, which is required due to the time-distance delay, clock instability, delay spread, and transient responses [30].

Synchronization in the network is obtained in a primary-secondary fashion, where the CP serves as the primary and each remote as a secondary. A periodic transmission is sent by the CP for this purpose.

4. Station Identifiers

Each operational unit in the military is assigned a 4-digit (decimal) unique identifier (sometimes together with its echelon name). We use these unique operational numbers to identify the remotes for control and signaling purposes (e.g., during registration of a remote in the network, at the beginning of a call-setup procedure, etc.). Sixteen bits are used for unique node identification.

E. PROTOCOL LAYERING

Several ATM-based transport architectures for the next-generation, multiservices, wireless PCN have been proposed in the literature [76] [89]. Here we propose to develop an *ATM-based, wireless, transport* mechanism to achieve the following objectives [76]: flexible bandwidth allocation, efficient multiplexing of traffic, and ease of interface with the wireline B-ISDN at the CP.

In this scheme, shown in Figure II.13, the basic 48-octet ATM cell from the wireline portion of the network is modified to form a MATM packet prior to its transmission over the radio channel. The cell payload is segmented into submultiples of 48 (e.g., 24 or 16 octets), and the header is adapted for wireless communication use (e.g., addition of error detection/correction code). The CP provides the interface between the wireline and wireless segments, thus it is in charge of translation between the packet formats. Since the backbone link is not error free, the ATM cells are wrapped by a data link, error-control header to ensure reliable communication over this link.

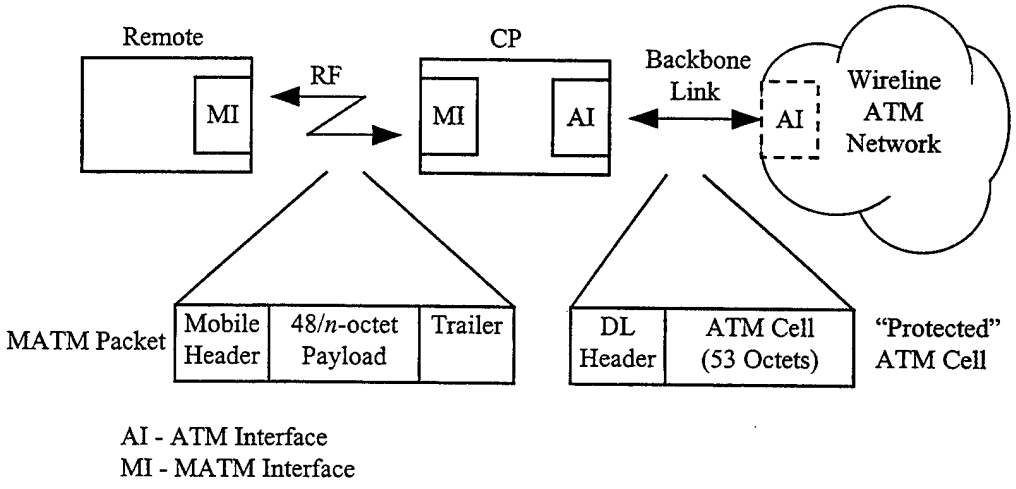
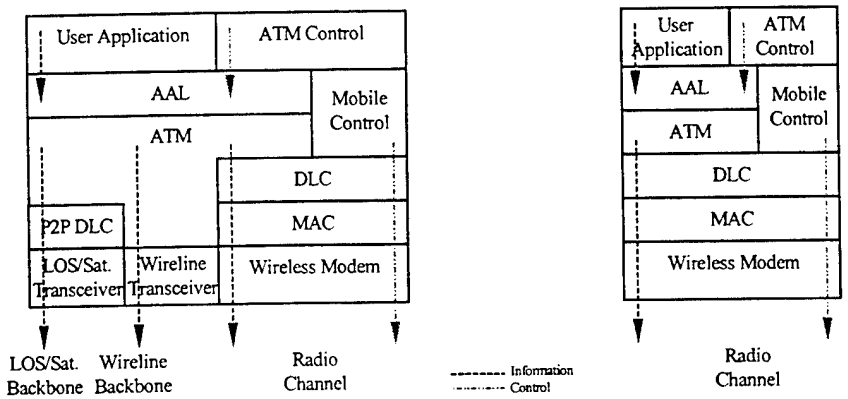


Figure II.13: Packet Formats in the Extended Integrates Services Network [76]

As shown in Figure II.14, the protocol layering of the wireless segment maintains the basic ATM layered architecture with additional layers to operate in a radio environment. These are the MAC and DLC layers, required mainly for channel allocation

and error-control, and the mobile-control layer, which adds some signaling functionality that is specific to the wireless medium. The ATM control module, which in the wireline case is in charge of signaling (using ATM cells) according to standards ITU-Q.2931 [41] or The ATM Forum-UNI 3.1 [8], is extended to support the mobile network as well. This extension includes capabilities, such as remote registration, request to access the channel for new call setup, information transmission, and call admission control over the radio channel.

In Figure II.14, the dashed arrows identify the possible flows of information and control/signaling data across the layers. At the CP, the information is exchanged with other nodes on the wireline segment of the network (via a backbone link) following the regular ATM-layering scheme with an additional data-link layer. This layer, called the point-to-point data-link-control (P2P DLC) layer, aims to protect the ATM cells against the backbone-link impairments. The CP may be stationary and connected directly to the wireline ATM backbone via a wireline link rather than over a LOS/satellite link. In this case, its physical carrier is different and the information does not flow through the P2P DLC. In the wireless segment (CP and remotes), the information passes through the DLC and the MAC layers as well. Control and signaling cells are generated by the ATM control and the mobile-control modules. Mobile-control messages bypass the ATM layers and are directly sent to the dedicated wireless layers.



(a) CP **(b) Remote**

Figure II.14: Protocol Layering in a Wireless ATM Network

F. PROPOSED ARCHITECTURE

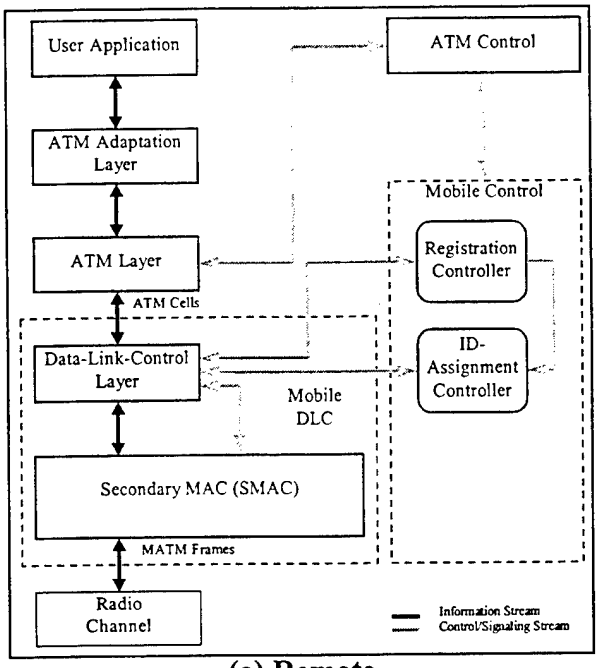
Here we describe a detailed architectural layout of the nodes in the radio network and the data and control flows among their components. Figures II.15a and II.15b illustrate the structures of a remote and the CP, respectively.

1. DLC and MAC

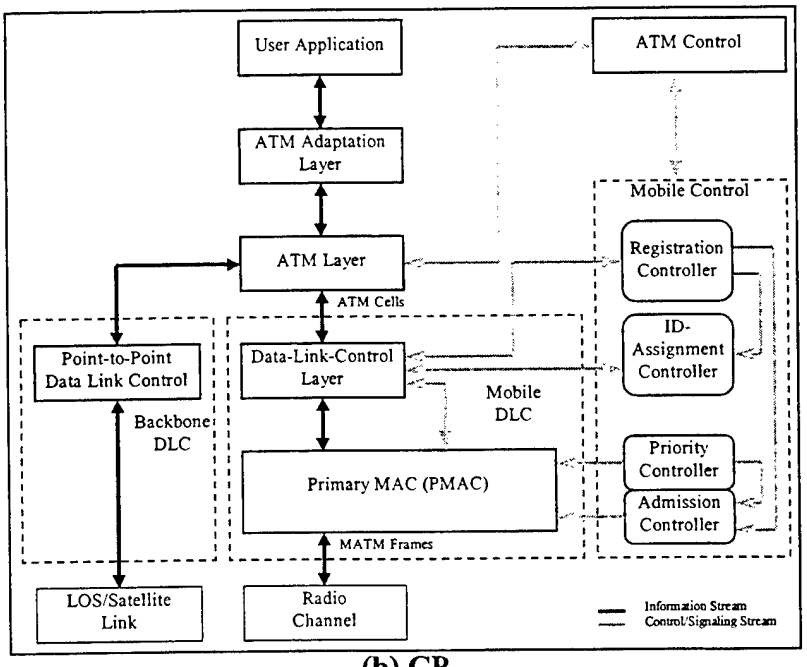
Standard ATM cells corresponding to a particular VC arrive at the DLC layer. The DLC layer segments the cells into packets, generates the error-control information, and replaces the standard ATM header with a MATM header. Then the packets are passed on to the primary MAC (PMAC) at the CP or secondary MAC (SMAC) in the remotes to schedule their transmission. Following a central allocation concept, the PMAC schedules all transmissions of cells within the network and informs the SMACs about their scheduled transmissions. A TDMA-based protocol is used, where time is divided into frames and frames into slots. At the scheduled transmission instant, the MAC translates the VCIs into a shorter-form mobile virtual channel identifiers (MVCIs) and passes the information to the physical layer. The receiving MAC, upon successful reception, passes the packets to the DLC. After error recovery, if required, the DLC replaces the MATM header of the received packets with the standard ATM header and reassembles the packets into ATM cells.

The PMAC at the CP is responsible for channel allocation for all mobile virtual channels (MVCs) in the system. Subject to the fulfillment of QoS requirements of each MVC, the PMAC arrives at a policy of scheduling transmissions of data and acknowledgments (to enable error recovery by the DLC). At the remote station, the SMAC processes the control information received in each MAC protocol frame from the PMAC and builds a schedule table that contains its allocated transmission instants on that frame. A (dynamic, TDMA-based) MAC frame contains a number of slots on downlink and uplink portions to support changing traffic loads. A reservation scheme that totally avoids collisions of cells in the channel is desired for this purpose. Using this scheme, the

SMAC informs the PMAC, by a short control message, that a given MVC has information to send. The PMAC then reserves a future slot for that source.



(a) Remote



(b) CP

Figure II.15: Proposed Mobile-Station Architectures

2. Registration Control

On power-up or when joining another unit's network, a remote follows a message exchange sequence with the CP in order to complete the registration procedure within that network. The remote requests the CP to join the network; a dedicated channel bandwidth is allocated for that purpose. The CP usually requires the remote to go through a process of authentication. After successfully completing the authentication, the remote can join the network. If accepted into the network, the remote is allocated a unique mobile signaling identifier (MSI), to be used as an identifier for all mobile signaling messages from that remote to the CP (i.e., it is common to all sources of the remote).

When a remote plans to leave the network (proper shut off), it so informs the CP. The CP then releases all active connections associated with that remote (if they have not been properly released yet) and de-allocates the MSI associated with it. In order to detect whether remotes have been improperly disconnected, each registered remote sends a *keep-alive* message indicating that it is still alive. In order to save battery power, the keep-alive messages are sent only if the station has no transmission at all over a pre-determined period. Larger timeouts can be assigned for registered but non-active remotes.

In summary, the registration controller performs three functions. First, it handles the registration (or reregistration) of remotes that join the network. Second, it manages the (proper) exit procedure of a remote from the network such that the allocated resources are released. Third, for cases in which a remote is improperly disconnected (e.g., due to power failure), it schedules periodic keep-alive message transmissions from the remotes to the CP to ensure their ongoing participation in the network.

3. Call Admission Control

In the case where the end-to-end path of a call has a wireless segment, the admission controller at the CP is in charge of the admission decision only for that portion of the path. (A call from a remote to a fixed station could be accepted in the mobile network but might be rejected somewhere along the backbone network.) Assuming three user-traffic classes, the admissible region forms a three-dimensional space as shown in

Figure II.16. Peak-rate and mean-rate allocation to each source determine the lower and upper bounds, respectively, of the allowed number of connections for each service class.

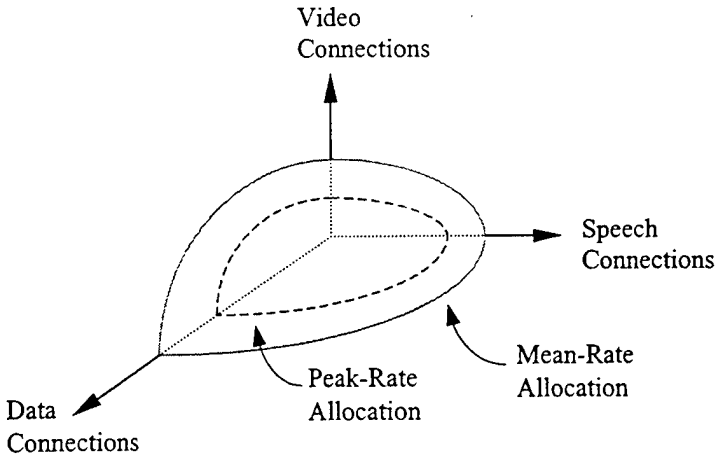


Figure II.16: Three-Dimensional Admissible Region

Call-admission-control function for the wireless link resides at the CP (see Figure II.15b). Admission-control requests from the wireless network are received by the ATM control unit and transferred to the (mobile) admission controller. The admission controller sends a request to the PMAC indicating the new call request parameters (QoS requirements and the originating station identifier) [3]. The PMAC uses this information to either accept or deny admission to the new MVC. If the call is accepted, the PMAC updates its database to schedule channel allocation to the new MVC according to its allocation policy. In case a remote has been improperly disconnected from the network, the registration controller informs the admission controller and the ID-assignment controller of this, and all the resources associated with this remote are then released.

4. Priority Control

Reflecting the importance of the task being performed or the hierarchical level of the user, there is a need for user priorities in the network [35]. User priorities are taken into consideration at the time of call admission. During the call-admission process, a high-priority user may force disconnection of a low-priority user call. Once admitted into

the network and active, all users receive their guaranteed QoS regardless of their priority level. Nevertheless, under extreme congestion conditions, the network may give preference to high-priority user cells if cells are to be discarded.

The priority controller supplies the admission controller with information on the user priorities in the network. This information is provided by a function of a network manager. This information is also utilized by the PMAC for discarding cells under instantaneous network congestion conditions.

5. ID-Assignment Control

An ID-assignment controller, located in each station, identifies the CP and the other remotes in the network at the network setup stage and in case the structure of the unit changes with time. The unique operational identifiers are used to identify the remotes for control and signaling purposes (e.g., during registration of a remote in the network, at the beginning of a call-setup procedure, etc.). The ID controller allocates MVCIs to admitted calls and releases them when the connections are terminated. It also allows translations among standard ATM addresses, operational unit identifiers, ATM-oriented VPIs/VCIs, and MATM-oriented MVCIs. The assignment of signaling identifiers is performed at the start of the network operation while the translation is an ongoing process.

6. Summary

This chapter has described several issues related to mobile integrated services networks with emphasis on a tactical framework. Such a network is expected to seamlessly integrate with a wireline network via a line-of-sight or satellite link to enable exchange of traffic with the external world. A protocol architecture for the mobile network is described. The proposed scheme is an extension of schemes proposed in the literature for wireless (as well as wireline) networks. User priorities are an important part of tactical environment, and the proposed scheme incorporates user priorities both for call admission and cell discarding during extreme congestion conditions.

III. MAC IN MOBILE INTEGRATED SERVICES NETWORKS

The medium-access-control protocol in a mobile integrated services network performs two essential functions. First, it provides a mechanism for the radio channel allocation among the nodes in the network (CP and remotes). The channel allocation is performed while maintaining the requisite QoS agreed upon at call setup for each active MVC. Second, it supports a control and signaling channel (in band or out of band) that is necessary for network functioning, such as remote registration or connection establishment.

In the proposed network, a remote station is assumed to maintain a radio connection only with the *mobile network coordinator* (MNC), although direct connections with other stations are possible as well. The MNC is part of the CP (see Section II.D). Consequently, and by using separate *downlink* and *uplink* channels, each transmission in the network passes through the CP. In fact, from operational view point, most traffic in the network is of type remote to CP or CP to remote. The MAC needs to support not only CP-to-remote and remote-to-CP connections, but also remote-to-remote connections and connections involving external users. Possibilities *for broadcast* and *multicast* are also desirable, especially for distribution of information from the CP to subsets of its forces. The high cost of the radio channel as a resource (due to its constrained bandwidth) forces the design of an efficient scheme to support all these requirements while maintaining high channel throughput. This implies reduced overheads, small (if any) number of collisions, etc.

Section A describes design requirements of the MAC protocol in a mobile integrated services network. Section B presents the terminology used throughout this work of the proposed MAC protocol and the aspects of its design. The issue of control signaling in the mobile network is thoroughly addressed in Section C. Next, Section D describes a detailed structure of the proposed MAC, followed by a description of the various information and control messages used by the protocol. Appendix A extends the discussion on the MAC, detailing a conceptual mobile-station database and intra- and

inter-station processes. Performance-related issues of the MAC protocol, such as the size of the admissible region and the channel throughput, are discussed in Section E.

A. DESIGN REQUIREMENTS

Medium-access-control protocol in mobile integrated services networks is required to perform several tasks. It is required to efficiently allocate the radio channel among the nodes in the shared medium, support control and signaling procedures necessary for the operation and management of the network, and support all possible end-to-end connections. Support of user priorities is desirable as well, especially in military networks. This section addresses in more detail the issues that influence the design of the MAC.

The main task of the MAC is to allocate the radio channel among the nodes in the network (CP and remotes). Once a call has been set up and a virtual channel has been established between the involved parties, the MAC divides the channel capacity in a way that maintains (for each MVC) the QoS requirements agreed upon at call establishment. The types of applications that need to be supported are speech conversation, real-time video, and bursty data. On the other hand, a request for channel allocation prior to remote registration or a request for channel allocation using illegal MVCI are to be ignored. (MVCI is used to indicate the virtual channel identification assigned by the *mobile* network to the mobile portion of the connection after call setup.)

In allocating the radio channel, the objective is efficient channel throughput or utilization. To achieve this, channel allocation for sources seeking channel access needs to be performed in an optimal fashion. (A detailed discussion on channel-allocation algorithms over wireless networks appears later in Chapter VI.) From a system architecture point of view, one can support this goal by reducing protocol overhead to the minimum possible, allocating bandwidth on demand, minimizing the number of separate transmissions (to eliminate preamble and guard time overheads), minimizing the number of collisions, and making use of broadcast and multicast as much as possible.

A signaling channel is essential in any network for effective control and management of the network. In wireline ATM networks, signaling cells (e.g., call setup, call release, etc.) are sent as standard 53-octet cells. The mobile portion needs to support these cells for connections involving external wireline users as well as other signaling messages specific to the mobile environment. Shortened mobile-control messages (rather than standard cells) should be used wherever possible. Dedicated MVCIs are allocated and used for all signaling information whether the MAC transports it in-band or out-of-band. Some procedures unique to the wireless channel are registration of a remote station with the CP, disconnection from the host network, and multiparty mobile connections. When using a reservation-based MAC, the control channel is also required to support reservation of the channel for information transfer, in order to reduce the number of collisions. Reduction of the overhead of the DLC sublayer may be achieved by defining mobile frames that include both access-control and error-control fields. This architecture somewhat deviates from the layered structure proposed by ISO/OSI but provides increased utilization of the channel.

All possible connection types must be supported by the MAC: CP to remote, remote to CP, remote to remote, broadcast, and multicast. The proposed network considers a logical star topology centered at the CP. Consequently, all remote-to-remote transmissions must pass through the CP even if a direct connection is possible between the remotes.

B. ADDRESS NOTATIONS

We propose an ATM-based, wireless, transport mechanism in which the basic unit is a complete ATM cell or a segment of an ATM cell. (See Chapter II for details.)

Unique ATM user addresses are 20-octet long [8]. It is very inefficient to send the addresses of the calling and called parties within every cell. ATM networks use a 24-bit VP/VC addressing (at UNI; expanded to 28 bits between network entities) in order to increase the efficiency of transportation [25]. In a mobile network where the channel capacity is very limited, imposing even stricter constraints on the address-field size is

desirable. At the MAC level, it is desirable to use the shortest addressing form possible in order to reduce overhead. In addition to the VP/VC identifiers of ATM, several addressing identifiers are used in the mobile network: remote-station operational ID, mobile signaling identifier (MSI) of a remote, and mobile virtual channel identifier (MVCI). Some additional address identifiers are found in ATM signaling: subassignment within the same MVCI used to distinguish different remote sources seeking new connection simultaneously, and mobile-channel allocation.

1. ATM Addressing

Each ATM end station requires a unique address [64]. Private and public networks use different ATM address formats. Public ATM networks use E.164 addresses (i.e., telephone numbers) whereas the addresses of private ATM networks are based on the OSI network service access point format but must support E.164 as well [13]. Due to their large size, ATM end-user addresses are used during the call-setup procedure only, to identify the calling and called nodes and sources.

2. Military Operational ID

In the mobile network, new or returning remotes are required to register in the network (see Section C for more details). A remote wishing to join the network identifies itself to the CP by its unique decimal 4-digit (16 bits) *operational-ID* number. This identifier is used by the CP to uniquely associate transmission/reception to/from that station.

3. Mobile Signaling Channel

Upon successful registration of a remote in the network, the CP assigns a 6-bit MSI[4] to it. MSI values 0, 1, 2, and 3 are reserved for use by the CP. Every remote has only one MSI associated with it for identification during *channel allocation* by the CP. The channel allocation is necessary for transmission of standard ATM signaling cells (see

[4] In the literature [25], the term *meta-signaling* is used to identify the process of negotiation on signaling VCI and resource allocation.

below). *Mobile signaling cells* on the downlink and uplink are identified by one of two forms. Prior to completion of registration in the network, the remote is identified and addressed by the CP using the remote's operational ID. After successful completion of the registration process, the remote is identified either by its assigned MSI (e.g., in a keep-alive message) or by the MVCI of the specific connection (e.g., in a channel-request message).

4. Mobile Virtual Channel Identifiers

Once a call has been set up, the network assigns two fields to the end users that participate in the connection; together, they compose the mobile user identifier (MUI). The fields are a 6-bit MVCI and a 4-bit identifier within the virtual channel (IDVC) as illustrated in Figure III.1. The MVCI field identifies the virtual channel number. Up to 60 channels are allowed in the network simultaneously. The four values of 0000XX (000000 through 000011) in the MVCI field are reserved for the mobile, signaling, virtual channel identifier (MSVCI). Within a MVC, the participating users are assigned unique IDVCs by the CP. For example, the CP may assign IDVCs to the users in a sequential manner. This addressing structure supports both point-to-point and point-to-multipoint connections having up to 16 parties per connection.

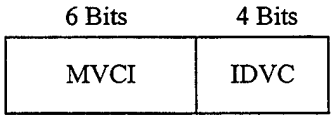


Figure III.1: Mobile User Identifier (MUI)

To increase the utilization of the network, we wish to use broadcast and multicast mechanisms for transmission of cells. For this purpose, a slightly different MVC, called extended MUI (EMUI) is used. The 6-bit MVCI field remains the same while the IDVC field is extended to a 16-bit bitwise-IDVC field as shown in Figure III.2. The bitwise-IDVC field is used to mark the destination users such that each party in the connection corresponds to one bit in the field (IDVC 0 corresponds to bit 0, IDVC 1 to bit 1 and so

on). The value 111...1 is therefore used as a *broadcast address* in the mobile network. This 16-bit notation is used only for cells sent from the CP to the remotes. When a remote sends a cell to another destination (via the CP), the bitwise-IDVC field is unused. When the CP receives a cell that has *at least one* remote addressee (i.e., cell to be relayed onto the downlink channel), it modifies the bitwise-IDVC field to include 1's at *all* the positions corresponding to the remotes taking part in the notified MVC. As shall be seen later, the CP functions as a "logical root" for transmission of cells even if it has not originally established the connection. All the remotes participating in that MVC (the *group* or *multiparty*) and receiving a cell correctly (other than the cell originator), with the corresponding bit in the bitwise-IDVC field turned on ("1"), pass the cell to the upper layers. This method allows the CP, in the case of a noisy channel, to retransmit cells *only to parties that have not acknowledged yet*, rather than resend it to all parties of the MVC. Within a multiparty connection, subgroups can be formed; thus, the information from one party within the group may be destined to a specific subgroup. In order to do that, each remote needs to maintain a table of the IDVCs of all parties within the connection.

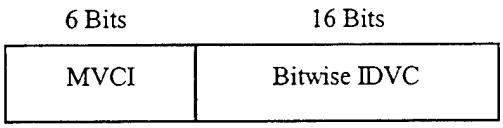


Figure III.2: Extended Mobile User Identifier (EMUI)

Besides the MVCI and bitwise-IDVC fields discussed above, a 4-bit source-IDVC field is added to a cell to indicate the source of the cell within a given MVC. This is used to ensure the received cell will be discarded at the *originating* remote.

5. ATM Signaling

In a wireline ATM network, a signaling cell originated by the ATM control unit arrives at the ATM layer as a regular cell with reserved VPI/VCI values (0/5, respectively). An ATM signaling cell is always destined to a single addressee.

In the mobile network, we take a different approach. The ATM control unit at each node, which assigns a VPI/VCI pair to a new connection, reserves the value VPI = 0 for ATM signaling purposes (over the wireless channel only). This value serves as the standard ATM signaling MVCI (SSMVCI) for standard ATM signaling cells over the wireless channel. When the ATM control unit at the CP has an ATM signaling cell[5] to transmit over the radio network, it sets the VPI field to zero and the VCI field to the addressee's MSI, and passes the cell to the ATM layer. When the DLC at the CP receives an ATM cell with VPI set to zero, it builds a mobile information frame with the SSMVCI and the MSI (obtained from the VCI field) as the identifiers of the destination user. The combination of these two fields, namely the standard ATM signaling mobile identifier (SSMI), then characterizes the source of the signaling cell. Since ATM signaling cells have a structure similar to information cells, we use the EMUI notation having SSMVCI and the destination's MSI fields as the cell identifiers (see Figure III.3). (Only the first 4 out of 16 bits of the bitwise-IDVC field are used.) In the remotes, since each signaling cell flows through the ATM control unit at the CP, both VPI and VCI fields are set to zero prior to transferring the cell to the ATM layer.

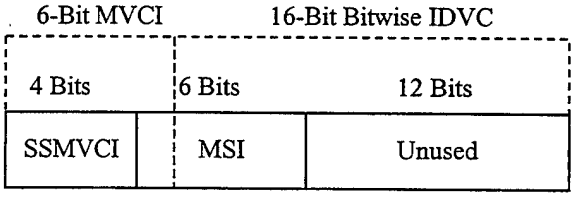


Figure III.3: Standard ATM Signaling Mobile Identifier (SSMI)

6. MVCI Assignment

A remote station in general may have more than one user/source. Since a single, mobile, signaling channel serves all the connections within the node, a mechanism is

[5] At the ATM UNI, signaling messages may contain hundreds of octets [8]; however, the ATM adaptation layer segments these messages into standard cells.

required to distinguish between the different sources. An ATM control unit handles the allocation of VPI/VCI at call setup on the wireline portion while a MATM controller handles the allocation of MVCI to a new connection in the mobile segment of the network. This requires synchronization between the two controllers. For example, the ATM control unit at the CP, which accepts a call from a remote, should be informed about the MVCI/IDVC pair that has been assigned to the calling user in that remote.

We propose a modification to the ATM control unit to include the capability to utilize the VPI/VCI fields to assign MVCI/IDVC in coordination with the MATM controller. The procedure for MVCI allocation is then being done as follows (see later in Figure III.7a for the case of a remote-to-CP connection). The remote sends a request for channel allocation using its MSI. After allocation of the channel by the CP, the remote sends a standard "Setup" signaling message [8]. At the CP, if the call is accepted into the network, the ATM controller requests an available MVCI/IDVC pair from the MATM controller and includes it in the "Connect" reply message (instead of the VPI/VCI fields). This way the remote obtains the MVCI/IDVC assignment as well.

7. Channel-Allocation Identifiers

The CP coordinates the transmission within the wireless channel using a reservation approach. A remote seeking to transmit any cells requests the CP for allocation of the channel. The CP in turn assigns the available capacity to the requesting users using channel-allocation identifiers. Here, we distinguish between channel allocation for ATM signaling cells and ATM information cells.

Whenever ATM signaling cells need transmission over a wireless channel, channel reservation and allocation are required. Figure III.4 illustrates a mobile-channel allocation procedure and the addressing format associated with it (detailed description of the messages transferred appears later in this chapter). First, the remote that has some (standard) ATM signaling information to send transmits a request for allocation on the uplink channel on, say, frame k, using the MSI addressing notation. If the request has been successfully received, it is acknowledged by the CP in frame k+1. The CP performs the channel allocation using SSMI format (Figure III.3). It sets the four most significant

bits (MSBs) of the MVCI field to 0000. The remaining two bits are appended to the IDVC field to form a 6-bit MSI for that remote. This 10-bit user identifier is used to synchronize the CP and the remote regarding channel allocation for the ATM signaling cells. In general, the assignment may take place on frame $k+i$, where $i > 1$, according to the channel-allocation scheme used by the CP and the current load in the network. Usually, the ATM signaling messages are organized in a dedicated FCFS queue, and channel allocation for them follows the order of arrival.

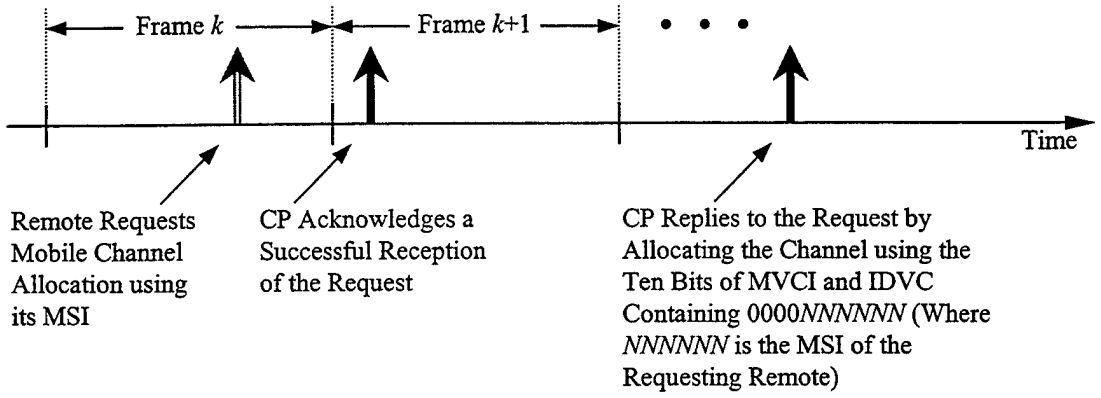


Figure III.4: Mobile-Channel Allocation Procedure for (Standard) ATM Signaling

In summary, once a call is admitted into the network, it is assigned a MUI (a 6-bit MVCI that identifies the source/destination within the connection and a 4-bit IDVC that differentiates between the parties of that call). This MUI is then used to allocate the channel to active sources.

C. SIGNALING IN THE MOBILE NETWORK

This section is devoted to control and signaling procedures in the mobile integrated services network. We consider the various procedures necessary for the appropriate operation of the network. The discussion is divided according to the stages of network operation: registration and disconnection, connection setup and release, mobile-

channel allocation, and error control. The issue of multiparty connections is also discussed.

1. Registration and Disconnection Procedures

The following describes the remote-station registration and disconnection procedures associated with the existence of remotes in the network. For simplicity, the channel-allocation requests and replies are discarded in the discussions.

On power-up or when joining another unit's network, the remote and the CP follow a message exchange sequence in order to complete the registration procedure as shown in Figure III.5 [101] [3]. The remote sends a REGISTER_REQUEST message containing its operational ID (equivalent to the unique mobile ID [3]) to the CP. After receiving this message, the mobile admission controller at the CP queries the network manager in order to decide whether or not to admit the remote into the network. Such a decision is based on operational circumstances only. Up to 63 remotes may be registered at a given time in the network. The CP then sends its response in a REGISTER_REPLY message to the remote. If the remote is admitted, a 6-bit MSI is allocated to it. The remote then confirms the acceptance into the network by sending a REGISTER_CONFIRM message. In case it is rejected (e.g., the network is full, unauthorized remote, etc.), the cause of rejection is supplied by the CP through REGISTER_REPLY. Alternatively, a REJECT message (which is defined in the system as a general error message) can be used for the reply. For *authentication* purposes, the remote may be inquired by the CP to submit specific identification information prior to the REGISTER_REPLY message (indicated in Figure III.5 as dashed AUTHEN_REQUEST and AUTHEN_REPLY messages).

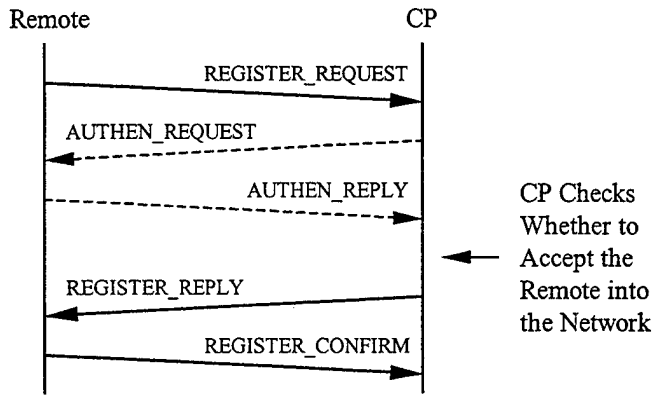


Figure III.5: Registration Procedure Diagram [101]

Disconnection is an event associated with a registered remote, which occurs when such a remote disconnects from the network. If the disconnection is done properly or purposely, the remote sends to the CP an EXIT_REQUEST message as shown in Figure III.6a. The CP then responds with an EXIT_REPLY message detailing whether the request is accepted or rejected (and the reason for that, if any). If accepting the disconnection request, the CP ensures that all resources associated with that remote are released.

Improper disconnection usually happens due to power failure or sudden uncontrollable degradation of signal quality [101]. Such situations are required to be detected reliably such that the CP can release all the resources tied with the disconnected remotes. A remote sends KEEP_ALIVE messages to the CP periodically if *no* other transmission has occurred during a pre-determined timeout (see Figure III.6b). The CP sets a *disconnection* timeout for each registered remote, and if no transmission is received within this period, the CP assumes that the remote has been improperly disconnected from the network. The timeout is set to be a small multiple of the KEEP_ALIVE timeout, to avoid inappropriate disconnection of a remote due to lost KEEP_ALIVE messages. In order to save battery power, the KEEP_ALIVE timeout can be longer for registered remotes having no active connections. A remote that was disconnected from the network and wishing to rejoin it needs to repeat the entire registration process discussed above.

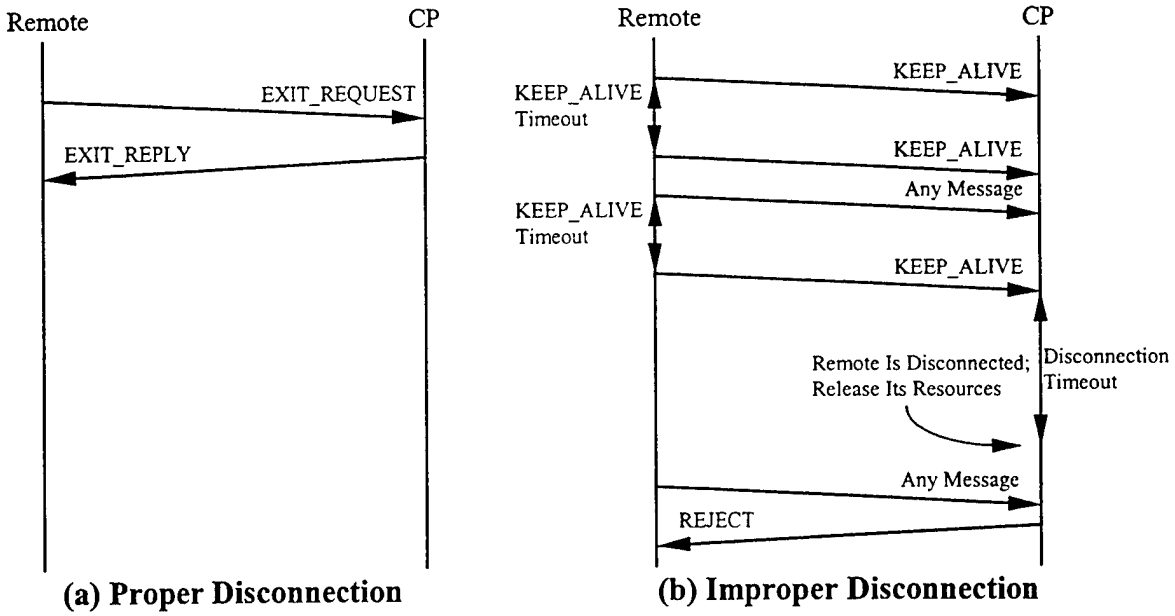


Figure III.6: Disconnection Procedure Diagrams

2. Call-Setup and Call-Release Procedures

A registered remote that wishes to go active and set up a new call needs to request allocation of information slots to transmit its setup control information (traffic descriptors, QoS requirements, etc.). This is achieved by sending an ALLOCATE_REQUEST message with the MSI of the node on the uplink channel (see Figures III.7a and III.7c for examples of remote-to-CP and remote-to-remote setup procedures, respectively). The CP responds with a LAST_FRAME_ACK message in the following frame on the downlink channel. The remote then waits for channel allocation by the CP, which uses the MSVCI and the remote's MSI as its identifiers. The allocation itself is set by the CP on the frame header of the frame following the request *or* any frame(s) afterward. A remote is allowed to start multiple call-setup procedures [8]; however, the signaling cells need to be multiplexed between them (usually FCFS) since a single signaling channel is used for all sources within the remote. A remote receiving channel allocation sends its "Setup" signaling message that is responded by an assignment of MVCI/IDVC by the CP in case the call is admitted. The admission controller informs the MAC about the allocated MVCI and the class of service, upon

acceptance of the call into the mobile network. The class type is translated into QoS requirements of the connection to be used for appropriate channel allocation by the PMAC; only a finite number of traffic classes is allowed in the network. For a mixed stationary and mobile connection, QoS requirements can be defined separately for each of the wireline and mobile segments. The MVCI/IDVC pair is stored in the connection-identifier field of the "Setup" message while external sources within the connection use the standard VPI/VCI notation. The rest of the setup procedure follows on the lines of regular ATM, which consists of signaling messages "Setup," "Connect," "Call Proceed," and "Connect Ack." In case a remote is seeking to transmit any of these control messages, channel-allocation request and reply are preceded the actual signaling-message transmission in the channel. The last transmission of any signaling message within the setup procedure is performed by the CP to each participating remote ("Connect" or "Connect Ack"). These messages contain a call-reference field that serves as a local identifier of the VC/MVC that has just been established. At the time the call is released, this field is used in order to relate to the appropriate VC/MVC of the remote.

For the case in which the CP initiates a call (CP-to-remote connection), there is no need to send any control information (channel allocation, setup, etc.) prior to the admission decision. The call parameters are transferred within the CP internally, and an admission decision is made as shown in Figure III.7b. If the call is accepted in the network, the CP sends the "Setup" signaling message together with the assigned MVCI/IDVC to the called remote. In a wireline ATM network, such a situation cannot happen since the switch at the access point of the network can never initiate a call. Figure III.7c illustrates the case of a remote-to-remote call-setup procedure; the second part is similar to the CP-to-remote connection case.

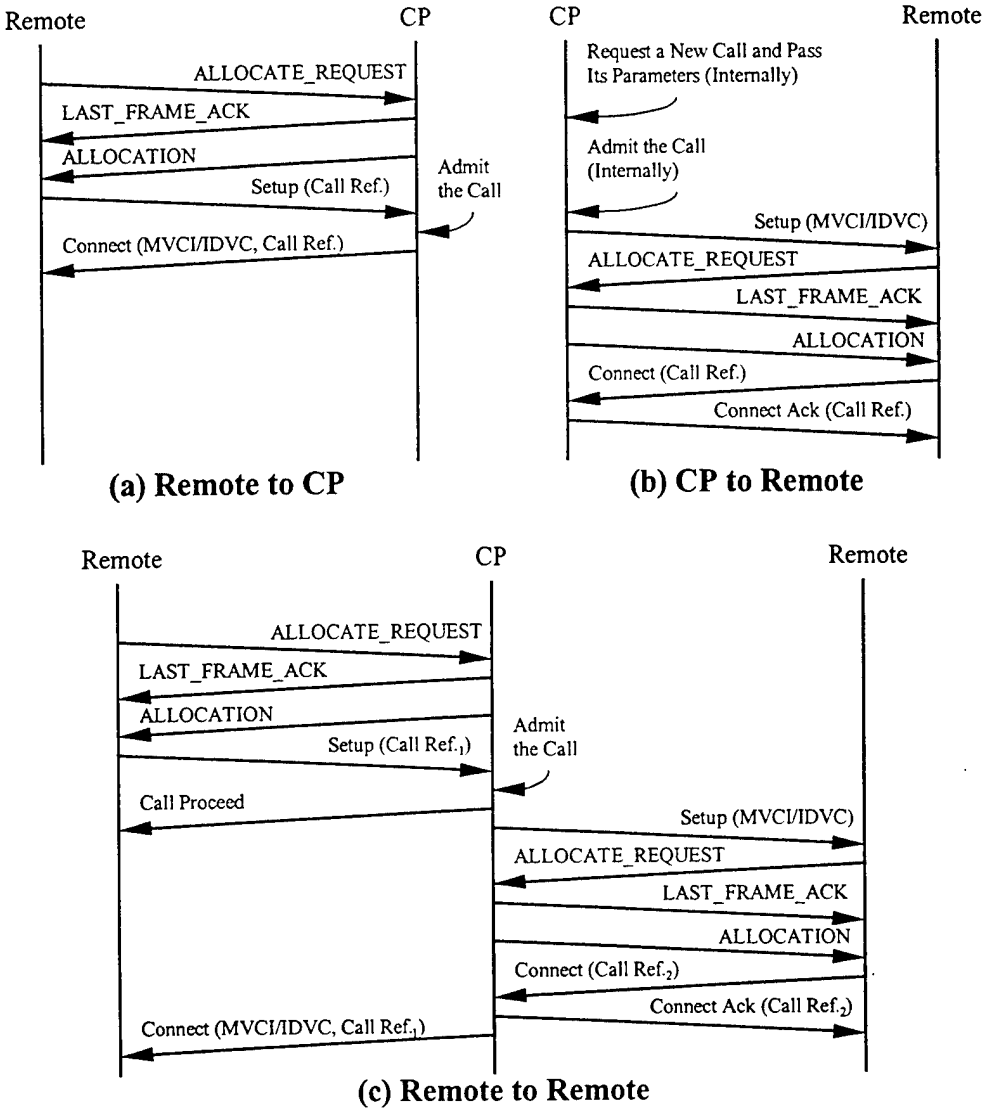
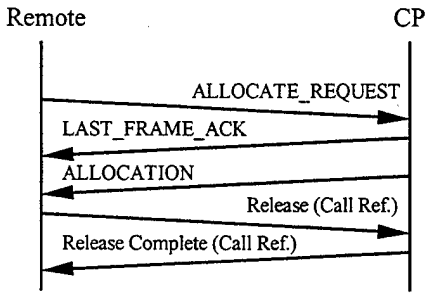


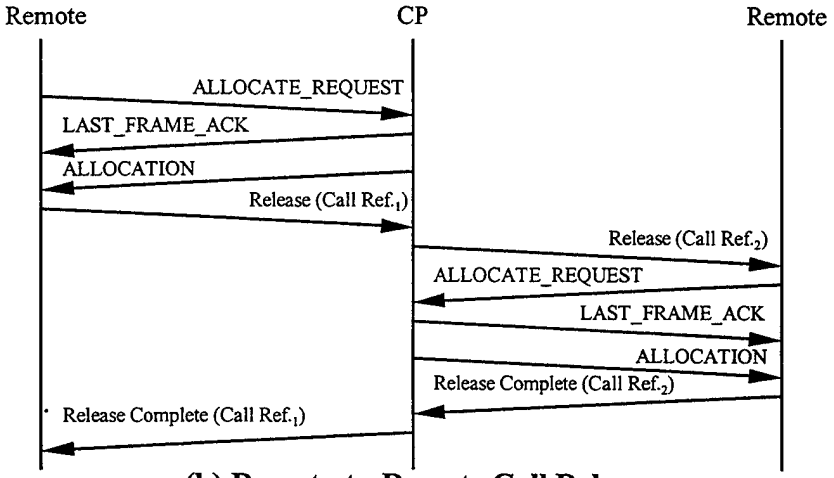
Figure III.7: Call-Setup Procedure Diagrams

A call is released by any of its parties using the procedures shown in Figure III.8. If a remote wishes to tear down a connection, it sends a standard ATM "Release" signaling message to the CP with the call-reference field assigned to the connection (see Figure III.8a). The CP then replies with a standard ATM "Release Complete" signaling message using the same call-reference field, and the connection is released. In a remote-to-remote call, the request and the reply of the call teardown must pass through the CP as shown in Figure III.8b. The CP manages a different call reference against every party of

the connection. When a party is released, its IDVC becomes available for use in the network; when all parties are released, the MVCI becomes available as well. The information regarding the resources that have been released is obtained by the ATM controller at the CP (through the signaling messages it exchanges with the remotes) and passed to the MATM controller for updating purposes.



(a) Remote-to-CP Call Release



(b) Remote-to-Remote Call Release

Figure III.8: Call-Release Procedure Diagrams

3. Multiparty Connection Procedures

A multiparty connection, such as video teleconferencing, allows the information from the source to be spread to more than one destination. In ATM, only point-to-point and point-to-multipoint connections are allowed [8]. We adopt these connection types in the wireless network as well. According to ATM definitions for a point-to-multipoint

connection, the endpoints form a logical tree topology that has one *root* (the originating node) and several *leaves* (the other parties) [14]. In this approach, when a root node sends traffic cells all other nodes receive copies; when the root receives information from one of the leaves, it extracts a copy for itself and spreads the information to all the other connection participants. No direct communication is allowed between the leaves.

The terminology of the wireline ATM is slightly modified for use in the mobile network. The originating remote is still the root concerning call establishment, call release, and addition and deletion of parties. However, the CP *always* functions as the logical root for information transmission purposes for all remotes within the network (it may not be the root for all parties, especially if external sources are involved). If there is another root in the wireline segment, the CP delivers a copy to it as well. The standard ATM control unit needs to be modified to include relaying a copy of these cells to all parties.

A multiparty connection is obtained by a regular point-to-point connection-setup procedure, followed by a series of *add-party* ATM signaling procedures. Only the root may add parties to the connection. The QoS parameters of the message sent by the root adding a leaf must be the same as those of the original call.

We describe here the procedure for the case of a remote (root) that has originated a call and wishes to add another remote party to it. A time-sequence diagram describing the procedure is shown in Figure III.9. Other processes for different scenarios follow in a similar manner. The root remote issues a standard "Add-Party" signaling message to the CP, with the call reference of the already active connection. The CP then issues a "Setup" signaling message to the called remote with the same MVCI and an available IDVC. "Connect," "Connect Ack," and "Add Party Ack" messages complete the procedure.

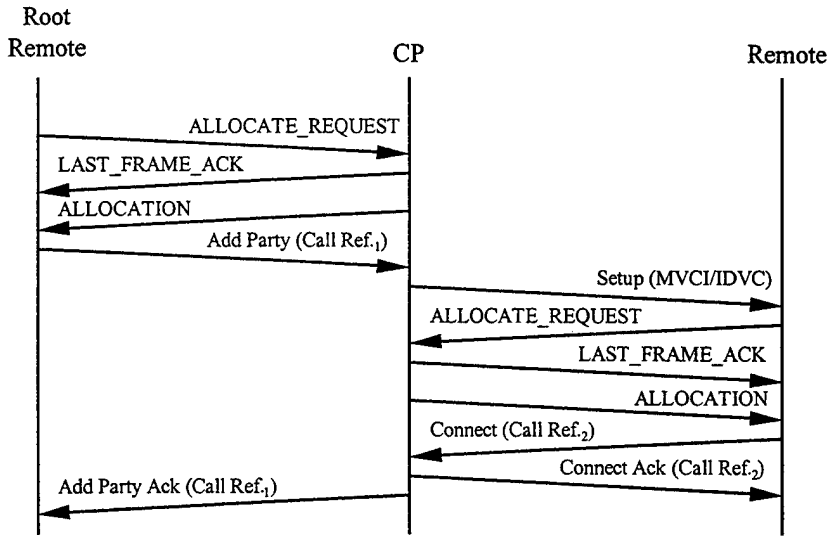


Figure III.9: Add-Party Procedure Diagram

A connection may be partially or fully terminated in one of two ways. If a party wishes to be released from the connection, it follows the previously described call-release procedure. If the root of a multiparty connection decides to drop a specific party, it follows a *drop-party* procedure shown in Figure III.10, which provides a function opposite that of the add party. One or more parties can be dropped from the connection using "Drop Party" (from a remote) and "Release" (from the CP) signaling messages. Both messages use the call-reference field to handle this.

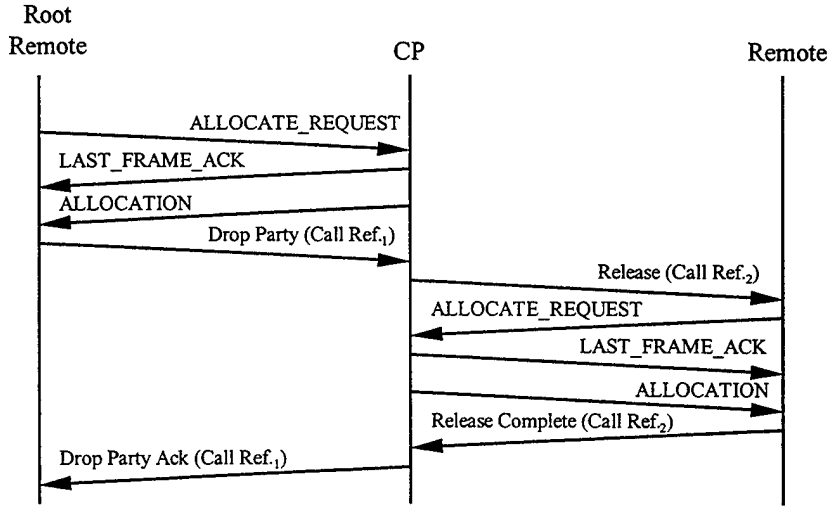


Figure III.10: Drop-Party Procedure Diagram

4. Mobile-Channel Allocation Procedures

The uplink channel is divided into control and information subchannels. A remote wishing to transmit on the information subchannel contends on the control subchannel using a *slotted-ALOHA* [78] or a *binary-stack* [46] protocol scheme. The throughputs of the two seem to be similar; however, the latter is more stable [46].

A static allocation of control slots to the remotes is not possible for two reasons. First, at a given time there may be nodes that are not part of the network yet. Second, on the downlink channel, we use a shortened identifier for the remotes (10 bits) rather than the operational ID (16 bits). Thus, even for a registered remote, there is a problem to "locate" the control slot of a source that wants to go active. The problem can be solved by assigning a *core ID* to a remote at registration time, to indicate the location of the control slot allocated to it. This however forces constant allocation of 60 control slots (as many as the number of possible MSIs), hence degrades channel throughput considerably. We therefore adopt a contention-based scheme on the uplink control subchannel as proposed in the literature [76] [53] [102].

The uplink control subchannel is used to request allocation of information slot(s) (using the ALLOCATE_REQUEST message with the MVCI) when the queue of the requesting remote has been empty and the first cell arrives. The CP responds to the remote immediately with the LAST_FRAME_ACK message on the downlink control subchannel of the next frame. This indicates to the contending remotes if a retransmission of their request is needed or not. (A LAST_FRAME_ACK message is sent only if at least one control message on the uplink channel has been successfully received.)

Allocation for transmission (if any) is set by the CP on the current or future frame(s) header using the MVCI/IDVC (MUI) notation discussed in the previous section. The channel is allocated to the different sources based on the *channel-allocation* algorithm used at the MAC. (Several allocation or scheduling schemes are described later in Chapters V and VI.) Since the nature of the traffic streams is not deterministic, the CP shall allocate *all* the vacant slots on the uplink channel (even if they have not been planned for specific, known, waiting cells) for remote usage, in order to avoid any

channel resource waste. Extra slots are allocated among the active sources, e.g., based on their mean arrival rate. A remote that sends a cell and whose queue has not been empty can piggyback allocation requests in the cell header for additional information slots. These piggyback mechanism significantly reduces the number of remotes contending on the uplink control subchannel [46].

Prior to transmission of the ALLOCATE_REQUEST message, the remote checks whether or not the message is valid. There are two scenarios, in which this control message is not valid thus discarded:

- A cell arrives during the uplink information subchannel at the MAC whose information queue is empty. If the CP has been allocating extra information slots to the remote, the remote uses one of those slots to transmit the cell in the current frame. The information queue becomes empty again, but the ALLOCATE_REQUEST message associated with it is still waiting for transmission.

- A cell is passed for transmission, and an ALLOCATE_REQUEST message is generated. By the time the control message is sent, the allowed cell transfer delay of the cell has expired, and the cell has been discarded. If the queue is empty, the ALLOCATE_REQUEST message must be discarded.

5. Error-Control Procedures

Error control is used in the mobile network to overcome the impairments imposed by the radio channel. The error-control scheme uses positive or negative acknowledgments to indicate successful or unsuccessful receptions, respectively. Transmission on the downlink or uplink follows different procedures, e.g., remotes must seek channel allocation prior to cell transmission, thus we can distinguish between two error-control policies.

The CP informs at the beginning of each frame the results of remote transmissions on the last frame via two LAST_FRAME_ACK messages. One message, corresponding to the uplink *control* messages, is sent if the size of the uplink control subchannel of the previous frame was greater than zero. The other message, corresponding to the uplink cells, is sent if at least one *successful* cell has been received by the CP on the previous

frame. This scheme thus allows an immediate feedback to the remotes as per their transmissions in the previous frame and enhances the loss performance in the network.

In the uplink channel, a different mechanism is used for error control. A *sliding window* of size 17 is used for (*selective-reject*) retransmission of cells from the CP to the remotes [78]. A remote responds to cell(s) sent from the CP by sending a GROUP_ACK control message. This message indicates which of the last cells has been successfully received. A GROUP_ACK message is sent by the remote only if *at least one* cell in the window has been successfully received. Thus, the CP needs to employ a retransmission mechanism based on timeout expiration, for cases in which none of its "windowed" cells have been successfully received. In a multiparty connection, the GROUP_ACK message and the bitwise-IDVC field in the cell header are used for retransmission to the non-acknowledging remotes only, rather than to all the original addressees of the cell.

The discussion above applies to both information and control ATM cells. Although ATM signaling cells are generally sent in wireline networks under best-effort-oriented services, the DLC/MAC have no way to distinguish them from information cells. Thus, error control is also applied for the signaling cells over the mobile network. For MATM signaling messages on the other hand, no positive or negative acknowledgments are returned by the receiving nodes.

D. MAC STRUCTURE

This section details the structure of the MAC. The formats of time frames utilized by the MAC as well as a detailed description of the subchannels composing the frame are covered. All possible protocol data units (PDUs) of the protocol including field contents are explained.

The structure of the MAC protocol is based on TDMA/TDD. The time is divided into *frames*, 12 milliseconds long each (12,000 bits). The channel is assumed to be of capacity 1 Mbps. We define a group of 60 consecutive bits to be a *control slot* (an information slot accommodates an ATM cell plus mobile-channel overheads). We have a total of 200 control slots per frame.

The frames are divided into downlink and uplink channels as illustrated in Figure III.11. Cells from the CP are sent on the downlink information subchannel while remotes share the uplink information subchannel. Control subchannels are allocated on each link for signaling and data-link-control purposes. Each transmission begins with a preamble sequence, 40 bits long, followed by an 8-bit channel-indicator (CI) field. The latter indicates the type of the transmission as shown in Table III.1 (the remaining six bits are used for error detection/correction). The field is mainly used by the remotes joining an existing network to immediately align their MAC protocol to the position within the frame. On the downlink, the data is sent by the CP only; thus, only one set of preamble sequence and guard period is required. On the uplink, on the other hand, each transmission from a different remote requires a separate set of preamble and guard time.

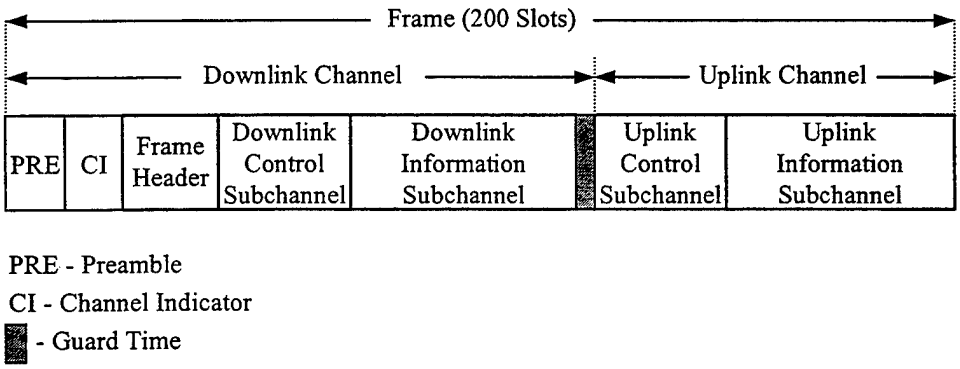


Figure III.11: MAC Frame Format

(Sub)Channel Indicator	Field's Value
Downlink	00
Uplink Control	10
Uplink Information	11

Table III.1: (Sub)Channel-Indicator Field Values

1. Frame Header

The frame header contains the sizes of each of the four subchannels and allocation for the remote sources on the uplink information subchannel. The structure of the frame header is shown in Figure III.12. The size of each field is dynamic in nature and depends on the instantaneous load in the network. The allocation of control subchannels aims to allow smooth operation of the system from control and management points of view. On the other hand, the allocation of information slots is based on a pre-determined channel-allocation scheme (see Chapter VI) in order to achieve the largest channel throughput possible.

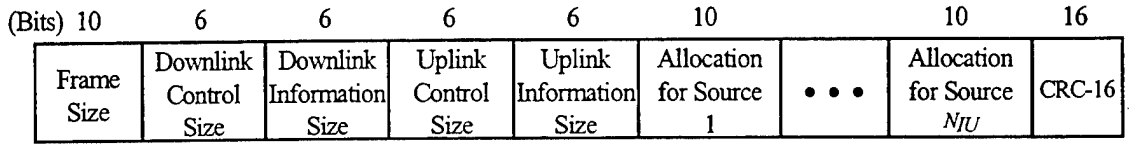


Figure III.12: MAC Frame Header

The length of the frame header is $50+10N_{IU}$ bits, where N_{IU} is the number of information slots on the uplink information subchannel. For the default $N_{IU} = 11$, this length is 160 bits or 2.67 control slots. The frame header contains the following fields:

- Frame size: The size of the frame (in control slots). The maximum value is 1023 (61.38 milliseconds) and the default value is 200 (12 milliseconds).

- Downlink control size: The size of the downlink control subchannel in units of mobile-control messages. Each downlink control message occupies exactly one control slot (60 bits), thus this number is given in control slots. The maximum value of the field is 63 and its default value is 4 (4 slots).

- Downlink information size: The size of the downlink information subchannel in units of cells. Each downlink cell contains one ATM cell plus overhead and occupies 7.6 control slots (456 bits). The maximum value of this field is given by $\lfloor [200 - 0.8 - (48 + 25 \times 10)/60 - 0.1]/7.6 \rfloor = 25$ (190 control slots) and the default value is 11 (83.6 slots).

- Uplink control size: The size of the uplink control subchannel in units of mobile-control messages. On the uplink channel, a control message (occupying one control slot) is preceded by a preamble and CI field (0.8 control slot

together) and followed by a guard time (0.1 control slot). Thus, the length of an uplink control message is 1.9 control slots. The maximum value of this field is 63 and the default value is 8 (15.2 slots).

- Uplink information size: The size of the uplink information subchannel in units of cells. An uplink cell contains one cell, overhead, preamble, and guard, resulting in 8.5 control slots (510 bits). The maximum value of the field is given by $\lfloor [200 - 0.8 - (48 + 22 \times 10)/60 - 0.1]/8.5 \rfloor = 22$ (187 control slots) and the default value is 11 (93.5 control slots).

- Allocation for source i ($1 \leq i \leq N_{IU}$): Each field contains six bits of MVCI plus four bits of IDVC that together uniquely identify the source to which an information slot is being allocated. Note that N_{IU} is the value that appears in the uplink-information-size field. Up to 22 information slots may be allocated on the uplink information subchannel.

- CRC-16: 16-bit cyclic redundancy code (CRC) for detection of errors in the frame header.

In setting the default values for the subchannels, we have made two implicit assumptions. The user information load on the uplink and downlink are equal; thus, the number of information slots on both subchannels is the same. Most of the control messages are of type ALLOCATE_REQUEST (on the uplink) or LAST_FRAME_ACK (on the downlink). Other control messages, such as registration request or exit request, are seldom generated and transmitted. Consequently, since ALLOCATE_REQUEST message is more frequent than the LAST_FRAME_ACK, the number of control slots on the uplink is set to a larger value than that of the downlink.

2. Downlink Control Subchannel

The downlink control subchannel comprises N_{CD} independent control slots. These are transmitted in sequence without any separation between them as depicted in Figure III.13. However, if no control messages are to be transmitted, the CP can allocate the time space of this subchannel for use by other subchannels or it can shorten the current frame length. Alternatively, if the channel-allocation algorithm uses a fixed value for N_{CD}, the CP pads the unused messages with 0's or 1's to maintain bit timing by the remotes.

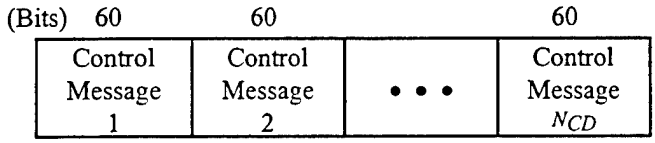


Figure III.13: Downlink-Control-Subchannel Frame Structure

Several mobile-control messages are defined in the MAC protocol. Each mobile-control message has a mobile-payload-type (MPT) field (first four bits) that is used to indicate the type of the message. The length of each is one control slot (60 bits). Each control message ends with a CRC-16 field used for error detection on the control message body. LAST_FRAME_ACK and MATM_SIGNALING are typical control messages.

a. LAST_FRAME_ACK

Figure III.14 shows a control message of type LAST_FRAME_ACK. The CP sends two types of this message: CONTROL_ACK and INFORMATION_ACK. The ACK-type field identifies the message type.

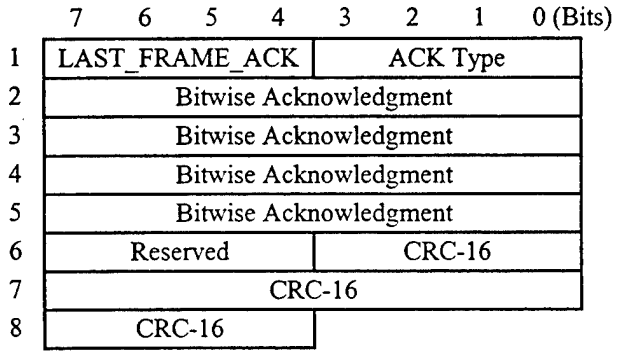


Figure III.14: LAST_FRAME_ACK Control Message

The CONTROL_ACK message is sent in a given frame if the following two conditions are satisfied: the size of the uplink control subchannel of the previous frame is greater than zero, and at least one mobile-control message on the uplink channel has been successfully received. The bitwise-acknowledgement field is used to indicate

which of the control messages sent on the previous uplink control subchannel has been successfully received by the CP. This way, the remotes get immediate feedback about transmission of control message in the previous frame. If a channel-allocation request fails, a retransmission may take place immediately, even for real-time services, such as speech. Up to 32 control messages can be acknowledged by a single LAST_FRAME_ACK message of type CONTROL_ACK. Starting from the MSB of the second octet and up to the least significant bit (LSB) of the fifth octet, a "1" bit indicates successful reception and a "0" otherwise, corresponding to the uplink control messages in the previous frame.

The INFORMATION_ACK message is transmitted in a given frame provided *at least one* cell has been successfully received by the CP in the previous frame. It indicates which cells sent on the previous uplink information subchannel have been successfully received by the CP. If transmitted, this message immediately follows the CONTROL_ACK message (if the latter is transmitted) on the downlink control subchannel. Up to 22 cells (the maximum possible number of cells in one frame) can be acknowledged by a single LAST_FRAME_ACK message using the bitwise-acknowledgement field. The corresponding field is from the MSB of the second octet to Bit 2 of the fourth octet; a "1" bit indicates a successful reception and a "0" otherwise, corresponding to the uplink cells in the previous frame.

b. MATM_SIGNALING

Another type of control uplink message is the MATM_SIGNALING message. The structure of this message, shown in Figure III.15, is common for both downlink and uplink channels. The message contains several subtypes that are identified according to the mobile-signaling-type (MST) field. The source-identifier field is used to identify the *destination* of the mobile signaling message. The field may receive one of two forms according to the message subtype. The first form is the remote's operational ID, which is used in messages that are sent when neither the MSI nor the MVCI/IDVC is available. Example of such a message is a request for registration within the network. The

second form is the shortened MSVCI (MVCI/IDVC) after registration (call establishment). Both forms are used to identify the *source* of the mobile signaling message. In both cases, however, the 16-bit source-identifier field is allocated for identification purposes. The following describes each of the available subtypes of the MATM_SIGNALING message.

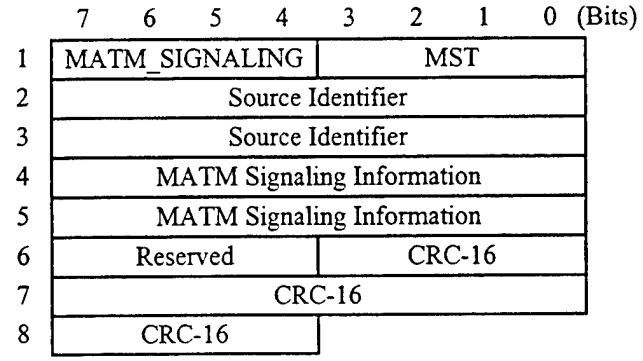


Figure III.15: MATM_SIGNALING Control Message

(1) REGISTER_REPLY. This control message is sent by the CP in response to a request by a remote to join the network using REGISTER_REQUEST message (see below). The MST field contains the value REGISTER_REPLY, and the source-identifier field contains the remote's operational ID. The possible contents of the MATM-signaling-information field are ACCEPTED plus a 6-bit MSI that has been assigned to the remote or REJECTED plus the rejection cause.

(2) EXIT_REPLY. The CP sends this control message in response to a request by a remote to leave the network using EXIT_REQUEST message (see below). The MST field contains the value EXIT_REPLY, and the source-identifier field contains the remote's operational ID (the remote's MSI may be used here as well).

(3) REJECT. If a non-registered remote sends a control or information message, the CP responds with the REJECT control message. The MST field contains the value REJECT, the source-identifier field includes the remote's operational

ID, and the MATM-signaling-information field includes the cause of the rejection (e.g., UNREGISTERED_REMOTE). The possible contents of the MATM-signaling-information field are ACCEPTED plus a 6-bit MSI that has been assigned to the remote or REJECTED plus the rejection cause.

3. Downlink Information Subchannel

The downlink information subchannel includes N_{ID} identical and independent cells. These are transmitted contiguously without any separation between adjacent cells as shown in Figure III.16. However, if no message is available for transmission, the CP can allocate the time space of this subchannel for use by other subchannels, or it can shorten the current frame length. Alternatively, if the channel-allocation scheme uses a fixed value for N_{ID}, the CP pads the unused messages with 0's or 1's to maintain bit timing by the remotes.

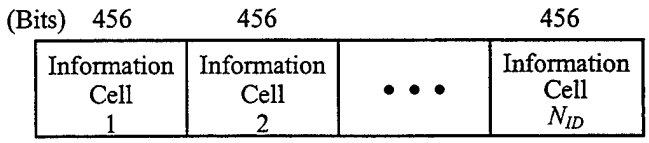


Figure III.16: Downlink-Information-Subchannel Frame Structure

Structure of a mobile cell transferred on the downlink information subchannel is shown in Figure III.17. Each cell is 456 bits long (7.6 control slots). A mobile-payload-type (MPT) field is used to indicate the type of the cell (user information or ATM signaling), although this information may be obtained from the MVCI and IDVC fields. A cell ends with a 16-bit CRC-16 field used for error detection on the entire cell body.

A mobile cell contains a compressed ATM-like header together with the payload and the CRC as seen in Figure III.17 [78]. The ATM-payload-type (APT) and cell-loss-priority (CLP) fields are copied from the ATM header. The generic-flow-control (GFC) and header-error-control (HEC) fields are not transmitted over the wireless link and have been discarded (the latter due to the CRC-16 error-detection field provided by the DLC [100]). The MVCI and bitwise-IDVC fields uniquely identify the addressees of the

message. The regular IDVC field is extended here to 16 bits (bitmap layout corresponding to the 16 possible IDVCs). This allows the CP to retransmit a cell using a multicast form only to remotes that have not yet acknowledged the reception rather than to all original addressees of the cell. The message-sequential-number (MSN) field corresponds to the identifier of the first cell to be acknowledged. The cells are numbered by the sending DLC in a cyclic manner using eight bits. The segment-counter (SC) field that contains two bits, completes the cell identifier if the cell is segmented (up to four segments of a cell are allowed). The source-IDVC field contains the IDVC of the originating source of the cell. It is used to discard the cell at the originating remote in case the cell has been relayed by the CP. The piggybacked-allocation-request (PAR) field is unused here but applicable in the case of the uplink information subchannel.

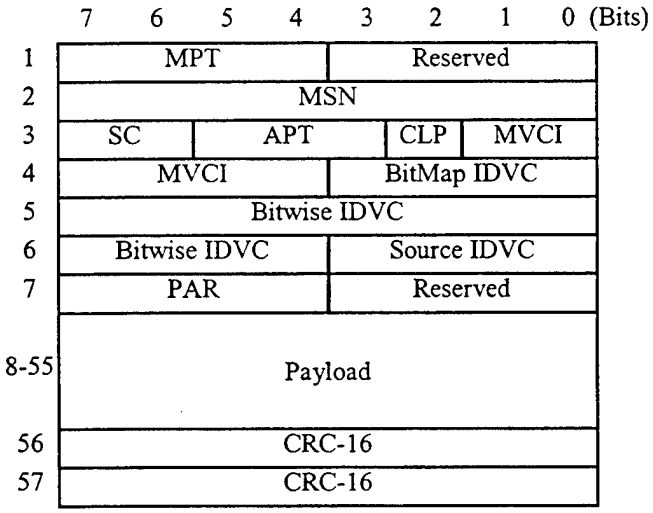


Figure III.17: Mobile Cell

4. Uplink Control Subchannel

The uplink control subchannel comprises N_{CU} independent control messages as shown in Figure III.18. We allocate the entire "pool" of uplink control slots to all the remotes via contention. If a remote has more than one control message to transmit at a time, it has to contend over several slots within a single frame or over several consecutive

frames. If a remote has no control messages to transmit, it just remains silent. Each control message on the uplink channel is preceded by a preamble sequence and a CI field (see Figure III.18), and followed by a guard period. The available types of control message are GROUP_ACK and MATM_SIGNALING.

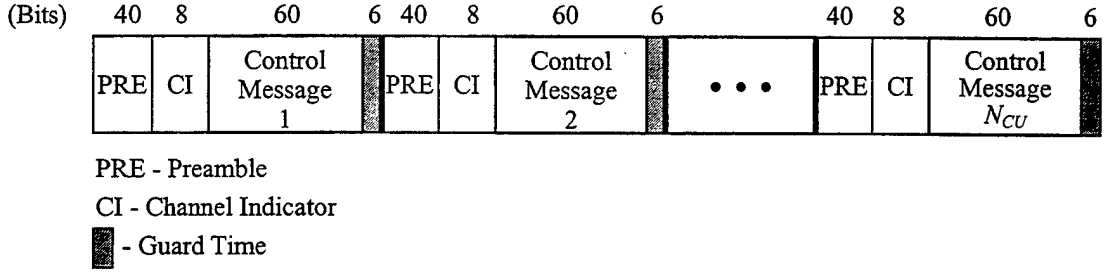


Figure III.18: Uplink-Control-Subchannel Frame Structure

a. GROUP_ACK

The structure of GROUP_ACK is shown in Figure III.19. It is sent by a remote to acknowledge up to 17 cells of a specific virtual channel for error-control purposes. The destination of the GROUP_ACK message is obtained by the combination of MVCI and IDVC fields, as discussed earlier. Fields MSN and SC together identify the first acknowledged cell of the source; hereafter, a "1" bit in the bitwise-acknowledgment field corresponds to consecutive MSNs. For example, if cells 45, 46, 48, 49, and 61 are to be acknowledged, then Octets 3, 4, and 5 of the control message contain 45 (decimal), 10110000, and 00000001, respectively.

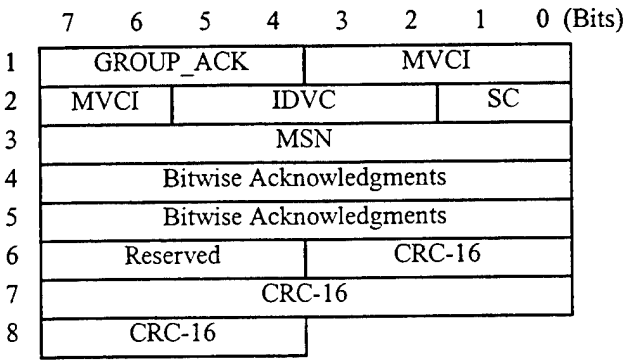


Figure III.19: GROUP_ACK Control Message

b. MATM_SIGNALING

The structure of this message is given in Figure III.15. The source-identifier field is used to identify the *source* of the message (i.e., the sending remote or service within the remote) rather than the *destination* as in the downlink case. The available subtypes of the message are given below.

(1) REGISTER_REQUEST. This message is sent by a remote wishing to join the network. A remote may be allocated time slots for information transmission only after it has been successfully registered in the network by the CP. The MST field contains the value REGISTER_REQUEST, and the source-identifier field includes the remote's operational ID.

(2) REGISTER_CONFIRM. This message is sent by the remote in response to a REGISTER_REPLY message from the CP. It confirms that the remote has received the message and accepts it. The MST field contains the value REGISTER_CONFIRM, and the source-identifier field includes the remote's operational ID.

(3) EXIT_REQUEST. This message is sent by a remote prior to leaving the network (proper exit procedure). The CP responds with an EXIT_REPLY

control message. The MST field contains the value EXIT_REQUEST, and the source-identifier field includes the remote's operational ID.

(4) KEEP_ALIVE. This message is sent by a remote to indicate to the CP that it is alive. The message is sent after a pre-determined timeout in which the remote has transmitted neither a control message nor a mobile cell. The MST field contains the value KEEP_ALIVE, the source-identifier field includes the remote's operational ID, and the MATM-signaling-information field contains a short form of the *time of day* at which the message is sent.

(5) ALLOCATE_REQUEST. This message is sent by a remote seeking channel allocation for transmission of cells of any type (signaling or data). The message can be sent only after an appropriate registration procedure has been completed by the remote and a MSI has been assigned; otherwise, a REJECT message is returned by the CP. The ALLOCATE_REQUEST message is transmitted only if a new cell is enqueued into an empty queue in the remote. If the queue of a remote is not empty, it includes a piggyback request for additional allocation(s). A remote is allowed to send only *one* ALLOCATE_REQUEST control message per frame per source. This ensures that no wasteful transmissions are made before the CP is able to respond to the request in the frame header of the next frame. Alternately, the remote can request within one ALLOCATE_REQUEST message multiple information slots. This approach forces the remote to build an ALLOCATE_REQUEST message a short time prior to its transmission such that "enough" cells are accumulated in the queue. This way the request is transmitted efficiently rather than for every individual arriving cell. The remote contends on the uplink control subchannel using the MSVCI and MSI for identification.

The MST field contains the value ALLOCATE_REQUEST. The source-identifier field includes the MVCI and IDVC of the source requesting the allocation. The MATM-signaling-information field contains the number of information slots requested by the source and/or by the remote or some other information required by the channel-allocation algorithm.

5. Uplink Information Subchannel

The uplink information subchannel includes N_{IU} identical and independent cells as illustrated in Figure III.20. The order in which the cells are transmitted is determined by the CP using the frame header portion of the frame, as discussed above. Each cell on the uplink channel is preceded by a preamble sequence and a CI field, and followed by a guard period.

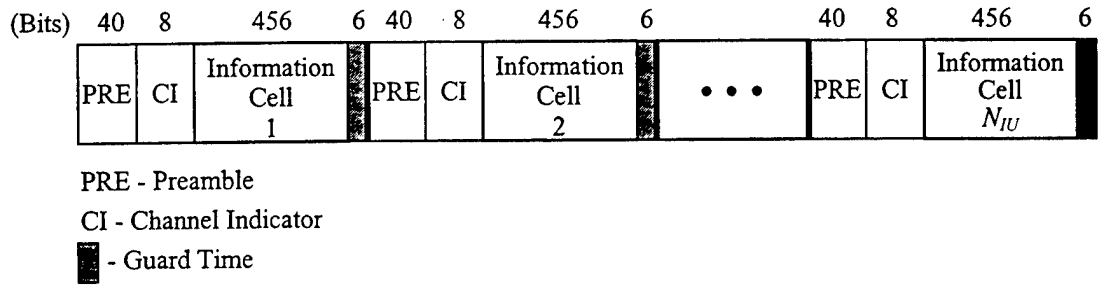


Figure III.20: Uplink-Information-Subchannel Frame Structure

The structure of the cell on the uplink channel is identical to the one on the downlink (see Figure III.17). The bitwise-IDVC field is not used here since the remote does not have the knowledge of the IDVCs assigned to other remotes (if any) within the connection. The PAR field is used to piggyback requests for additional allocation of information slots [46]. The remote that is currently transmitting a cell may set the PAR field to any value that would benefit the channel-allocation scheme in force; otherwise, the PAR field is set to zero. In other words, only cells arriving at a remote's empty queue will cause contention on the uplink control subchannel.

E. PERFORMANCE ISSUES

A variety of parameters, issues, and scenarios in the mobile system may affect the performance of the MAC protocol. Some of these are:

- Traffic characterization: Nature of individual class, mix of classes, and exact scenario.

- Connections type: Remote to remote, remote to CP, CP to remote, or a mixture of these.
- Load: Division of load between downlink and uplink channels.
- Scheduling scheme: Static per service, static per class, dynamic per node, etc.
- MAC frame size: Static or dynamic.

In this section, we define several quantitative performance measurements for the mobile network in order to be able to evaluate it under different channel-allocation policies and traffic loads. Unlike in a wireline single-queue single-server system, in the mobile network the CP and the remotes usually arrive *together* at the decision regarding the cells to be serviced (distributed allocation algorithm). In addition, the low channel capacity and the relatively-infrequent remote-status updates imply low computational and memory requirements (if the remotes' status are updated in every frame, the capacity becomes even smaller). Thus, the main performance measurements of the network are the admissible region and the normalized channel throughput.

1. Admissible Region

The admissible region (sometimes called the *schedulable region*) determines the number of calls that can be admitted into the network. The channel-allocation scheme affects the size of the admissible region. The more efficient the scheme is, the wider the admissible region, thus more calls can be accepted into the network while still maintaining the QoS constraints for all connections. The merits of using a schedulable region to guarantee the QoS in networks are recognized in [54], leading to the development of several scheduling policies. The admissible region is upper bounded by the mean-rate assignment, and it is guaranteed to include at least the peak-rate allocation [80].

The admissible region measures the efficiency of the channel-allocation algorithm utilized in a single-queue system as well as in a mobile network. The limited channel capacity (compared to wireline-ATM fiber-optic links, for example) demands even stricter requirements on the scheduler to efficiently allocate the channel. The region in the

wireless case consists of the number of stations and sources per station that can be established simultaneously.

Unlike in the wireline case, the possible number of scenarios, each defining the number of nodes, the type and the number of active sources within each node, and the connection types, is tremendously large. Thus, we need to define several *representative* scenarios in the mobile network and measure the admissible regions for those. We allow two degrees of freedom (N_I and N_{II}), representing the number of sources of given traffic class and type within two chosen stations (may be two different service classes within the same node). The admissible regions so obtained are thus two-dimensional surfaces.

2. Channel Throughput and Normalized Channel Throughput

The *mobile-channel throughput*, S_M, measures the rate of traffic (in cells/sec) that passes through the mobile channel successfully (for transmission of traffic from higher layers than the MAC). It is equivalent to the probability that a given information slot is used successfully by exactly one user [1]. Since the wireless channel capacity is fixed, one can also define the *normalized mobile-channel throughput*, $\bar{S}_M$, in the range from zero to one. It is given by the ratio between the channel throughput and the available channel capacity for information transmission:

$$\bar{S}_M = \lim_{t\to\infty} \frac{E\{\lambda\}\times t}{C_M \times t} = \frac{E\{\lambda\}}{C_M},$$

where $E\{\lambda\}$ is the mean number of cells per second that have been successfully received by all nodes in the channel during a period t. C_M denotes the mobile-channel capacity in cells/sec, which is the maximum theoretical value of $E\{\lambda\}$. The capacity of the proposed mobile network equals the number of information slots per second that may be utilized by the MAC, which is $22/0.012 \approx 1833$ cells/sec. The preamble and guard bits are excluded from the calculation since they are physical-layer overheads.

An important factor that affects the performance of a MAC protocol is the channel load, G, related to the number of cells transmitted by all stations in the wireless

channel. Maximum channel throughput may be obtained when the load is greater than or equal to the channel capacity ($G \geq C_M$), given that G is asymptotically constant.

Like the size of the admissible region, the throughput is also a measurement of the efficiency of the MAC protocol. An increase in throughput indicates better use of the radio channel for transmission of integrated services information via effective channel allocation. In doing so, the scheduler must fulfill the QoS requirements for *all* the active sources in the mobile network. The throughput is expected to reach extremes on the boundary of the admissible region, where maximum possible sources are admitted. As long as the system operates within the schedulable region, the required QoS is met for all sources. Thus, at a given point *in* the admissible region, a lower bound for the normalized channel throughput would be

$$\bar{S}_M \geq \frac{\text{Transmitted Cell Rate}}{\text{Channel Capacity}} = \frac{\sum_{i \in \{S_S\}} E\{\lambda_i\} \times (1 - CLP[i])}{C_M}, \qquad \text{(III.1)}$$

where $\{S_S\}$ is the set of active sources within the network, $E\{\lambda_i\}$ the mean rate of cells successfully received from source i, and $CLP[i]$ the allowed loss for i. Equation (III.1) forms a lower bound because the performance for each source $i \in \{S_S\}$ can be just better than that "promised" to the source at call setup. Since the required cell loss probabilities for our sources are rather small (at most 10^{-3} for speech, see Section IV.B), we can approximate the normalized throughput with an accuracy of 0.1% or better by

$$\bar{S}_M \approx \frac{\sum_{i \in \{S_S\}} E\{\lambda_i\}}{C_M}.$$

The throughput calculation ignores the physical-layer overhead. The MAC *utilization* of the mobile network, ρ_M, takes this into account. The utilization measures "how well the MAC uses the available channel capacity (C_M) for transfer of upper-layer information." Based on the MAC structure previously described, the maximum utilization of the MAC for the default subchannel sizes, as $\bar{S}_M$ approaches one, is given by

$$\rho_M\big|_{\max} = 1 - \frac{(40+8+160+4\times 60+6)+(8\times 1.9\times 60)+(11\times(40+8+6))}{12{,}000} = 83.67\%,$$

where the load caused by the ATM signaling messages is assumed negligible in comparison with the user data.

F. SUMMARY

The MAC protocol performs the tasks of channel allocation and signaling support. In the past two decades, researchers investigated much effort developing protocols for multiple-access networks, where the objectives were to achieve maximum throughput or minimum delay (e.g., [12]). Mobile integrated services networks have introduced the concept of QoS within the channel-allocation process; the objective of the MAC is now to maximize the throughout while satisfying the service constraints of all active connections in the wireless network.

In this chapter, we have proposed a MAC protocol for the mobile B-ISDN. The protocol segments the time into contiguous frames, each of which includes downlink (CP-to-remotes) and uplink (remotes-to-CP) channels. These channels are further divided into control and information subchannels. The proposed scheme requires remote reservations prior to slot allocations by the CP for cell transmissions. Together with a piggyback mechanism, where future allocation is reported in the cell header, the scheme guarantees a very small number of collisions in the contention-based uplink control subchannel. This is essential to support the multimedia sources having distinct QoS requirements. Once the infrastructure of the MAC protocol has been established, the rest of this work is dedicated to obtain an efficient channel-allocation algorithm over the wireless channel. The measurements of efficiency would be the size of the admissible region and the normalized channel throughput.

IV. SOURCE MODELING

This chapter discusses modeling of speech, video, and data traffic. Section A details existing models, and in Section B we propose low-bit-rate models for these traffic classes, especially applicable for outdoor wireless integrated services networks.

A. EXISTING TRAFFIC MODELS

Models for voice, video, and variable-bit-rate data available in the literature are presented in this section.

1. Voice

Several models for human voice have been proposed in the last thirty years. The basic one, called the on-off model, consists of a 2-state Markov model in which the speaker can be in a silent (inactive) or in a talkspurt (active) state as shown in Figure IV.1 [80]. The states are assumed to be exponentially distributed, and a mean time of 0.65 second for the inactive state and a mean time of 0.352 second for the active state are commonly used [36]. In the active state, cells are transmitted at a constant rate, forming a Markov-modulated deterministic process. For an active rate of 64 kbps, information is transmitted at a rate of R = 170 ATM cells/sec (using 47 voice samples per cell as the appropriate AAL requires) or R = 62.5 packets/sec (using 128-sample packets).

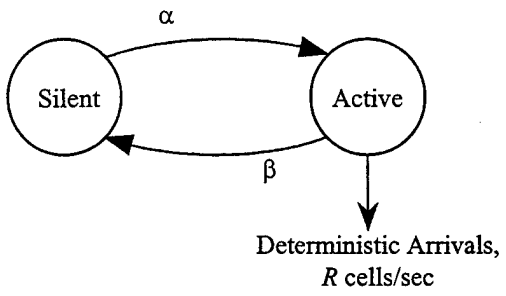


Figure IV.1: On-Off Model of a Voice Source [80]

Another scheme called the Markov-modulated Poisson process (MMPP) (see Figure IV.2), although not accurate, provides a convenient formulation for mathematical analysis. The MMPP model differs from the on-off model by its behavior during talkspurt; rather than generation of R packets per second at a constant rate, the voice packets are generated randomly, obeying a Poisson process with average rate of R cells (packets) per second [80]. Several MMPP sources that are multiplexed together give rise to an aggregate MMPP source as well. Typical values to be used for mean-silent and mean-talkspurt periods are $1/\alpha = 1.65$ seconds and $1/\beta = 1.35$ seconds, respectively [20], which result in a voice source having an *activity factor* of $\alpha/(\alpha+\beta) = 0.45$.

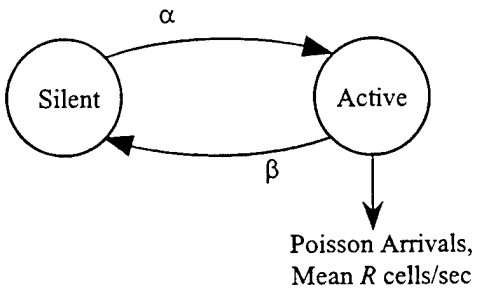


Figure IV.2: MMPP Model of a Voice Source [80]

On the basis that there are not only principal spurts and gaps related to talking, pausing, and listening in voice, but also mini talkspurts and mini gaps due to short silent intervals that punctuate continuous speech, an improved model is proposed by Goodman, Nanda, and others [33] [63]. This model, shown in Figure IV.3, uses slow and fast speech activity factors to capture the two types of silence, where subscripts P and M indicate principal gap and mini gap, respectively. Symbols α and α_M represent the transition rates from principal- and mini-silent states, respectively, to the active state. The mean number of mini talkspurts in each principal talkspurt is given by n; thus, $1/n$ is the probability that a mini talkspurt is the final one in its principal talkspurt. The state sojourn times are exponentially distributed with the following common values, resulting in an activity factor of 0.36: a mean principal-talkspurt duration of 1.00 second, a mean principal-

silence duration of 1.35 seconds, a mean mini-talkspurt duration of 0.275 second, and a mean mini-silence duration of 0.05 second.

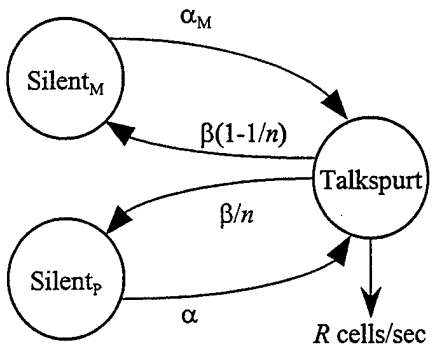


Figure IV.3: Voice Model Including Mini Gaps and Mini Talkspurts

The models mentioned so far represent independent speakers, i.e., two-way conversation characteristics are not taken into consideration. The two-way communication actually reduces the cell generation rate compared to two independent speakers [32]. Based on measurements of 32 conversations involving various speakers, Brady [16] developed a Markovian model for a two-way voice conversation between persons A and B as shown in Figure IV.4. Parameters α_i, β_i (i = 1, 2, 3) define the exponential rates at which one person stays in the inactive and talkspurt states, respectively, while parameters γ_i, δ_i are the corresponding rates for the other person. Measured values of these parameters (for 16 male speakers) appear in Table IV.1.

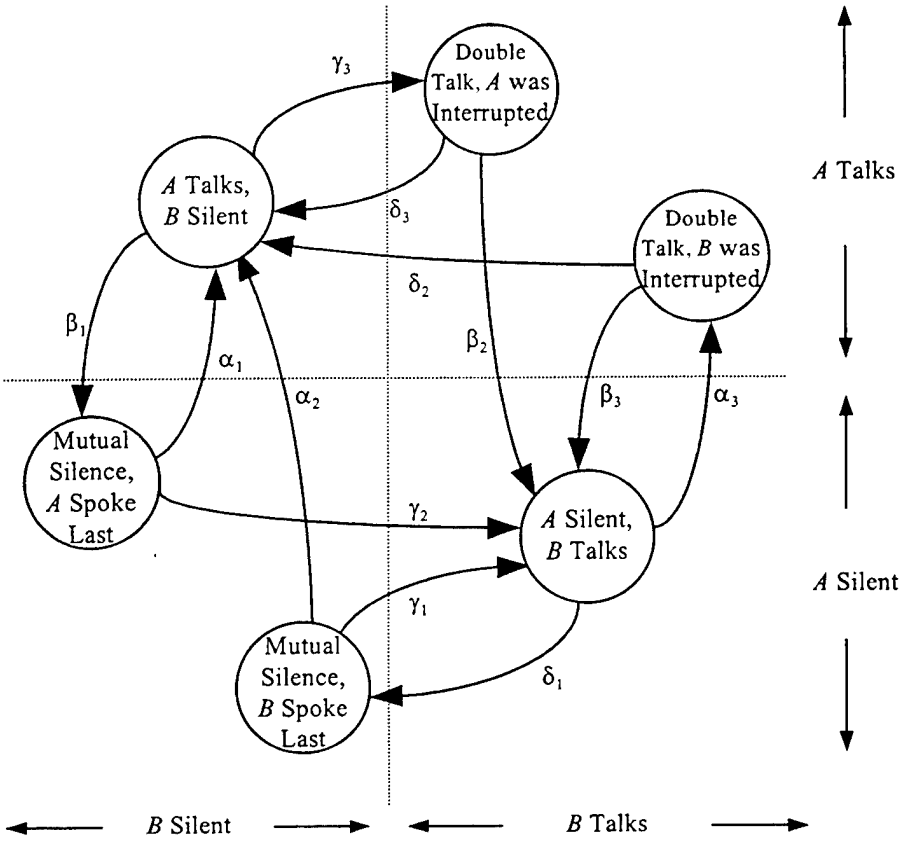


Figure IV.4: Markovian Model of a Two-Way Conversation [16]

Rates	Range (sec^{-1})	Mean ($second^{-1}$)	Std. Dev. ($second^{-1}$)
α_1, γ_1	1.26-3.23	2.04	0.63
α_2, γ_2	0.46-2.74	1.05	0.60
α_3, γ_3	0.09-0.69	0.25	0.15
β_1, δ_1	0.22-1.14	0.65	0.22
β_2, δ_2	1.11-6.01	2.30	1.34

Table IV.1: Exponential Rates of a Two-Way Conversation Model [16]

For the case where the two speakers have the *same* speech characteristics, with mean rates as listed in Table IV.1, the steady-state probabilities, $\pi = [\pi_0, \pi_1, \ldots, \pi_5]$, are calculated using

$$\pi \times Q = \mathbf{0} \text{ and } \sum_{1 \le i \le 6} \pi_i = 1, \tag{IV.1}$$

where Q is the *infinitesimal generating matrix* of the underlying Markov chain, and the bolded zero represents a vector of zeroes. The elements of Q represent the transition rates between states, with row elements summing to zero. Solving Equation (IV.1), the steady-state probabilities are then given by

$$P\{\text{Mutual silence}, A \text{ spoke last}\} = P\{\text{Mutual silence}, B \text{ spoke last}\} \cong 0.08$$
$$P\{A \text{ talks}, B \text{ silent}\} = P\{A \text{ silent}, B \text{ talks}\} \cong 0.4$$
$$P\{\text{Double talk}, A \text{ was interrupted}\} = P\{\text{Double talk}, B \text{ was interrupted}\} \cong 0.02.$$

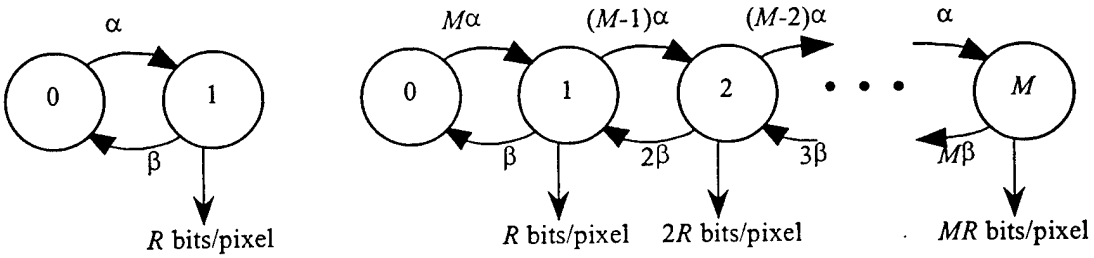
2. Variable-Bit-Rate Video

There does not seem to be any consensus in the literature on a model for generating video traffic, particularly for cell-based networks [84]. The cell arrival process depends on a variety of factors, such as the type of application and the expected picture quality, the amount of movement and the rate at which scene changes occur, and the type of coding technique used (DPCM, predictive, hierarchical, etc.) [74].

Maglaris *et al.* [60] used a first-order autoregressive model to accurately model the average bit rate and the source autocorrelation function. The output of this model has a Gaussian distribution and an exponentially-decaying autocorrelation function. However, the model is entirely based on short measurements of the head of a talking person, thus is not general enough and does not capture the effect of scene changes in a video sequence. Additionally, it cannot be used in analyses of ATM buffers and multiplexers.

Sen *et al.* developed a correlated Markov-modulated model [82]. This model, which can be applied in queueing analysis, assumes that the multiplexed output of N video sources can be represented by summing the outputs of $M >> N$ minisources, each modeled by an identical two-state Markov chain. The Markov models of the minisource

and the video source are shown in Figure IV.5. The equivalent (M+1)-state Markov-chain source model is a quantized version of the original source; the time-varying bit rate has been quantized to the values R, $2R$, …, MR bits/pixel only, where R is a constant.



(a) Minisource Model **(b) Equivalent (M+1)-State Markov Chain**

Figure IV.5: Multiple Minisources Model of a Video Source [82]

Reduction of the effect of quantization is achieved by increasing M to 20 or more times N. This model can fit the average source rate and approximate the autocorrelation with an exponentially-decaying function. Doing so, one obtains the following equations to calculate the values of α, β, and R [60]:

$$\beta = 3.9 \Big/ \left(1 + 5.04458 \frac{N}{M}\right)$$

$$\alpha = 3.9 - \beta$$

$$R = 0.1 + 0.52 \frac{N}{M}.$$

For M/N = 20, the values β = 3.12, α = 0.78, and R = 0.13 are obtained. A drawback of the model is that its bit rate follows a binomial distribution and cannot be used to fit an arbitrary histogram of a video source [84].

Ramamurthy and Sengupta [74] considered the empirical data obtained with a VBR video codec by Verbiest and Pinnoo [95]. They developed a different type of model that can be applied to a variety of video applications, ranging from low-bit-rate video telephony to full motion video with large movements and scene changes. The model is based on the sum of two autoregressive processes, X_n and Y_n, and a process, Z_n, which determines the additional bit rate generated during scene changes. The two autoregressive

processes discussed in [74] are required to fit the quick drop-off rate for short lag times and the slow drop-off rate for larger lag times that are found in video signals. Defining T_n as the bit rate in the n^{th} frame (n = 1, 2, ...), modeled as a stationary random process, the model suggests that

$$T_n = X_n + Y_n + Z_n$$
$$X_n = c_1 X_{n-1} + A_n$$
$$Y_n = c_2 Y_{n-1} + B_n$$
$$Z_n = K_n C_n,$$

where $A_n \sim N(\mu_1, \sigma_1^2)$, $B_n \sim N(\mu_2, \sigma_2^2)$, and $C_n \sim N(\alpha/2, \beta^2/4)$ are independent random processes and c_1 and c_2 are constants.

A three-state, discrete, Markov-modulated process, K_n, shown in Figure IV.6, is used to generate bursts due to scene changes at random intervals. Usually, the chain is at State 0. A transition (with probability p) to State 2 models a scene change. The next transition to State 1 propagates the effect of the scene change and makes it last for two frames. In the first frame after a scene change (K_n = 2), the extra bits generated have mean α and standard deviation β. In the next frame (K_{n+1} = 1), they become $\alpha/2$ and $\beta/2$, respectively. The time between scene changes is two frames plus a geometric random variable with mean $1/p$. The model's statistics between bursts are determined by the autoregressive processes and do not change with time (stationary); thus, the model is limited since it assumes that the statistical behavior of all scenes is identical.

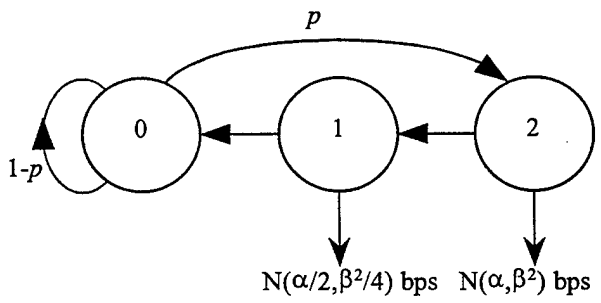


Figure IV.6: Three-State Markov Chain (K_n) Modeling Scene Changes

While the previously discussed models use exponential autocorrelation functions, Huang [37] obtained different results. He considered 30 independent video sources, 15 seconds long each, with various activity levels. The frames contained 477×490 pixels (256 levels each) and were divided into subframes consisting of 9 lines each (totally 477/9 = 53 subframes in one frame). Based on this data, it was found that, on the average, only 35% of the pixels changed values from frame to frame. Moreover, most of the motions usually occurred at the center of the video scenes while the edges of the picture were often background with much less activity. The autocorrelation function of the video sources did not obey the exponential behavior but rather exhibited a pseudo-periodic property (at multiples of 53 subframes) [37, Figures 2, 3, 4]. Further research by Joseph *et al.* [43] found that the bit-rate statistics of video sources were not even stationary.

Skelly *et al.* [84] modeled video traffic as a Markov-modulated rate process. Their model was aimed at providing a suitable approximation for the histogram of a video sequence recorded from a segment of the "Star Wars" movie (NTSC quality). The proposed Markov chain can capture the histogram of a video source quite accurately; it can also capture the relative transitions from a given state to all other states. The histogram uses only eight different bit rates (*bins*), thus the Markov chain consists of eight states. (While increasing the resolution of the histogram did not dramatically change the approximation, it was shown that less than eight bins resulted in a poor approximation.) The 8-state continuous-time Markov chain with selected transitions is illustrated in Figure IV.7.

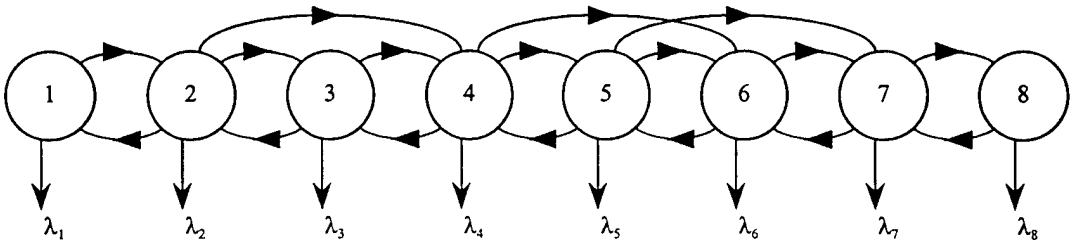


Figure IV.7: Eight-State MMPP Model of a Video Source; Selected Transitions [84]

Since the frame period is much larger than the cell service time, the state-transition probability matrix, P, is transformed into the infinitesimal generating matrix, Q, for a continuous-time Markov chain using

$$Q = f \times (P - I),$$

where f (taken in [84] as 24) is the frame rate of the video source and I the identity matrix. The transition rates, the constant state arrival rates (λ_i, $1 \leq i \leq 8$), and the steady-state probabilities (π_i) appear in Tables IV.2 and IV.3.

To State / From State	1	2	3	4	5	6	7	8
1	-4.737	4.105	0	0.316	0	0.316	0	0
2	0.643	-1.714	0.771	0.043	0.129	0.043	0.086	0
3	0	1.408	-2.254	0.845	0	0	0	0
4	0	0.082	0.740	-1.562	0.616	0.082	0	0.041
5	0	0	0.111	0.741	-1.333	0.407	0.074	0
6	0	0	0	0.095	1.518	-3.130	1.423	0.095
7	0	0	0	0	0.137	2.606	-2.743	0
8	0	0	0	0	0.189	0	0.189	-0.378

Table IV.2: Infinitesimal Generating Matrix, Q [84]

State i	λ_i (bits/frame)	π_i
1	140,000	0.025
2	170,000	0.190
3	200,000	0.145
4	230,000	0.210
5	260,000	0.240
6	290,000	0.090
7	320,000	0.060
8	350,000	0.040

Table IV.3: Histogram Rates and Steady-State Probabilities [84]

The histogram model is well suited for correlated sources (where most transitions occur between neighboring states), but it turns out that scene changes do not significantly

affect its accuracy. In live broadcast, where the picture statistics are unknown prior to the transmission, one needs to estimate the histogram transition rates and arrival rates *a priori*.

3. Variable-Bit-Rate Data

Until recently, the commonly used model for bursty data traffic was the on-off Poisson process, usually with a high degree of burstiness (defined as the peak-rate to mean-rate ratio). This model was introduced earlier in the context of voice traffic models. For data traffic, discrepancies from empirical data could not be explained by this model. Leland *et al.* [56] conducted a four-year study of Ethernet-traffic measurements at Bellcore. The LAN used in the study served researchers and engineering designers; the traffic mainly consisted of Internet protocol packets. The traffic tended to be self-similar (fractal-like) on scales of a few milliseconds to several hours. At every time scale, bursts consisted of high-burst regions separated by low-burst regions. This behavior is drastically different from that of the Poisson model, which produced traffic that is indistinguishable from white noise after aggregating over a few hundred milliseconds. The Ethernet traffic exhibited an increasing degree of self-similarity as the utilization of the network increased. Also, as the number of sources was increased, the traffic became more bursty while the traffic produced by the Poisson model becomes smoother under similar conditions.

The findings in [56] led to further work in the area of self-similar models for ATM networks. Paxson and Floyd [68] supported these findings by a study of the wide-area-network traffic consisting of Internet protocol packets measured on the Internet for over a year. The study offers the following observations:

- For small data transfers, such as in TELNET (virtual remote terminal protocol) or file transfer protocol (FTP) connection setup, the Poisson process is well suited to model the traffic.

- For large data transfers, such as FTP data, transport-control-protocol traffic, or email, the arrival model is better modeled by a self-similar approach.

- Overall, the wide-area-network traffic appears to be more bursty than that generated by a Poisson model.

An interesting outcome of these observations is that the concept of average arrival rate in ATM networks might no longer hold: cell blocking can occur in fixed-size queues even for a mean arrival rate of zero.

The following section contains a survey on several models proposed in the literature for data traffic. Appendix B introduces self-similar random processes, their properties, and suitable mathematical models representing them.

a. Data Models

Measurements performed on ISDN D-channel suggest decomposition of traffic into three types: active key strokes, irregular activity, and machine produced packets [94]. Based on this, the following fractal arrival process is proposed in [94] for ATM networks using a single parameter Pareto distribution:

$$F_X(x) = P\{X \le x\} = 1 - (x+1)^{-D}, \qquad D \in (0,1),$$

where $F_X(x)$ is the probability distribution function of a random variable, X.

Subramanian and Le-Ngoc [88] suggested a switched Poisson process, which is a mathematically-convenient aggregate model for ATM traffic. Data traffic is modeled as alternate periods of short bursts and long bursts, in which the Poisson process switches in between (see Figure IV.8). Self-similar heavy-tail distribution is modeled by the Pareto distribution, defined as

$$P\{X \ge x\} = x^{-\alpha}, \qquad \alpha > 0,$$

where if $1 < \alpha < 2$, the Pareto is self-similar with Hurst parameter $H = (3-\alpha)/2$. This model proves to be self-similar for short- and long-time ranges. It should be observed that the Poisson arrival rates should be "matched" with the aggregate traffic stream; this issue was not addressed in [88].

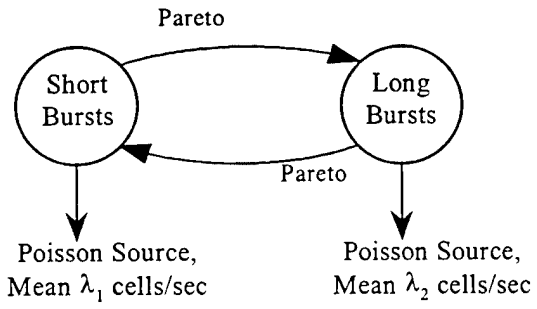


Figure IV.8: Switched Poisson Process with Heavy-Tail State Distribution [88]

Likahnov *et al.* [58] investigated the effect of aggregating M on-off sources ($M >> 1$) to develop an overall, simple, mathematically-tractable, self-similar model. They proposed the model shown in Figure IV.9 for an individual source. At any time, the source can be in one of two states, active or idle. Cells are generated only in the active state, at a constant rate R. The time spent in the active state is heavy-tail Pareto distributed with parameter $\alpha \in (1,2)$ such that $H = (3-\alpha)/2$. The time in the idle state has a generic distribution, $f_\Theta(\theta)$. Both distributions are required to have finite means. By aggregating many such sources while the length of the active period and the aggregate intensity are unchanged, the aggregate process is asymptotically self-similar with parameter H. As $M \rightarrow \infty$, the number of active sources appearing in a given time period is Poisson, thus the system can be observed as an $M/G/1$ queue with Poisson arrival process and Pareto service times.

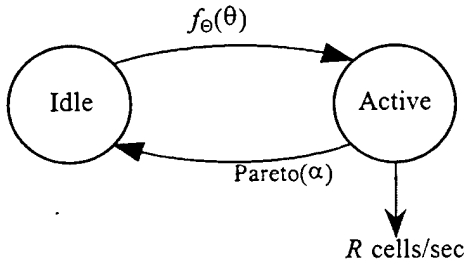


Figure IV.9: Self-Similar Process as an Aggregation of M On-Off Sources ($M >> 1$) [58]

Kiessler *et al.* [48] argued that the MMPPs do not exhibit the long-range dependence that is prominent in fractal processes. Nevertheless, superposition of MMPPs with distinct convergence rates can produce a locally self-similar behavior that is dominant at different resolutions while retaining the MMPP characteristics. The authors proposed a hybrid scheme for self-similar traffic, which is a mixture of distributions having an exponentially-converging covariance with a Pareto distribution. The Pareto distribution is defined here with two parameters as

$$P\{X \geq x\} = 1 - F_X(x) = \left(1 + \frac{x}{\alpha}\right)^{\rho},$$

where $1 < \rho < 2$. Figure IV.10 illustrates an example of the hybrid model.

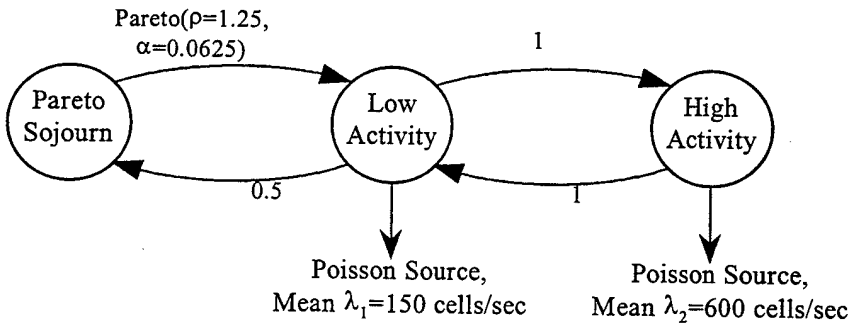


Figure IV.10: Hybrid Model for Self-Similar Traffic [48]

B. TRAFFIC MODELS FOR LOW-BIT-RATE ATM NETWORKS

In this section, we propose low-bit-rate models for bi-directional speech conversation, real-time variable-bit-rate video, and bursty data sources in wireless integrated services networks. The QoS parameters required to complete the definitions of the models are also described.

1. Bi-Directional Speech-Conversation Model

Here we propose a simpler model for a bi-directional voice conversation based on Brady's model [16] discussed in Section A.1. For simplicity, we use the same parameter values for all speakers as given in Table IV.1.

We represent a conversation using a birth-death Markov process that contains three states (0, 1, 2), where state i represents i active speakers as shown in Figure IV.11. The data rates within States 1 and 2 are deterministic at R_S and $2R_S$ cells/sec, respectively. The state probabilities of the model are derived from Brady's model as follows:

$$\pi_0 = P\{\text{Mutual silence}, A \text{ spoke last}\} + P\{\text{Mutual silence}, B \text{ spoke last}\} \cong 0.16$$
$$\pi_1 = P\{A \text{ talks}, B \text{ silent}\} + P\{A \text{ silent}, B \text{ talks}\} \cong 0.8$$
$$\pi_2 = P\{\text{Double talk}, A \text{ was interrupted}\} + P\{\text{Double talk}, B \text{ was interrupted}\} \cong 0.04.$$

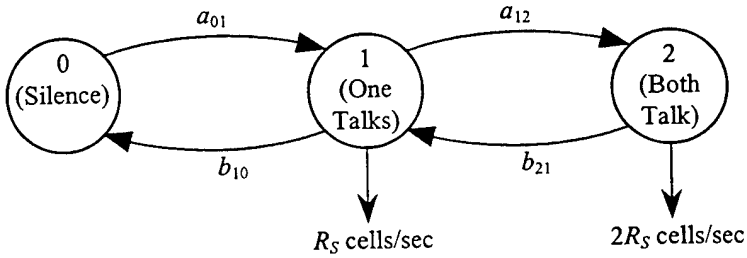


Figure IV.11: Three-State, Birth-Death, Voice-Conversation Model

Assuming that the speaker characteristics are similar, we obtain the state-transition rates of Figure IV.11 by summing the mean values of the outgoing and incoming rates shown in Figure IV.4:

$$a_{01} = \alpha_1 + \alpha_2 = 3.09$$
$$b_{10} = \beta_1 = 0.65$$
$$a_{12} = \gamma_3 = 0.25$$
$$b_{21} = \delta_2 + \delta_3 = 4.72.$$

The mean arrival rate, $E\{\lambda\}$, of a single conversation is given by

$$E\{\lambda\}\big|_{1\ Source} = \sum_{i=0}^{2} \pi_i \times (i \times R_S) = 0.8 \times R_S + 0.04 \times 2R_S = 0.88 \times R_S \text{ cells/sec.}$$

For an active speaker rate of 32 kbps, R_S is 85.1 cells/sec (47 speech samples per cell), and the corresponding mean arrival rate is 74.9 cells/sec.

We now expand the model to the case of N_S conversations. The resulting composite model is a Markov process, comprising $2N_S+1$ states (0, 1, 2, ..., $2N_S$). Finding the aggregate steady-state probabilities, given that N_S individual conversations are multiplexed, requires the use of Kronecker product [27]. The number of states then grows exponentially; however, the arrival rates in the individual model grow linearly with state. Consequently, one can obtain the $2N_S+1$ steady-state probabilities of the aggregate model by *convolving* the individual state probabilities [65]. Generally, the convolution of two steady-state probability sets $\{\pi_A: (\pi_{A,0}, \pi_{A,1}, \ldots, \pi_{A,n-1})\}$ and $\{\pi_B: (\pi_{B,0}, \pi_{B,1}, \ldots, \pi_{B,m-1})]$ results in a steady-state probability set $\{\pi_C: (\pi_{C,0}, \pi_{C,1}, \ldots, \pi_{C,n+m-2})\}$:

$$\pi_C = \pi_A * \pi_B \quad \rightarrow \quad \pi_{C,l} = \sum_{k=0}^{n} \pi_{A,k} \times \pi_{B,l-k}, \qquad 0 \le l \le n+m-2,$$

where $*$ denotes the convolution operator. The resulting $2N_S+1$ aggregate-source arrival rates are then given by $\{0, R_S, 2R_S, \ldots, 2N_S R_S\}$. The mean cell arrival rate of the aggregate source equals

$$E\{\lambda\}\big|_{N_S \text{ Sources}} = \sum_{i=0}^{2N_S} \pi_i \times (i \times R_S) \text{ cells/sec.}$$

For example, the aggregate Markov chain for $N_S = 2$ is shown in Figure IV.12.

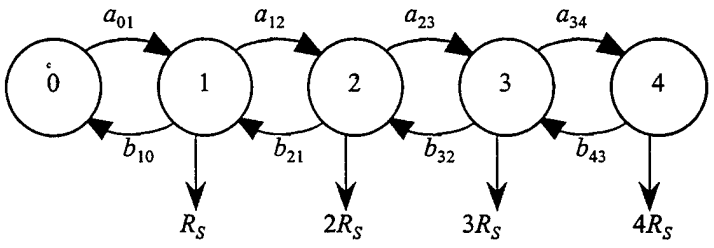


Figure IV.12: Aggregate Model for Two Conversations ($N_S = 2$)

The steady-state probabilities are obtained by convolving the steady-state probabilities of the individual chains:

$$\pi\big|_{2\text{ Sources}} = [\pi_0\delta(i)+\pi_1\delta(i-1)+\pi_2\delta(i-2)] * [\pi_0\delta(i)+\pi_1\delta(i-1)+\pi_2\delta(i-2)]$$
$$= {\pi_0}^2\delta(i)+2\pi_0\pi_1\delta(i-1)+\left({\pi_1}^2+2\pi_0\pi_2\right)\delta(i-2)+2\pi_1\pi_2\delta(i-3)+{\pi_2}^2\delta(i-4)$$
$$\pi_0\big|_{2\text{ Sources}} = {\pi_0}^2 = 0.0277$$
$$\pi_1\big|_{2\text{ Sources}} = 2\pi_0\pi_1 = 0.2636$$
$$\pi_2\big|_{2\text{ Sources}} = {\pi_1}^2 + 2\pi_0\pi_2 = 0.6405$$
$$\pi_3\big|_{2\text{ Sources}} = 2\pi_1\pi_2 = 0.0664$$
$$\pi_4\big|_{2\text{ Sources}} = {\pi_2}^2 = 0.0018$$
$$\pi_i\big|_{2\text{ Sources}} = 0, \qquad \forall i<0, i>4,$$

where $\delta(i)$ stands for the delta function at state i, and π_i ($0 \le i \le 2$) are the steady-state probabilities of an individual source.

Finding the rates of the aggregated source analytically is not possible. Normally, one solves for the steady-state probabilities given the state-transition rate diagram. Here, we can find the steady-state probabilities, but not the rates.

2. Real-Time Video Model

We adopt the histogram approach proposed in [84] to model slow-motion low-bit-rate video sources. We use the transition rates and cell arrival rates as given in [84][6] to provide a 64-kbps video source, i.e., approximately 178 cells/sec (using 45 video samples per cell) on the average. The steady-state probabilities, π_i, and the cell arrival rates, λ_i, are given in Table IV.4. Augmentation to an aggregate model for N_V video sources is performed using convolution, similar to the speech case.

[6] The results reported are based on 24 uncompressed frames/sec segments while here we are concerned with compressed frames. Nevertheless, the model is used as a representative example of the histogram approach.

State i	λ_i (cells/sec)	π_i
1	106.00	0.025
2	128.72	0.190
3	151.43	0.145
4	174.14	0.210
5	196.86	0.240
6	219.57	0.090
7	242.29	0.060
8	265.00	0.040

Table IV.4: Histogram Video Model – Rates and Steady-State Probabilities

3. Bursty Data Model

We consider two typical data applications involving transfer of either text or image/graphic information. In the first case, we have relatively-short text files (on the order of a few hundred to a few thousand octets), such as e-mail messages, location messages, and commands. The second case comprises massive amount of data (on the order of a few tens of thousands of octets), e.g., transfer of high-resolution images. These two types of data are also different by the frequency of their occurrence; the former is expected to occur more often than the latter. Although the nature of both sources of data is bursty, their burst length distribution may lead to different QoS requirements for each (for example, smaller allowed delays for the text data).

We propose a *single* model for the data source, which approximates both data applications appropriately. Figure IV.13 presents a hybrid three-state Markov chain in which we have one inactive state and two active states: low activity (for text) and high activity (for imagery). The steady-state probabilities of this model are found to be 0.9616, 0.0288, and 0.0096 for the inactive, low-activity, and high-activity states, respectively. The cell arrival rates in the active states, R_D, are equal (e.g., R_D = 100 cells/sec), and cells are generated at these states deterministically. The source enters the low-activity state 207.5 ($\cong$0.0288×2×3600) times per hour on average and produces short bursts (average length of 50 cells). It moves to the high-activity state 3.5 ($\cong$0.0096×0.1×3600) times per

hour and generates larger bursts (mean length of 1,000 cells). The total mean arrival rate is 3.85 cells/sec, which makes the source relatively bursty (burstiness factor of over 25). We assume the existence of an access-control mechanism that limits the peak rate of the data source to R_D cells/sec (e.g., using a traffic shaper) in both active states.

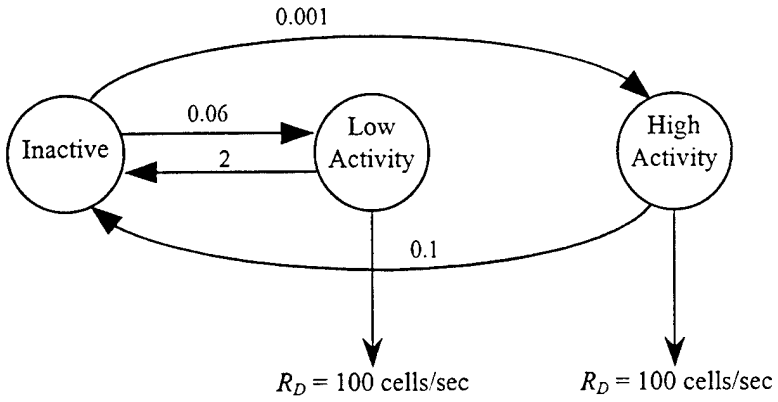


Figure IV.13: Hybrid Model of a Bursty Data Source

Expansion of the model to N_D data sources is possible too. Since the arrival rates at the low- and high-activity states are the same (do not grow linearly with the states), we cannot use the convolution operator to obtain the state probabilities of the aggregate source. On close examination, the aggregate process has only N_D+1 distinct arrival rates $(0, R_D, \ldots, N_D R_D)$, equally spaced by R_D cells/sec from each other. The aggregate (steady-state) probability, π_i^A $(0 \le i \le N_D)$, of being in state i is given by

$$\pi_i^A = \binom{N_D}{N_D - i} \times \pi_0^{N_D - i} \times \sum_{j=0}^{i} \binom{i}{j} \times \pi_1^{j} \times \pi_2^{i-j}, \qquad \text{(IV.2)}$$

where π_0, π_1, and π_2 are the steady-state probabilities corresponding to inactive, low-activity, and high-activity states, respectively, of an *individual* data source.

4. QoS Requirements

The source models alone do not fully characterize the source to be supported in a low-bit-rate network. The QoS requirements [9] to be fulfilled by the network must also be defined. Table IV.5 describes the requirements defined for each service class in our

case. We assume that a cell that has not been transmitted within a specified maximum cell transfer delay (maxCTD) is dropped. The dropped cells are the origin of the cell loss probability (CLP) for transmission over ideal error-free links. (Due to low arrival-rate to service-rate ratios, buffers along all paths in the system are considered to be of infinite length.) The cell delay variation (CDV) parameter is unspecified. It is assumed that a receiving station contains a mechanism to handle the variations in cell arrivals for real-time sources [100]. The requirements (for a macrocell environment) listed in Table IV.5 are somewhat more relaxed than those determined for similar sources in wireline as well as microcell wireless ATM networks [11].

QoS Requirement	Service Class		
	Speech	Video	Data
CLP	10^{-3}	5×10^{-5}	10^{-6}
maxCTD (milliseconds)	40	100	30000
CDV	Unspecified	Unspecified	Unspecified

Table IV.5: QoS Requirements for the Proposed Traffic Services

5. Summary

This chapter presented models for low-bit-rate sources applicable to low-capacity channels, such as a multiple-access outdoor (macrocell) wireless network. The proposed speech model utilizes the properties of a two-directional conversation. A histogram-based model having at least eight bins, which was proven in the past to accurately represent a variable-bit-rate, real-time, video stream, is adapted to model a 64-kbps video source. A hybrid model for data sources has been proposed as well. It represents the bursty nature of two typical multimedia applications: frequent text transfers and less frequent image transmissions. Expansion from a single source to the case of multiple sources is discussed for all the traffic classes.

V. SCHEDULING IN WIRELINE INTEGRATED SERVICES NETWORKS

In this chapter, we study the access point of a wireline integrated services network, represented as a single-queue single-server system. The concept of an access point has been introduced earlier in Section II.A. The goals in this chapter are to discuss existing scheduling techniques for the wireline case and to propose novel schemes that outperform those described in the literature. The input traffic streams to the scheduler are of type speech, video, and data, which follow the models addressed in Section IV.B.

Section A begins with a brief introduction to the wireline queueing system and an overview of related work in the area of scheduling. In Section B, we consider performance-related aspects of the schedulers, such as utilization and run time. In Section C, we determine the (minimum) required capacity of a server that multiplexes multiple homogeneous sources such that their QoS requirements are maintained. We denote this type of scheduling as *static allocation* since a guaranteed portion of the channel link is reserved for each source. The shortest-time-to-extinction (STE) algorithm is discussed in Section D. We thoroughly address, in Sections E and F, other novel, more efficient, scheduling algorithms, namely the balanced cell-loss-probability ratio (BCLPR) and the STE with BCLPR (STEBR). A proof of optimality of STEBR algorithm is included in Section G. In Section H, we present a simplification of STEBR that reduces the run time. Finally, Section I presents performance evaluation of the various algorithms based upon simulation results.

A. SCHEDULING AT AN ATM NETWORK NODE

An ATM network node, which can be an access point (e.g., multiplexer) or an output port of a switch, is represented as a single-queue single-server system [59]. In single-queue single-server systems, a single server having a fixed capacity multiplexes multiple sources entering a common queue. The server allocates the available bandwidth among the sources, according to some scheduling scheme such that the QoS agreed upon is maintained for all sources. Scheduling techniques discussed in the literature include the

classical round robin and its variants [47], approaches based on finishing times (epochs in which the cells must leave the buffer to fulfill the maxCTD requirement), and priority-based methods. After a brief introduction of the system model, an overview of early work on the topic of scheduling is presented. Special attention is given to STE, which will be used later for channel allocation in the wireless medium.

1. System Model

The system being considered consists of a single queue and a single server as shown in Figure V.1. ATM cells generated by multiple (n) sources of various traffic classes, some of which are of the same class, arrive at the queue. We refer to this set of n active sources as $\{S_S\}$. It is assumed that an arriving cell is placed at the tail end of the queue, although queue reordering by the server is possible. The fixed-size ATM cells require a constant service time for each cell. As a result, the system can be thought of as being slotted in time such that the duration of each slot is equal to the transmission time of one ATM cell. The symbol C_W (in cells/sec) is used to denote the wireline capacity of the ATM link under study. Every $1/C_W$ seconds (slot), the server scans the queue and (if it is not empty) services one cell according to its scheduling policy. If the queue is empty at the beginning of a slot, the server remains idle until the beginning of the next slot, even if cells arrive in the meanwhile.[7] The server is assumed to be non-preemptive, i.e., it cannot stop the service of a cell before the end of the slot. Each arriving cell must be transmitted completely by its deadline, or equivalently, service must begin $1/C_W$ seconds prior to the deadline. If this condition is not met, the cell is considered lost and removed from the buffer. This model is particularly appropriate for a single, shared, communication link and its associated buffer in an asynchronous, time-division, integrated-network setting [18].

---

[7] This is referred to as a *non-work-conserving* server [49], meaning the server can remain idle for a portion of a slot even if cells are present in the buffer.

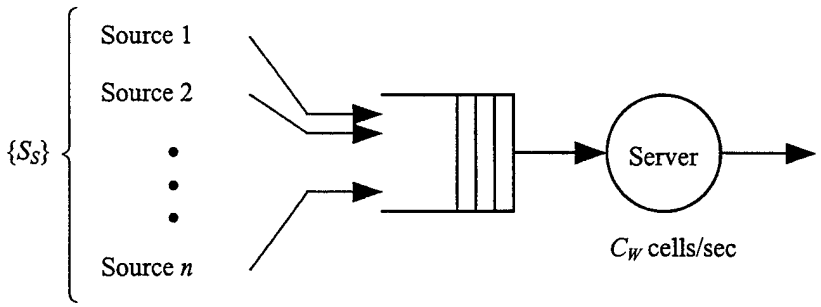


Figure V.1: Single-Queue Single-Server System Model

2. Early Work

Several established scheduling algorithms have received wide attention in the literature. Due to its simplicity, many current networks use the FCFS queue discipline, which applies the same QoS requirements (and hence the same service characteristics) to all cells. In FCFS, cells are served in the order of their arrival, completely ignoring the different allowed loss by the sources. It is inefficient, especially when the allowed QoS values are spread over a few orders of magnitude.

Static-priority-scheme (SPS) algorithms are used to allow simple differentiation of cells based on pre-assigned priorities. With SPS, cells from a given source are given priority over other cells in the queue. The server always services the cells having the highest priority in the queue, where cells having the same priority are serviced in a FCFS manner. The assignment of various priority levels allows the flexibility to vary the level of service given to sources based on their performance requirements. This scheme is class dependent and simple to implement. It may be inefficient, though, especially when high-priority cells have much longer deadlines than low-priority cells since it ignores variation of cell deadlines.

A variant of SPS, considered by many researchers as an appropriate solution for supporting traffic with deadlines, is the bandwidth reservation [85]. In this scheme, after negotiation with the server, sources are guaranteed bandwidth allocations in proportion to their allowed timeout. Given n input sources, source i ($1 \leq i \leq n$) having a deadline

(maxCTD) of T_i is guaranteed a capacity of $\left(T_i / \sum_{j=1}^{n} T_j\right) C_W$. This simplifies the implementation but may lead to low server utilization.

In minimum-laxity-threshold scheduling, a pre-defined hierarchy of priorities are assigned to different traffic classes [38]. Laxity of a cell from the highest-priority class is the amount of time the server may remain idle or service cells of other classes and still be able to service this cell by its deadline. Laxity of a cell from the second highest-priority class reflects the fact that highest-priority cells having deadlines before it would be serviced ahead of it. The rest of the laxities are obtained similarly. Once all the laxities have been calculated, the server selects the first cell from the highest-priority class that has not expired and whose laxity is less than the service time. If none exists, the scheduler looks for a cell from the second-highest-priority class using the same criterion and so on. As a result, the highest-priority cells are likely to be serviced but delayed just within their deadline bounds. Cells from lower-priority classes would meet their deadlines unless higher-priority cells are of sufficient quantity to require an inordinate capacity. A drawback with this scheme is the inflexibility due to assigned priorities, which are fixed throughout the connections' lifetime. The prioritized structure of the minimum laxity threshold scheme ignores the allowed loss permitted by each source thus does not fully utilize the capacity to admit more calls.

To handle variable-size packets with deadlines, the earliest-deadline-first algorithm was proposed by Jackson [42]. Deadlines are assigned to packets before they enter the queue, and a queued packet having the smallest deadline is serviced (even if its deadline has expired). The algorithm was proven to minimize the maximum lateness (defined as the finish time minus the deadline time) [42] and the expected lateness [87]. A variant of earliest deadline first for fixed-size packets (cells), drops cells that have missed their deadline. This scheme is called shortest time to extinction (STE) and is optimal with respect to the loss rate in discrete-time *G/D/*1 and continuous-time *M/D/*1 queueing systems [66]. STE (and earliest deadline first) are simple to implement, but they are not optimal for heterogeneous traffic streams having different loss and delay

requirements; the loss of cells from one source (application) is generally more important than that of others.

Some other algorithms advocate polling-based approaches [31] [67]. Time is divided into cycles of flexible length, containing several slots each. With such an approach, for example, cells from a given source are given a higher priority over others if the number of such cells transmitted in the current cycle has not exceeded a fixed threshold. This approach ensures fairness among sources; however, there is no way to dynamically set the thresholds.

A more sophisticated scheduling algorithm called MARS was proposed to run on top of a polling-based algorithm [38]. A schedule consisting of fractions of cycles dedicated to each traffic class in future cycles is maintained, and the schedule is modified continually based on the arrival streams. MARS assumes that all sources can be divided into three classes, each having very specific requirements (e.g., an upper bound on the average number of consecutively lost cells or a minimum on the average source throughput). The algorithm is effective for traffic meeting its exact assumptions regarding performance objectives, but relaxing these assumptions is difficult [71].

Another approach is occupancy-based scheduling, such as queue length threshold [18]. Scheduling decisions depend on the number of cells queued from each class; whenever the number exceeds a threshold value, the corresponding cells are given priority over others. Chipalkatti *et al.* [18] compared the relative performance of FCFS, SPS, minimum laxity threshold, and queue length threshold; the latter scheme was found to give the best overall tradeoff between performance and complexity. Unfortunately, traffic bursts from less delay-sensitive sources result in extended periods during which no delay-sensitive cells are transmitted [71].

In [69], the authors considered a general scheduling problem in which each arriving cell has an arbitrary deadline and weight. A non-real-time scheduling algorithm that minimizes the overall loss rate of the traffic was proposed; however, it requires complete knowledge (past and future) of the cell arrival processes. The algorithm is

meant, therefore, only for comparison with other algorithms and is impractical for real implementations.

Peha and Tobagi [70] [71] proposed cost-based scheduling. In their approach, arbitrary performance objectives of different sources (such as loss rate or mean delay) are expressed as cost functions, which map the queueing delay experienced by each cell to cost incurred. The algorithm then attempts to minimize the total cost incurred by all cells. A cell is assigned a priority according to the estimated cost that will be saved by transmitting the cell rather than keeping it in the queue. If a cell arrives into an empty queue, the scheduling algorithm is invoked (work-conserving system), and the cell with the highest priority is selected for transmission. The algorithm was shown to outperform other known techniques; however, it is difficult to implement in high-speed networks.

B. PERFORMANCE ANALYSIS OF SCHEDULING ALGORITHMS

In this section, we wish to discuss the principles to be used when analyzing a scheduling algorithm. The main aim of a scheduler in an integrated services network is to maximize the admissible region. As the number of calls admitted into the network is increased, the (fixed-capacity) channel is utilized more efficiently. In dealing with single-queue single-server systems, another useful measurement of efficiency is the fraction of the time in which the server does not remain idle. A desirable scheduling algorithm would aim to increase this factor to the maximum possible value. The discussion here covers also the time complexity of the algorithm and the memory (space) required for its operation.

1. Admissible Region

The admissible region determines the boundary of working points of the system. In this work, three traffic classes are considered, thus resulting in a three-dimensional admissible region. A point within this volume represents the number of speech, video, and data sources that enter the queue (see Figure II.16). Given a fixed-capacity channel, a more efficient scheduling algorithm leads to a larger volume of the admissible region. Sometimes differences in performance among different schedulers may be better

demonstrated using two-dimensional cuts. This is achieved by setting the number of sources from one traffic class to a constant and observing the resulting region as a function of the number of sources of the remaining two traffic classes.

2. Server Throughput and Normalized Server Throughput

In classical queueing theory, the normalized throughput factor in a queueing system is defined as the ratio of the rate at which "work" enters the system to the maximum rate (capacity) at which the system can perform this work [49]. The work an arriving cell brings into the system equals the service time it requires (i.e., $1/C_W$). The throughput of the system measures the actual rate at which cells are transmitted out of it. The (wireline) throughput and normalized throughput are denoted by S_W and $\overline{S}_W$, respectively. In a single-queue single-server system that is part of a wireline ATM network, all input sources are guaranteed their maximum loss probability. Therefore, other than the allowed portion of dropped cells, all the cell arrivals into the queue are serviced by the server on time, and the normalized throughput is bounded as follows:

$$\frac{\sum_{i\in\{S_S\}} E\{\lambda_i\}\times(1-CLP[i])}{C_W} \le \overline{S}_W \le \frac{\sum_{i\in\{S_S\}} E\{\lambda_i\}}{C_W}, \qquad \text{(V.1)}$$

where $E\{\lambda_i\}$ and $CLP[i]$ are the mean arrival rate and the cell loss probability of source i, respectively. Approximating $\overline{S}_W$ by its upper bound (the right term in Equation (V.1)), provides an accuracy of 0.1% or better since the maximal allowed loss among the traffic classes is 10^{-3}.

From Equation (V.1), the normalized throughput is proportional to the arrival rate of the traffic. Thus, the throughput is expected to reach maximum values on the boundary of the admissible region, where maximum number of traffic sources are allowed into the queue. For the trivial case of zero sources, the throughput, of course, equals zero.

3. Algorithm Time Complexity

When comparing scheduling techniques based on the resources required for proper operation, the most critical aspect is the run time or time complexity. A scheduling

algorithm, implemented in an ATM switch (usually in hardware), must operate in real time and make up to C_W decisions per second. Comparison of run times usually assumes a generic single processor having random-access-machine model of computation [19]. While the execution time of each primitive operation (step) in an algorithm is assumed constant, the number of steps is a distinguishing factor among different algorithms. Here, the most significant parameter is the number of cells, N, in the buffer when the algorithm needs to make a scheduling decision at the beginning of a slot. It is desirable to use an algorithm that runs in at most linear time (number of steps that linearly depends on N). The scheduling algorithm scans the queue in order to select a cell for service that meets the scheduling criterion. Thus, the run time depends on the number of times the queue is scanned. The algorithms are analyzed under worst-case scenarios to obtain upper bounds on their run-time performance.

4. Memory Requirements

Memory needed by the algorithm for proper operation is another required resource parameter, though less stringent than time complexity. With advances in the VLSI technology, memory is becoming inexpensive. Nevertheless, in large switches that service thousands or millions of inputs simultaneously, it still is a significant factor in switch design. The required memory is analyzed in terms of the number of cells in the buffer, N, or the number of active sources, $|\{S_S\}|$, for the worst case.

C. STATIC ALLOCATION IN A HOMOGENEOUS QUEUE

This section presents a novel scheme to determine the minimum required capacity of the server, to satisfy both loss and delay requirements in homogeneous queues. Alternatively, given the server capacity, one can use the scheme to obtain the maximum number of homogeneous sources that could be admitted by the server into the network.

1. System Model and Terminology

We consider a queueing system as shown in Figure V.2. An aggregate arrival process results from multiplexing several individual sources *of the same type* (such as

those discussed earlier in Section IV.B) with known parameters. This process fills a FCFS buffer with cells. The buffer is serviced by a server with a capacity of C_T cells/sec, where T stands for the service type: S (speech), V (video), or D (data). The (random) variable x is used to denote the instantaneous length of the buffer; K is the buffer size.

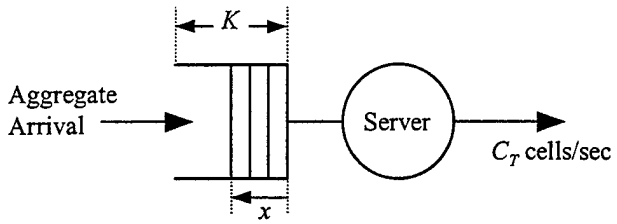


Figure V.2: Model of a Homogeneous Queueing System

The goal here is to find the minimum value of C_T such that both CLP_T and $maxCTD_T$ requirements are met for all multiplexed sources. The following procedure is proposed to achieve that. Suppose that N_T sources of type T are multiplexed, and the server capacity, C_T, is known. To satisfy the delay requirement, we limit the buffer size to

$$K = \lfloor C_T \times maxCTD_T - 1 \rfloor;$$

the first term is the upper bound on the waiting time *in the buffer* that a cell can experience before it is dropped while -1 reflects the service (transmission) delay. A case in which K = -1 is not of interest because it means that the allowed delay is less than the service time. Next, we analyze the CLP in a finite-size buffer that can occur due to cell blocking only (the loss due to delay is already incorporated in the value of K). The CLP in the finite buffer can be approximated by the *tail* probability that the occupancy, x, exceeds K for an infinite buffer case. The value of C_T is obtained iteratively. For a given C_T, the required CLP_T must be tightly bounded from below by the calculated loss; otherwise, C_T needs to be increased or decreased appropriately.

2. Cell and Burst Regions

The loss-probability performance curve of a statistical multiplexer, obtained by plotting the CLP as a function of the multiplexer buffer size, is divided into two separate

regions [80]. The region of small buffer sizes is termed the "cell region," and the region of large buffer sizes is called the "burst region." For small buffer sizes, the CLP is found to drop rapidly as the buffer size increases. At some point, depending on the utilization and traffic characteristics, the CLP begins to decrease more slowly as the buffer size continues to increase.

3. Cell Loss in the Cell Region

In the cell region, transitions between states are assumed to be relatively infrequent; equivalently, the maximum transition rate in the Markov chain is very small compared to the service rate.[8] This implies that the time between state transitions is long compared to the period between consecutive arrivals. It is assumed that the buffer is small enough for all states within the source to reach equilibrium. The buffer occupancy distribution and hence the loss probability are obtained by focusing on an individual source at a given time and ignoring the transition rates between states.

Once in state i that has a deterministic arrival rate of λ_i, the queueing system can be treated as a $D/D/1/K$ queueing system [83]. The loss probability of such a system is then given by

$$CLP[i]\big|_{D/D/1/K} = \begin{cases} 1 - \dfrac{1}{\rho_W[i]}, & \text{if } \rho_W[i] > 1 \\ 0, & \text{otherwise,} \end{cases} \tag{V.2}$$

where $\rho_W[i]$ denotes the (wireline) queueing system utilization in bin i and is given by

$$\rho_W[i] = \frac{\lambda_i}{C_T}.$$

The number of states having distinct arrival rates (bins) in the aggregate source is referred to as B_T, where T marks the type of the source. Speech and video sources may be considered as histograms consisting of B_S and B_V bins, respectively, indexed as integer

[8] In [80], a ratio of 10^{-4} is mentioned for the assumption to be valid. Here, due to low-bit-rate sources, this assumption is not justified; that is, the system reaches steady-state within every state only after residing for a certain, non-negligible amount of time in the state. Hence, our empirical results for loss probability are expected to be a bit smaller than the theoretical results based on [83].

values starting from zero. These histograms are characterized by a constant difference between arrival rates of neighboring states. Each bin has its own steady-state probability, π_i, and deterministic arrival rate, λ_i. The aggregate probabilities of the speech and video sources are obtained by convolution(s) of the individual source probabilities [27] (see Section IV.B). Individual data sources have $B_D = 3$ states while their aggregate process has N_D+1 distinct arrival rates; the aggregate (steady-state) probabilities are obtained using Equation (IV.2).

In this model, queueing loss occurs only in the higher states (overload states) for which the deterministic arrival rate is greater than the buffer service rate. The number of cells lost in state i over a long period T_∞ is $\pi_i \times \lambda_i \times T_\infty \times CLP[i]$ while the number of cells arrived during this period is $E\{\lambda\} \times T_\infty$. The CLP in the cell region is the ratio between the number of cells lost in all the overload states and number of cell arrivals, over a long period. Using the bin loss from Equation (V.2), we get

$$CLP\big|_{CELL} = \frac{1}{E\{\lambda\}} \times \sum_{\substack{i=0 \\ \lambda_i > C_T}}^{B_T-1} \left(1 - \frac{1}{\rho_W[i]}\right) \times \pi_i \times \lambda_i = \frac{1}{E\{\lambda\}} \times \sum_{\substack{i=0 \\ \lambda_i > C_T}}^{B_T-1} (\lambda_i - C_T) \times \pi_i . \qquad \text{(V.3)}$$

4. Cell Loss in the Burst Region

For large buffer sizes, the assumption that each histogram bin reaches steady state is no longer valid. The histogram steady-state model, neglecting state transitions, assumes that the loss probability will level off at a constant value, independent of the buffer size. The state transitions must now be taken into account, leading to continued reduction in loss probability as the buffer size increases rather than leveling off. This represents the onset of the burst region, where we use fluid-flow analysis [5].

From *large-deviation* theory, it can be shown [5] with considerable generality that the tail probability, $P\{x > K\}$, is asymptotically exponential:

$$\lim_{K \to \infty} P\{x > K\} = A \times e^{\delta K}, \qquad \text{(V.4)}$$

where δ is the negative slope of the burst region equal to the *dominant eigenvalue* of the system. The dominant eigenvalue has the smallest absolute value among the eigenvalues of the matrix MD^{-1}, where M is the transition-rate matrix of an individual source Markov

chain, and $D = \text{diag}(\lambda_i - C_T)$. The diagonal element $D_{ii} = \lambda_i - C_T$ of matrix D represents the drift or rate of change in the buffer content when the source is in state i; hence, D is called the *drift* matrix. The tail probability of Equation (V.4) approximates the loss probability of a large finite-size system (using fluid-flow approximation [5]). It is found to be accurate, especially for small utilization values. To determine the asymptotic constant A in Equation (V.4), we use the hybrid technique proposed in [83].

5. Hybrid Model

Recall that we were able to predict the loss in the cell region using the histogram model. An ad-hoc technique proposed in [83] allows joining of the curves of the cell and burst regions. This is achieved by comparing the slopes of the loss probabilities in the two regions and obtaining the cutoff point, K_0; K_0 is the maximum buffer size at which the slope of the cell region is still steeper than the slope of the burst region[9] [80]:

$$K_0 = \max K : \quad \left| \frac{d}{dK} \log\left[CLP(K)\big|_{CELL} \right] \right| \geq -\delta.$$

The loss probability in the burst region, as a function of the buffer size K, is calculated by Equation (V.4), where A is given by

$$A = CLP(K_0)\big|_{CELL} \times e^{-\delta K_0},$$

and the resulting loss in the burst region is equated as follows:

$$CLP(K)\big|_{BURST} = CLP(K_0)\big|_{CELL} \times e^{\delta(K-K_0)} = \frac{1}{E(\lambda)} \times \sum_{\substack{i=0 \\ \lambda_i > C_T}}^{B_T - 1} (\lambda_i - C_T) \times \pi_i \times e^{\delta(K-K_0)}.$$

When multiplexing N_T homogeneous sources, δ can be found directly by solving for the slope of a single source in isolation, with the capacity decreased by a factor of N_T (i.e., taken as C_T/N_T) [83]. This means that, for a given utilization, the dominant eigenvalue (the slope in the burst region) is the same regardless of the number of sources being multiplexed.

---

[9] Generally, the loss probability in the cell region is a function of the service discipline and dependent on K. Deterministic arrival model represents only a *single* point at $K_0 = 0$ on the loss-probability plot, as a function of K.

6. Summary of Method to Obtain Minimum Required Capacity

Given that N_T homogeneous sources are multiplexed in a buffer, the algorithm summarized in Figure V.3 is used to obtain the minimum required capacity to service them. The symbols in general refer to individual sources; superscript A is used to indicate the aggregate source. Matlab code for implementing the procedure is listed in Appendix C.

1. **Set** the channel capacity C_T to an arbitrary value.
2. **Calculate** the maximum buffer length:
$$K = \lfloor C_T \times maxCTD_T - 1 \rfloor.$$
3. **Calculate** the mean arrival rate of an individual source:
$$E\{\lambda\} = \sum_{i=0}^{B_T-1} \pi_i \times \lambda_i.$$
4. **Calculate** the aggregate steady-state probabilities:
$\pi^A = \pi * \pi * \cdots * \pi$ (N_T–1 convolutions), for speech/video sources
$$\pi_i^A = \binom{N_D}{N_D - i} \times \pi_0^{N_D - i} \times \sum_{j=0}^{i} \binom{i}{j} \times \pi_1^{j} \times \pi_2^{i-j}, \quad 0 \le i \le N_D, \quad \text{for data sources.}$$
5. **Calculate** the number of bins of the aggregate source:
$$B_T^A = \begin{cases} 3 \times N_S - 1, & \text{for speech sources} \\ 8 \times N_V - 1, & \text{for video sources} \\ N_D + 1, & \text{for data sources.} \end{cases}$$
6. **Calculate** the aggregate arrival rates in states $0 \le i \le B_T^A - 1$:
$$\lambda_i^A = \begin{cases} i \times R_S, \quad R_S = 74.9, & \text{for speech sources} \\ N_V \times \lambda_0 + i \times \Delta_V, \quad \Delta_V = \lambda_1 - \lambda_0, & \text{for video sources} \\ i \times R_D, \quad R_D = 100, & \text{for data sources.} \end{cases}$$
7. **Calculate** the aggregate mean arrival rate:
$$E\{\lambda^A\} = \sum_{i=0}^{B_T^A - 1} \pi_i^A \times \lambda_i^A.$$
8. **Calculate** the CLP in the cell region:
$$CLP\big|_{CELL} = \frac{1}{E\{\lambda^A\}} \times \sum_{\substack{i=0 \\ \lambda_i^A > C_T}}^{B_T^A - 1} (\lambda_i^A - C_T) \times \pi_i^A.$$
9. **Calculate** the drift matrix:
$$D = \mathrm{diag}(\lambda_i - C_T / N_T).$$
10. **Calculate** δ, the dominant eigenvalue of MD^{-1}.
11. **Calculate** the loss probability in the burst region for the maximum buffer size:
$$CLP(K)\big|_{BURST} = CLP\big|_{CELL} \times e^{\delta K}.$$
12. **If** the loss probability obtained in the burst region is greater/smaller than CLP_T:
    A. **Increase/decrease** C_T accordingly.
    B. **Repeat** Steps 2-11.

Figure V.3: Calculation of the Required Capacity in a Homogeneous Queue

7. Simulation Results

Figure V.4 presents plots of CLPs for the cases of 1, 5, 10 and 20 multiplexed speech conversations. The *x*-axis represents the required link capacity per multiplexed source. Simulation results follow the analytical results closely. Of interest is the case in which the buffer size is smaller than or equal to $K = \lfloor C_S \times maxCTD_S - 1 \rfloor$. To satisfy the QoS requirement of CLP_S, a minimum capacity as given by the intersection of the CLP plot (solid) and the allowed CLP_S (dashed horizontal line at a loss rate of 10^{-3}) is required. For example, the capacity required for five speech conversations is between 98 (based on simulation results) and 102 (theoretical calculation) cells/sec versus a peak rate of 170.2 cells/sec.

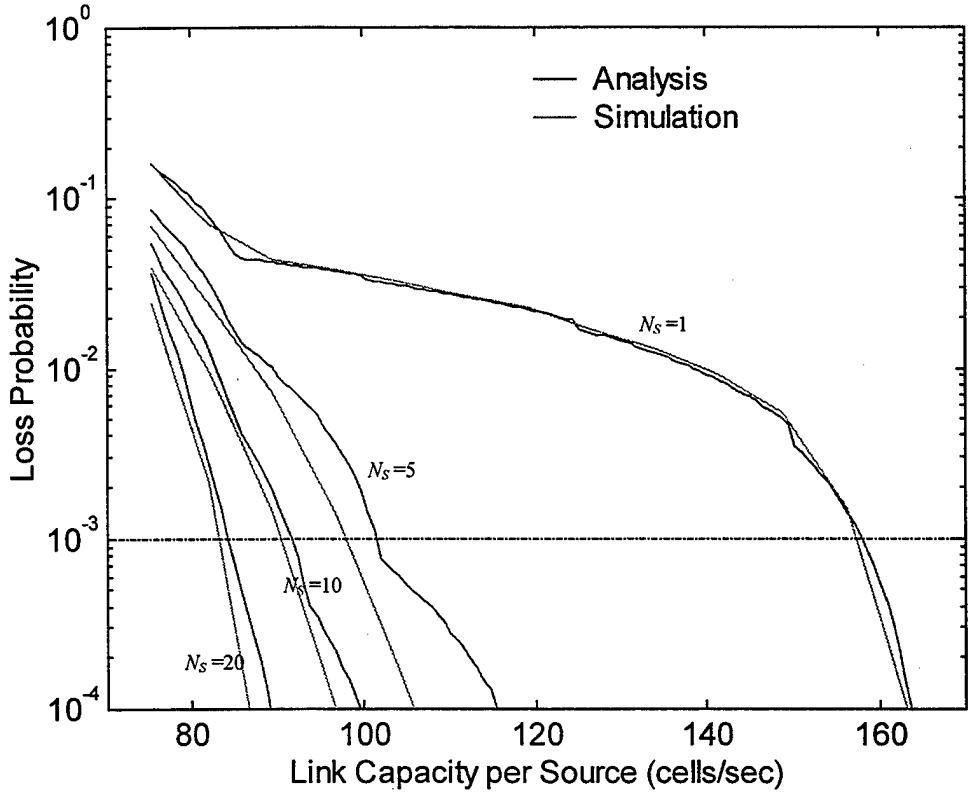


Figure V.4: Loss Probability for N_S Multiplexed Speech Sources

Results for N_V video sources and N_D data sources are presented in Figures V.5 and V.6, respectively. For data sources, $maxCTD_D$ has been taken to be 1 second (not 30 seconds as defined in Table IV.5) to allow convergence of simulation runs in a reasonable time.

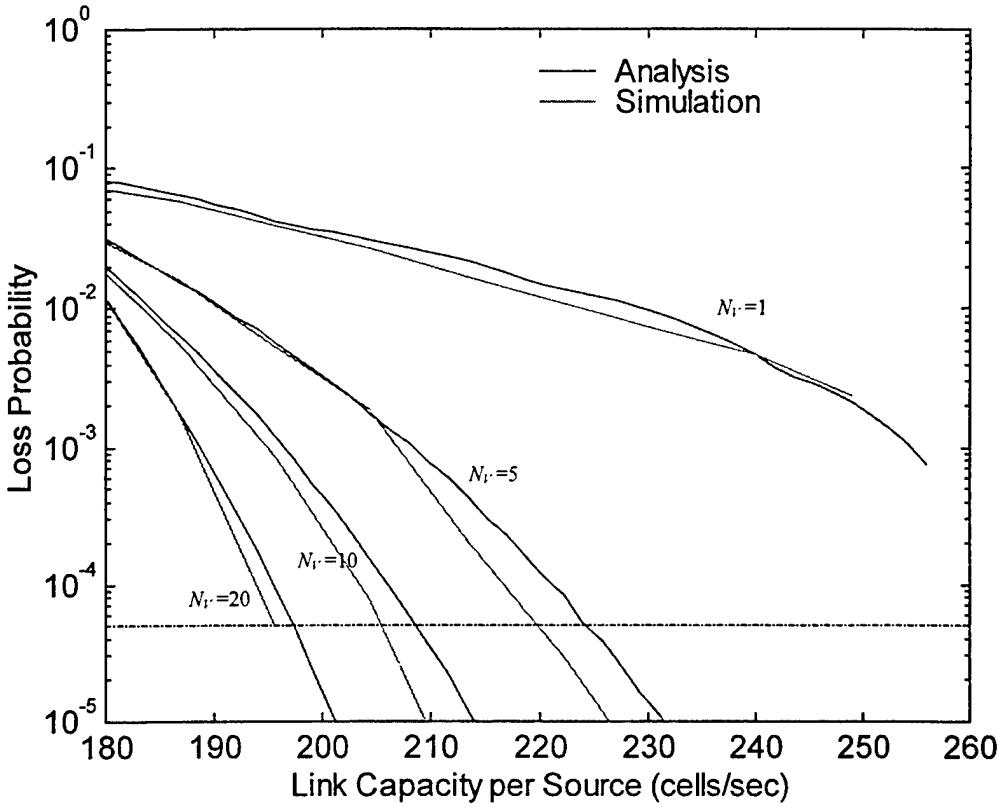


Figure V.5: Loss Probability for N_V Multiplexed Video Sources

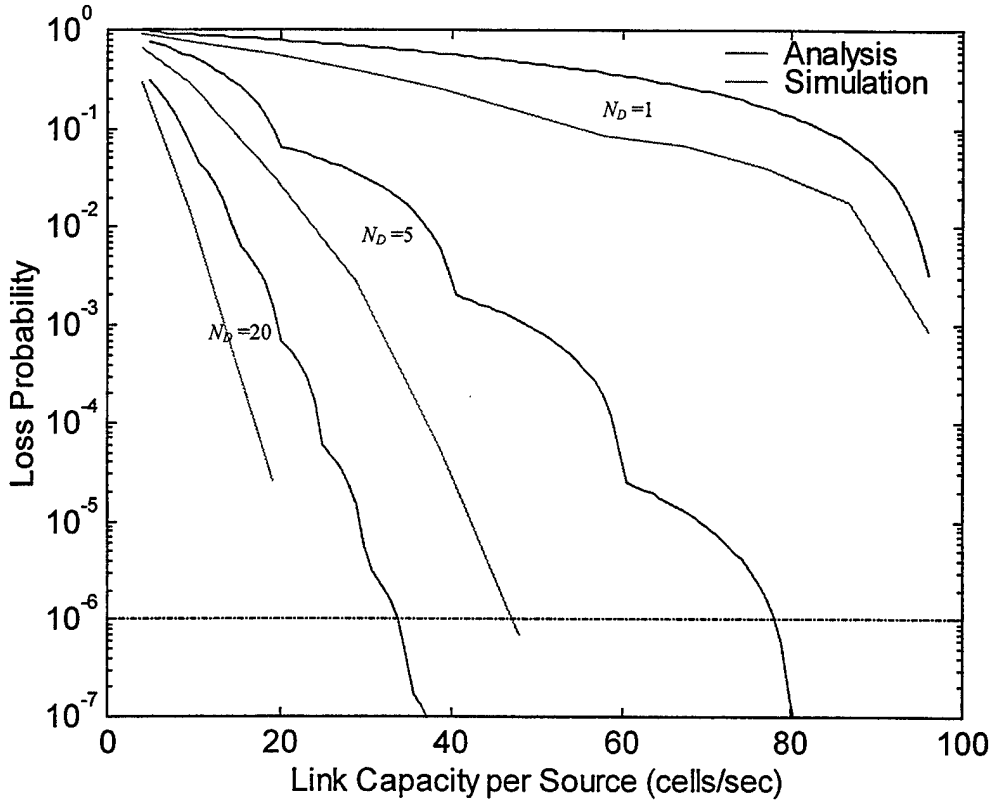


Figure V.6: Loss Probability for N_D Multiplexed Data Sources ($maxCTD_D$ = 1 Second)

8. Static-Allocation Scheme

Once the required capacities for multiplexed homogeneous sources have been obtained, the next step is to study the allocation mechanism. Here we discuss the static-allocation technique.

a. Database

The static-allocation scheme (and generally any other scheduling algorithm) requires a database to store the variables, arrays, lists, etc. that are essential for proper operation. The following arrays comprise the database maintained by the static-allocation scheme:

- *Required_Capacity*[i]: Array; entry $i \in \{S_S\}$ contains the required capacity by source i, where $\{S_S\}$ is the set of active sources multiplexed in the queue. It

takes into account the number of sources from the same class that are multiplexed in the buffer.

- *Deficit_Capacity*[i]: Array; entry $i \in \{S_S\}$ contains the number of slots that must be allocated to source i.

b. Algorithm

The arrays are initialized to zero. Every time a cell from source i arrives at the queue, the cell is enqueued at the tail of the queue. In every service slot, the algorithm summarized in Figure V.7 is carried out.

1. **Set** *source_to_service* = -1 (invalid source identifier).
2. **Set** *maximum_capacity_so_far* = $-\infty$.
3. **For every** cell in the queue:
   A. **Calculate** the time of expiry (ToE) of the cell (say, originated by source i).
   B. **If** the ToE of the cell is smaller than $1/C_W$ (the service time):
      i) **Discard** the cell.
   C. **Else**:
      i) **Calculate**:
      Deficit_Capacity[i] = *Deficit_Capacity*[i] + (*Required_Capacity*[i]/C_W)
      ii) **If** *Deficit_Capacity*[i] > *maximum_capacity_so_far*:
         a) **Set** *source_to_service* = i.
         b) **Set** *maximum_capacity_so_far* = *Deficit_Capacity*[i].
4. **Service** the first cell from source identified as *source_to_service*. If no cell from this source exists, service the first queued cell (if exists).
5. **Decrement** *Deficit_Capacity*[i].

Figure V.7: Static Allocation in a Wireline System

The algorithm relies on the fact that the admission controller does not admit more sources into the network than the scheduler is capable of handling. The admission controller recalculates the total required capacity, C_{TR}, for each new call,

taking into account the capacity required by the new source. The source is admitted if the new value of C_{TR} satisfies

$$C_{TR} \leq \sum_{i \in \{S_S\}} Required\_Capacity[i]$$

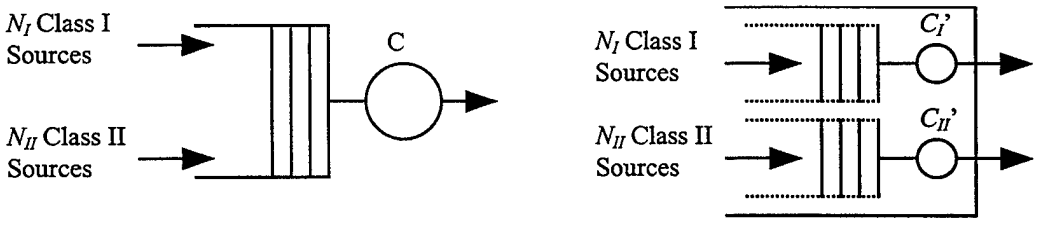
or rejected otherwise.

9. Heterogeneous Sources

Thus far the constant capacity required by the server has been obtained for the homogeneous-sources case only. When sources from different classes are active, the calculation of the capacity required for each class needs to take into account the different QoS requirements of the classes.

Lemma V.1: *Consider multiplexing of N_I sources of Class I and N_{II} sources of Class II into a single-queue single-server system such that each class has its own QoS requirements. Assume that C is the minimum required server capacity such that the QoS requirements for all sources are maintained. If C_I and C_{II} are the minimum capacities required separately to service Classes I and II, respectively, then $C \leq C_I + C_{II}$.*

Proof: Without loss of generality, one can consider a single-queue single-server system as being constituted of two logically-independent queues, one for each traffic class, each requiring server capacity C_I' and C_{II}', respectively (see Figure V.8). The total service capacity is $C = C_I' + C_{II}'$. Since capacities of $C_I' = C_I$ and $C_{II}' = C_{II}$ can satisfy the requirements of all sources of Classes I and II, respectively, it follows that $C = C_I' + C_{II}' = C_I + C_{II}$ is a solution as well.



(a) Single-Queue Single-Server System

(b) Equivalent Double-Queue Double-Server System

Figure V.8: Treatment of a Two-Class-Input, Single-Queue, Single-Server System as a Double-Queue Double-Server System

Now, assume that the scheduling policy of the double-queue double-server system is as follows: if server C_I' does not find any Class I cells in its queue, it uses the current time slot to service cells (if any) from the other queue and vice versa. Under this policy, the loss performance of each of the two queues is *better* (smaller) than required; thus, the minimum capacities required for the queues obey $C_I' \leq C_I$ and $C_{II}' \leq C_{II}$. From here we get that that $C \leq C_I + C_{II}$. ■

A solution to the heterogeneous case is not available in a wireline network at this time. Here, we take the capacity for each class to be the one calculated for the homogeneous case by which we ignore the gain obtained by multiplexing heterogeneous sources. This leads to somewhat conservative results when the minimum required capacity is calculated (i.e., some bandwidth is wasted). However, once a solution to the heterogeneous case becomes available, it can be easily adapted to calculate more precise admissible region and channel throughput for the system.

10. Discussion

The histogram approximation in the cell region, which assumes high service-rate to (source's) Markov-chain transition-rate ratios, has provided satisfactory results. These results were used as a basis for the fluid-flow analysis in the burst region. The analysis, supported by simulation results, provides the ability to fairly estimate the minimum required capacity to satisfy the loss and delay QoS requirements for multiple homogeneous sources. We have observed that the results of analysis match well with those of simulation for speech and video sources. It is believed that the data sources are expected to behave similarly, given "infinitely-long" simulation duration; practically, the simulation run times were limited, causing the high-bursty data sources to slow convergence of results.

The static-allocation algorithm is simple. The queue is scanned only once in order to discard the expired cells and to find the cell having the largest *Deficit_Capacity* to be serviced. Thus, the time complexity is O(N). For every active source, an array of two variables is required; therefore, the required memory is O($|\{S_S\}|$).

There are two main disadvantages of the algorithm. First, the algorithm can only be applied to sources with pre-defined statistics and QoS requirements. Hence, it cannot be used in a network in which admission of new sources follows a negotiation stage between user and network regarding the desired QoS. Second, at the time of this writing, it cannot be expanded to the case of heterogeneous sources. If multiple traffic classes are present in the queue, the technique does not utilize the multiplexing gain of mixing traffic from these classes; the technique only achieves the individual gains obtained for each class, separately.

D. STE SCHEDULING

We now revisit the STE algorithm and present the details of the steps involved. In their paper, Panwar, Towsley and Wolf [66] considered a discrete-time *G/D*/1 queueing system in order to be able to maximize the fraction of cells that begin their service before their respective deadlines. It is equivalent to minimizing the queueing loss. They proposed a class of scheduling policies similar to the one offered by Jackson [42], called shortest time to extinction (STE), which schedules only *eligible* cells with the smallest time of expiry (ToE). In other words, the STE never schedules cells that are past their expiration. The authors proved that the STE policy is optimal for the discrete-time *G/D*/1 queue where the service time is exactly one time unit.

1. Algorithm

The STE algorithm requires no database, other than temporary variables. Every time a cell from source i arrives at the queue, the cell is enqueued at the tail of the queue. Then, at every service slot, the algorithm summarized in Figure V.9 is carried out.

1. **For every** cell in the queue (say, originated by source i):
   A. **Calculate** the ToE of the cell.
   B. **If** the ToE of the cell is smaller than $1/C_W$ (the service time):
      i) **Discard** the cell.
2. **Sort** the remaining cells in the queue in a non-decreasing order of their ToE.
3. **Service** the first cell in the queue.

Figure V.9: STE Allocation in a Wireline System

2. Discussion

The STE algorithm is relatively simple. A single iteration over the queue is sufficient to discard the expired cells and to find the cell having the smallest ToE (i.e., Step 2 of the algorithm may be skipped). Thus, if N denotes the number of cells in the queue, the time complexity is O(N). The memory required by the algorithm is O(1).

The main disadvantage of the algorithm stems from its ignorance of the different allowed loss by the different sources. The STE obtains an optimal solution, only when the assumption that all streams of information have the same QoS requirements is valid.

E. BALANCED-CLP-RATIO (BCLPR) ALGORITHM

In this section, we describe a new scheduling scheme called the balanced cell-loss-probability ratio (BCLPR). Unlike the STE, the BCLPR algorithm makes use of the cell-loss-probability QoS constraints. It uses the deadlines of the cells only for detection and discarding of expired cells rather than for decision making.

1. Concepts

At the beginning of each service slot, the algorithm calculates the instantaneous CLP (ICLP) for each source i that has cells in the queue:

$$ICLP[i] = \frac{DS[i]}{A[i]} = \frac{\text{Discarded cells from source } i}{\text{Arrived cells from source } i}. \qquad \text{(V.5)}$$

Next, the cell-loss-probability ratios (CLPR) of *non-empty* sources (i.e., having at least one cell in the queue) defined as ratios between the ICLP and the allowed cell loss probability (ACLP) of these sources are calculated:

$$CLPR[i] = \frac{ICLP[i]}{ACLP[i]}, \quad i \in \text{Non-empty sources within } \{S_S\}, \tag{V.6}$$

where ACLP is a QoS parameter. The server then processes the first cell from the source having the largest CLPR.

2. Database

The following arrays constitute the database maintained by the algorithm:

- $A[i]$: Array; entry $i \in \{S_S\}$ contains the number of cells arrived thus far from source i.
- $DS[i]$: Array; entry $i \in \{S_S\}$ contains the number of cells discarded thus far by the server from source i.
- $CLPR[i]$: Array; entry $i \in \{S_S\}$ contains the CLPR of source i.

3. Algorithm

All arrays are initialized to zero. Every time a cell from source i arrives at the queue, the cell is enqueued at the tail of the queue and $A[i]$ incremented. Then, at every service slot, the algorithm summarized in Figure V.10 is carried out, where the sources within $\{S_{ES}\}$, the set of sources having non-expired cell(s) in the queue, are denoted as eligible sources.

1. **Set** $\{S_{ES}\} = \{\varnothing\}$.
2. **For every** cell in the queue:
   A. **Calculate** the time of expiry (ToE) of the cell (say, originated by source i).
   B. **If** the ToE of the cell is smaller than $1/C_W$ (the service time):
      i) **Discard** the cell.
      ii) **Increment** $DS[i]$.
   C. **Else if** $i \notin \{S_{ES}\}$:
      i) **Set** $\{S_{ES}\} = \{S_{ES}\} \cup i$.
3. **Calculate** the ICLP for every eligible source $j \in \{S_{ES}\}$ using Equation (V.5).
4. **Calculate** the CLPR for every eligible source $j \in \{S_{ES}\}$ using Equation (V.6).
5. **Service** the first cell in the queue from the eligible source having the largest CLPR. Ties among sources are broken through randomization.

Figure V.10: BCLPR Allocation in a Wireline System

4. Discussion

The BCLPR algorithm has an interesting significance. Suppose that $CLPR[i]$ is greater than one for some source i. This means that the network does not fulfill the QoS requirements for i. If the CLPR is less than one, then the source has a better QoS than has been guaranteed at call setup. The scheduler always chooses to process a cell from the source having the largest CLPR in an attempt to maintain the QoS bounds for that source. An outcome of the algorithm is that the source closest to violating its guaranteed QoS is serviced continuously (so long as it has cells in the queue) until it is no longer closest to violating its QoS agreement. The term *balanced* CLPR comes from the fact that, over a long period, the CLPRs of *all* sources approach the same value.

Although a bit more complex than the STE algorithm, the BCLPR runs in time $O(N)$ as well. Each of Steps 1 through 4 runs in time $O(N)$ and so is the total run time. While scanning the queue (looking for expired cells), the algorithm marks the eligible sources in order to avoid redundant computations of ICLP and CLPR of empty sources. Every active source requires a finite number of database elements; therefore, the total required memory is $O(|\{S_S\}|)$.

F. STE WITH BCLPR (STEBR) SCHEME

In this section, we describe a novel scheme that is a combination of STE and BCLPR schedulers, which we call STEBR. The STE algorithm was proven to be optimal for sources with non-distinct QoS constraints; STE does not consider the different allowed loss rates by different sources. The BCLPR algorithm incorporates distinct cell-loss rates but ignores the advantage of scheduling using the (different) allowed cell delays. We now propose a new scheme that, at service decision time, makes use of both the deadlines of the cells and the loss rate experienced by the sources up to that instant. Decisions at the service slots take into account the deadlines of *all* waiting cells in the queue and the expected loss in the future given that no more arrivals are allowed. Since different traffic classes have different ACLP values, the scheduler tries to minimize a "system cost," in case the constellation of the queued cells guarantees that some cell(s) will expire by the time scheduled for its service. The system cost is a function of individual costs that are assigned to the sources. The cost of a source, associated with its CLPR, is determined by the number of cells arrived and the number of cells discarded from the source.

Given the queue occupancy, waiting cell deadlines, and the history of arrivals and discards, we can prove that this scheme makes the *optimal* decision at every service slot. Consequently, among all scheduling algorithms in the single-queue single-server system, the STEBR has the *largest* admissible region. The algorithm is proven optimal for a causal system in which knowledge about future cell arrivals is not available. We assume that the server makes its decision (on which cell to service) at every service slot based on the cells present in the queue at these times. The server may make better decisions if information about future cell arrivals into the queue is partially or fully available. Alternately, future arrivals can be predicted based on prior knowledge of the sources' statistics. The use of these options, however, is beyond the scope of this work.

We begin by introducing the STEBR algorithm and the underlying concepts. Then, an example that demonstrates its operation is presented. Two versions of the proposed algorithm are presented. The first, described in this section, runs in time $O(N^2)$,

where N denotes the number of cells in the queue at decision time. The second, discussed in Section H, simplifies the first to require time O(N) (*linear time*).

1. Concepts

As introduced above, the STEBR algorithm utilizes two previously discussed techniques: the STE to obtain an optimal scheduling if no cell loss is expected in the queue and the BCLPR to prioritize the sources when loss is expected. The STEBR utilizes the advantages of both techniques using two phases. In the first phase, the algorithm schedules a cell for service at a time when it is closest to expiration except in cases where the server becomes idle. A cell having the smallest ToE that is greater than the service time is serviced first. (This procedure thus "prefers" sources having stricter maxCTD requirements.) In the second phase, in situations in which cells must be discarded (due to the use of the earliest-deadline-first policy), we assign a *cost* to each source. The assigned cost is the source's modified CLPR, given that an additional cell from this source is to be discarded:

$$Cost[i] = CLPR[i]\Big|_{\substack{\text{if additional}\\ \text{cell discarded}}} = \frac{(DS[i]+1)}{A[i]\times ACLP[i]} = CLPR[i] + \frac{1}{A[i]\times ACLP[i]}.$$

Thus, it is not the absolute value of source cost that determines its (instantaneous) priority but the relative value compared to other sources' costs. The best decision would be to service the sources that would 'cost the most' given the set of queued cells and the arrivals and discards thus far. The costs of some sources may increase throughout the operation of the algorithm. The increase of the source costs at decision time (if any) is linear, as will be explained later. An intermediate decision regarding discarding of a cell forces recalculation of the cost (larger value) of its originating source. The new value of the cost may change the relative positioning of the source for service, possibly giving the source that just discarded a cell a higher priority as the algorithm continues its operation.

We now describe the STEBR algorithm using an approach equivalent to the two-phase method detailed earlier. At every service instant, the queue is sorted in a non-decreasing order of the cells' time of expiry as in STE. The scheduler then assigns the cells having the *largest* cost (from oldest to newest) the *latest* service slot possible. Cells

are promoted up the slot assignment, if possible, to allow service of all largest-cost cells on time when more than one cell are contending on the same service slot. Then, the second-largest-cost cells are assigned service slots (again, cells are promoted up the slot assignment, if possible, to allow the service of all largest- and second-largest-cost cells on time). If a cell cannot be scheduled prior to its expiration, the oldest least-cost cell 'loses' its slot assignment, and the cost of its originating source is recomputed assuming the cell is discarded. The process is repeated until no more cells can be scheduled. After the process is completed, the cell chosen for service is the one assigned the first (current) slot. The algorithm does not assign service slots to those cells having the least cost (at the time of discarding), which is the basis for optimality here.

We demonstrate the behavior of the algorithm using Figure V.11, where T_i ($i \geq 0$) marks the beginning of service slot i. At time T_0, the queue contains three cells from Source 1 (1_1, 1_2, 1_3), three from Source 2 (2_1, 2_2, 2_3), and two from Source 3 (3_1, 3_2). Each cell in the queue must be scheduled for service prior to its deadline to beginning of service (DBS) as depicted by the vertical arrows in Figure V.11. The initial values of the source costs as well as the increasing costs as one or two cells from these sources 'lose' a slot assignment are given in Table V.1. The algorithm schedules cells for service, starting with cells having the largest cost (3_1, 3_2), continuing on with the cells of Source 2 (2_1, 2_2 and 2_3), and so on. The order of the queue at each iteration is shown in Table V.2. The cell finally chosen for service on current slot is 2_1.

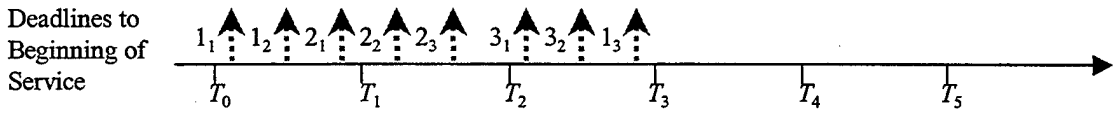


Figure V.11: STEBR Algorithm, Demonstration of Operation under Expected Loss

Source	Source Cost		
	Initial Value	After 'Losing' One Service Slot	After 'Losing' Two Service Slots
1	0.40	0.44	0.48
2	0.43	0.45	0.47
3	0.46	0.50	

Table V.1: Service Costs in the Example Given in Figure V.13

Cell Considered	Scheduled Transmissions					Cell that 'Lost' Service Slot	Source Cost		
	T_0	T_1	T_2	T_3	T_4		1	2	3
3_1			3_1				**0.40**	**0.43**	**0.46**
3_2		3_1	3_2				0.40	0.43	0.46
2_1	2_1	3_1	3_2				0.40	0.43	0.46
2_2	2_1	3_1	3_2			2_2	0.40	**0.45**	0.46
2_3	2_1	3_1	3_2			2_3	0.40	**0.47**	0.46
1_1	2_1	3_1	3_2			1_1	**0.44**	0.47	0.46
1_2	2_1	3_1	3_2			1_2	**0.48**	0.47	0.46
1_3	2_1	3_1	1_3			3_2	0.48	0.47	**0.50**

Table V.2: STEBR Operation Example – Queue Content and Source Costs

2. Linked Lists

The scheduler maintains a database, which includes arrays, linked lists, and variables. We describe some of the properties of a generic *linked list*, necessary to the understanding of the algorithm operation. A linked list contains (an "unlimited" number of) entries, each of which includes a pointer to the next entry in the list, where the pointer of the entry at the tail of the list indicates that it is the last entry. The first entry (the head of the list) is indexed as 0, the second entry as 1, and so on. If at a given time the list contains $n > 0$ entries, it allows the following two operations (if $n = 0$, only the first is allowed):

- *Add*(j, *val*), $0 \le j \le n$: Add an entry with value *val* at position j; entries $j, j+1, \ldots, n-1$ (if any) are demoted one position down the list (assigned indices $j+1, j+2, \ldots, n$, respectively).

- *Delete(j)*, $0 \leq j \leq n\text{-}1$: Delete the entry at position j; entries j+1, j+2, ..., n-1 (if any) are promoted one position up the list (assigned indices j, j+1, ..., n-2, respectively).

3. Database

The following variables constitute the database maintained by the algorithm:

- $A[i]$: Array; entry $i \in \{S_S\}$ contains the number of cells arrived thus far from source i.

- $DS[i]$: Array; entry $i \in \{S_S\}$ contains the number of cells discarded thus far by the server from source i.

- $CLPR[i]$: Array; entry $i \in \{S_S\}$ contains the CLPR of source i.

- $Cost[i]$: Array; entry $i \in \{S_S\}$ contains the value that would be assigned to $CLPR[i]$ if an additional cell from source i is discarded.

- *Service_List*[j]: Linked list; entry j (j = 0, 1, ...) contains the identifier of the source that originated the cell, which is scheduled for service on the j^{th} slot (i.e., T_j). The current slot is T_0 (the 0^{th} service slot).

- *Deadline_List*[j]: Linked list; entry j (j = 0, 1, ...) contains the cell's DBS corresponding to *Service_List*[j].

- *list_index*: Pointer to the position in *Service_List*[] (and *Deadline_List*[] as well) in which its current cell is under test.

4. Algorithm

All database elements are initialized to zero. Every time a cell from source i arrives at the queue, the cell is enqueued at the tail of the queue and $A[i]$ incremented. Then, at every service slot, the algorithm summarized in Figure V.12 is carried out.

1. **For every** cell in the queue:
   A. **Calculate** the ToE of the cell (say, originated by source i).
   B. **If** the ToE of the cell is smaller than $1/C_W$ (the service time):
      i) **Discard** the cell.
      ii) **Increment** $DS[i]$.
2. Suppose that N_L cells were left in the queue after Step 1.
   A. **Sort** the remaining cells in the queue in a non-decreasing order of their ToE.
   B. **For each** cell in position j ($0 \le j \le N_L$-1) in the queue:
      i) **Set** in *Service_List*[j] the source in which the cell belongs to (using *Add*).
      ii) **Set** in *Deadline_List*[j] the cell deadline (using *Add*).
3. **For every** source $i \in \{S_S\}$ having at least one cell in the queue:
   A. **Calculate** $Cost[i]$ to be the value of $CLPR[i]$ should an additional cell from source i be discarded:

$$Cost[i] = CLPR[i]\Big|_{\substack{\text{if additional} \\ \text{cell discarded}}} = \frac{(DS[i]+1)}{A[i] \times ACLP[i]} = CLPR[i] + \frac{1}{A[i] \times ACLP[i]}.$$

4. **Set** *list_index* = 1.
5. **Repeat** N_L-1 times:
   A. **If** *Deadline_List*[*list_index*] < *list_index*/C_W:
      i) **Set** *least_cost_index* = *list_index*.
      ii) **Set** *least_cost_value* = *Cost*[*Service_List*[*list_index*]].
      iii) **For** j = *list_index*-1 **downto** 0:
         a) **If** *Cost*[*Service_List*[j]] < *least_cost_value*:
            (1) **Set** *least_cost_index* = j.
            (2) **Set** *least_cost_value* = *Cost*[*Service_List*[j]].
      iv) **Calculate** the value of *Cost*[*Service_List*[*least_cost_index*]] (for simplicity, we use the letter i instead of *Service_List*[*least_cost_index*]):

$$Cost[i] = Cost[i] + \frac{1}{A[i] \times ACLP[i]}.$$

      v) *Delete*(*least_cost_index*) from *Service_List*[].
      vi) *Delete*(*least_cost_index*) from *Deadline_List*[].
   B. **Else**:
      i) **Increment** *list_index*.
6. **Service** the first queued cell from the source identified by *Service_List*[0].

Figure V.12: STEBR Allocation in Time O(N^2) in a Wireline System

5. Discussion

The STEBR algorithm runs in time $O(N^2)$, where N denotes the number of cells originally in the queue. Steps 1 and 6 are performed in time $O(N)$ each, and so is Step 3 (since the number of sources having cells in the queue at that time is at most $N_L \leq N$). Step 2 (queue sorting) may be run in time $O(N)$ if the queue is maintained sorted in a non-decreasing order of the cell deadlines at all times as follows. When a cell arrives at the queue, it is enqueued in the appropriate location, demoting all cells having larger deadlines one position backward; and when a cell has been serviced or is discarded (dequeued), all cells having larger deadlines are promoted one position forward. The next operation in Step 2 is to update *Service_List*[] and *Deadline_List*[] by scanning the (already sorted) queue; updating each list takes time $O(N)$. Step 4 runs in O(1) time.

Theorem V.2: *Step 5 of the algorithm, and the algorithm overall, run in time* $O(N^2)$.

Proof: Step 5 of the algorithm may be written as in Figure V.13, where N_L is replaced by its upper bound N:

1. **Set** *list_index* = 0.
2. **Set** *sum* = 0.
3. **For** $j = 1$ **to** N-1:
   A. **If** a cell indexed *list_index* is to expire (repetition E):
      i) **Calculate** *sum* = *sum* + *list_index* + 1.
   B. **Else** (the cell is scheduled on time; repetition O):
      i) **Increment** *list_index*.

Figure V.13: Simplified Representation of STEBR's Step 5

The variable *sum* counts the number of times the operations in the inner loop of Step 5 are performed. We are interested to know the maximum possible value of *sum*, which presents the worst-case performance of the algorithm.

Lemma V.3: *The sequence of repetitions O-O-...-O-E-E-...-E maximizes sum, where O and E represent (see* Figure V.13*) the events of cell which is scheduled on-time and cell which will expire, respectively.*

Proof: Consider a different sequence of repetitions having an E before O. Suppose that the value of *list_index* before reaching this E-O sequence is a. After the sequence E-O, sum will be equal a+1. On the other hand, following the sequence O-E one gets $sum = a+2$. By a sequence of replacements of every E-O by O-E in the original sequence (each of which increases *sum*), we get the sequence O-O-...-O-E-E-...-E that maximizes *sum*. ■

Lemma V.4: *The maximum value of sum is time $O(N^2)$.*

Proof: Suppose the sequence of N-1 steps that maximizes *sum* comprises n O's followed by N-1-n E's (following Lemma V.3). We write *sum* as a function of n and find the extreme value of its continuous extension:

$$sum = (N-1-n)\times(n+1)$$

$$\frac{d}{dn}sum = N-1-n-n-1 = 0$$

$$n_{opt} = \frac{N}{2}-1.$$

Since $\frac{d^2}{dn^2}sum = -2$, it follows that $\max\{sum\} = sum\big|_{n_{opt}} = \frac{N}{2}\times\frac{N}{2} = \frac{N^2}{4}$.

If N is odd, then since *sum* is quadratic in n, the maximum is quadratic in N as well:

- Option 1: (N-1)/2 O's followed by (N+1)/2 E's. The value of *sum* then is

$$sum_1 = \left(\frac{N+1}{2}\right)\times\left(\frac{N+1}{2}\right) = \frac{N^2+2N+1}{4},$$

- Option 2: (N+1)/2 O's followed by (N-1)/2 E's. The value of *sum* then is

$$sum_2 = \left(\frac{N+3}{2}\right)\times\left(\frac{N-1}{2}\right) = \frac{N^2+2N-3}{4}.$$

The value of sum_1 is clearly larger than sum_2 for all $N > 0$. Consequently, we write:

$$\max\{sum\} = \begin{cases} \frac{N^2}{4}, & \text{for } N \text{ even} \\ \frac{N^2+2N+1}{4}, & \text{for } N \text{ odd.} \end{cases}$$

In either case, the maximum value of *sum* is time $O(N^2)$. ■

The Proof of Theorem V.2 follows directly from Lemma V.4. ■

Each queued cell requires a finite number of memory elements, thus the total required memory for the operation of the algorithm is $O(N)$. We now discuss a necessary and sufficient condition regarding the constellation of cells in the queue to avoid cell discarding.

Corollary V.5: *Any cell whose deadline to beginning of service (DBS) is within* $[T_j, T_{j+1})$ *for* $j \geq 0$ *and which is scheduled for transmission in one of the service slots* T_0, T_1, ..., T_j, *shall not be discarded.*

Proof: Self-explanatory. ■

Lemma V.6: *Considering the set of the first j cells in the queue, a necessary and sufficient condition to avoid discarding is satisfied if*

$$Deadline(j_x) \geq T_{j_x-1}, \qquad \forall 1 \leq j_x \leq j, \tag{V.7}$$

where $Deadline(j_x)$ is the DBS of the j_x^{th} cell.

Proof: Assume Condition (V.7) is met. Schedule the j_x^{th} cell ($\forall 1 \leq j_x \leq j$) in service slot T_{j_x-1}. From Corollary V.5, no loss is expected. On the other hand, if for a cell j_x ($1 \leq j_x \leq j$), $Deadline(j_x) < T_{j_x-1}$, then the first j_x cells in the queue should be scheduled on j_x-1 slots (since the queue is sorted according to deadlines). Thus, one cell must be discarded. ■

Lemma V.6 provides the reason why Condition (V.7) is tested in Step 5.A of the algorithm. If the condition is satisfied, we end the loop; otherwise, we check for the least-cost cell for discarding.

At Step 5, if a given cell is not scheduled for service before its DBS (Step 5.A.iii), then it is guaranteed that one entry in each of *Service_List*[] and *Deadline_List*[] will be deleted. We wish to explain why. Suppose that the time in which the algorithm is run is T_0, and the value of *list_index* equals j (testing the $(j+1)^{th}$ cell) when the *if* statement is 'TRUE' (meaning that *Deadline_List*[*Service_List*[j]] < T_j). Because the queue is sorted

by the cell deadlines, $Deadline\_List[Service\_List[j_x]] \leq Deadline\_List[Service\_List[j]]$, for all j_x $(0 \leq j_x < j)$. If no service slot could be found for the cell corresponding to location j prior to its expiration, then it should obviously be deleted. On the other hand, if a cell with a smaller cost in location j_x $(0 \leq j_x < j)$ is found, then exchanging its place with the one in location j is not worthwhile; the cell at location j_x should be deleted instead since $Deadline\_List[Service\_List[j_x]] \leq Deadline\_List[Service\_List[j]] < T_j$.

The uniqueness of the algorithm lies in the fact that the source costs not only can change with time (vary between service slots), but they can also change (increase) dynamically during the process of decision making. The value of appropriate location in *Cost*[] array is recalculated whenever a future decision about cell discarding is made. A larger value is then possibly utilized (depending on the CLP ratios) to assign a higher priority to the source that originated the cell to be discarded in the future.

6. Operation of the Algorithm

In this section, we again consider the example of Figure V.11, to demonstrate the operation of the algorithm as implemented in Section 4. The CLPRs of the sources (implying the source costs) are given in Table V.3. The values of the relevant variables of the algorithm throughout its operation are given in Table V.4. From Table V.4, after the last (7^{th}) loop in Step 5, the first element in *Service_List* is 2; thus, the cell to be serviced on current slot is the first cell from Source 2, i.e., 2_1.

Source	CLPR			
	Initial Value	After Discarding 1 Cell	After Discarding 2 Cells	After Discarding 3 Cells
1	0.36	0.40	0.44	0.48
2	0.41	0.43	0.45	0.47
3	0.42	0.46	0.50	

Table V.3: CLPRs (Source Costs) of the Example given in Figure V.11

Stage	List	Queue Index								*list_*	*Cost*[...]		
		0	1	2	3	4	5	6	7	*index*	1	2	3
After Steps 1-4	*Service_List*	1	1	2	2	2	3	3	1	1	**0.40**	**0.43**	**0.46**
	Deadline_List	0	0	0	1	1	2	2	2				
After the 1st Loop of Step 5	*Service_List*	1	2	2	2	3	3	1		1	**0.44**	0.43	0.46
	Deadline_List	0	0	1	1	2	2	2					
After the 2nd Loop of Step 5	*Service_List*	1	2	2	3	3	1			1	0.44	**0.45**	0.46
	Deadline_List	0	1	1	2	2	2						
After the 3rd Loop of Step 5	*Service_List*	1	2	2	3	3	1			2	0.44	0.45	0.46
	Deadline_List	0	1	1	2	2	2						
After the 4th Loop of Step 5	*Service_List*	2	2	3	3	1				2	**0.48**	0.45	0.46
	Deadline_List	1	1	2	2	2							
After the 5th Loop of Step 5	*Service_List*	2	2	3	3	1				3	0.48	0.45	0.46
	Deadline_List	1	1	2	2	2							
After the 6th Loop of Step 5	*Service_List*	2	3	3	1					3	0.48	**0.47**	0.46
	Deadline_List	1	2	2	2								
After the 7th Loop of Step 5	*Service_List*	2	3	1						3	0.48	0.47	**0.50**
	Deadline_List	1	2	2									

Table V.4: STEBR Variables throughout Execution of the Example (Figure V.11)

G. PROOF OF OPTIMALITY OF STEBR

This section is devoted to proof of optimality of the STEBR algorithm when no information about future cell arrivals is available. STEBR makes *locally*-optimal (greedy) decisions, thus we briefly introduce the concepts of greedy algorithms. Next we show that, although the algorithm is not globally optimal because of the non-suboptimal structure of the problem, it is optimal in a local sense (i.e., it makes an optimal service decision at every service slot).

1. Greedy Algorithms

Greedy algorithms are used for solving optimization problems. A greedy algorithm obtains an optimal solution to a problem by making a sequence of choices. At each decision point in the algorithm, the choice that seems best at that moment is chosen [19]. Most greedy algorithms that solve optimization problems present two properties: the greedy-choice property and the optimal substructure.

The *greedy-choice property* states that by making locally-optimal (greedy) choices, one can arrive at a globally-optimal solution. One implication of this property is that the choice at a given decision point does not depend on solutions to subproblems. A problem exhibits *optimal substructure* if an optimal solution to the problem contains within it optimal solutions to subproblems.

2. STEBR Algorithm is Greedy and Locally Optimal

In this section, we show that STEBR provides optimal scheduling in a single-queue single-server system, where no future knowledge of cell arrivals is known. We use a technique called algorithmic proof [97] having the following three steps: 1) State the algorithm, 2) Prove that it always terminates, and 3) Prove that it yields an optimal solution.

The algorithm statement has been provided in Section F.4. Other than sequential executions, the algorithm contains two loops. The loop in Step 1 includes a finite set of executions and is performed exactly N (the number of cells originally in the queue) times. The main loop in Step 5 is repeated up to N_L-1 $\leq N$-1 times; the queue is scanned at each repetition j ($1 \leq j \leq N_L$-1) no more than j times. Thus, the algorithm terminates at all times. Proof of optimality of the algorithm is provided hereafter. The proof comprises the following stages:

- Definitions of a source cost and an *objective function*, followed by a proof that the proposed objective function is the one desired in our case.
- A proof that, given a situation in which one cell must be discarded, the objective function is obtained by discarding a cell from the source having the least cost.
- A proof that, given a situation in which several cells must be discarded, a sequence of intermediate decisions of the previous stage, each that keeps the objective function at its minimum, results in an optimal overall decision.
- A proof of a necessary condition that an optimal algorithm must satisfy over an infinitely-long period.
- Proof of optimality of the algorithm by first assuming that a better algorithm exists and then contradicting this assumption.

- Satisfaction of the necessary condition by the algorithm.

a. Definitions of Source Cost and Objective Function

Definition V.7: *The cost of source $i \in \{S_S\}$ is defined as its CLPR, in case an additional cell from this source is discarded:*

$$Cost[i] = CLPR[i]\Big|_{\substack{\text{if additional} \\ \text{cell discarded}}} = \frac{(DS[i]+1)}{A[i]\times ACLP[i]} = CLPR[i] + \frac{1}{A[i]\times ACLP[i]},$$

where $DS[i]$ and $A[i]$ are the number of cells discarded and arrived thus far at the queue, respectively, from source i.

Lemma V.8: *The objective function of the system, f, is defined as*

$$f = \max_{i\in\{S_S\}}\{CLPR[i]\}.$$

The purpose is to find a scheduling assignment that minimizes f.

Proof: We consider an arbitrary scheduling algorithm and a set of active sources, $\{S_S\}$. The performance of the scheduler is based on the *least QoS* associated with one or more of the sources $i \in \{S_S\}$ over a long period. In other words, over a long period, even if the ACLP requirement is met for all but one source, then this violation concludes that the scheduler under consideration *cannot service* the set $\{S_S\}$ (or the set is outside the admissible region). The QoS supplied to a source by the scheduler is its instantaneous CLP (the maxCTD is already incorporated into it). However, since sources may have different ACLPs, then the CLPR is the measurement for comparison among sources having a variety of loss requirements. Constraints on the scheduler performance can thus be determined by the maximum CLPR value over $\{S_S\}$, which is the objective function, f. It is obvious that one wants to schedule service for the queued cells such that f is minimized, in order to increase the size of $\{S_S\}$ to the maximum allowable value. ■

b. Least-Cost Cell Discarding Minimizes the Objective Function

Lemma V.9: *Whenever a cell must be discarded due to contention on a service slot, the objective function is minimized if the discarded cell is the one (among the*

candidates for discarding) that will have the least CLPR, after discarding the cell. (Equivalently, a cell to be discarded is the one having the least cost.)

Proof: The issue here is not whether to discard a cell or not, for the decision on cell discarding is pre-determined. The question at hand is: given that one cell out of a set of cells *must* be discarded, which to discard?

The proof follows the objective function in Lemma V.8. Given the CLPR of all sources $i \in \{S_S\}$ at decision time, assuming that one cell from a subset of sources $\{S_D\} \subseteq \{S_S\}$ must be discarded sometime in the future due to contention on a service slot. We calculate the CLPR of all sources $i_x \in \{S_D\}$ *after* a cell from source i_x is discarded (this is exactly the definition of $Cost[i_x]$). It is clear that $Cost[i_x] > CLPR[i_x]$ for the non-trivial case $A[i_x] > 0$. The new value of f, *after* the cell discarding is then

$$f = \max\left\{ \max_{i \in \{S_S\} - \{S_D\}} \{CLPR[i]\}, \max_{i_x \in \{S_D\}} \{Cost[i_x]\} \right\}.$$

The first term in the outer braces does not influence the value of f because it concerns sources that are not candidates for cell discarding. In $\{S_D\}$, the source having the least value of *Cost* is the one that minimizes f. ■

c. Optimal Decision is Obtained by Minimizing Objective Function

The scheduling of queued cells may involve cell discarding due to contention on service slots. Whenever a cell *must* be discarded due to contention on a service slot, an optimal (intermediate) decision regarding the cell to discard would be to minimize the objective function.

Corollary V.10: *Given a situation in which several cells must be discarded, a sequence of intermediate decisions regarding cell discarding, each that keeps the objective function at its minimum, results in an optimal overall decision (minimum value of f).*

Proof: The proof follows immediately from Lemma V.9, using induction on the number of decisions made regarding cell discarding. ■

d. Necessary Condition

Lemma V.11: *Over an infinitely-long period and for the maximal possible number of traffic streams, an optimal scheduling scheme balances the CLP ratios of all active sources to values approaching 1 from below.*

Proof: The CLPR of a source depends on two non-decreasing factors: the number of cells discarded and the number of cells arrived. An increase in the former increases CLPR while an increase in the latter decreases it. When a decision to discard a cell is made, the source chosen (say, i) has the least value of *Cost* (Lemma V.9). Therefore, the number of cells discarded from source i would be incremented, thereby increasing the expected $CLPR[i]$ (and $Cost[i]$). Overall, over an infinitely-long period, the CLPRs of all active sources are balanced to the same value. When a maximum number of possible sources is present (although this number is unknown), all CLPRs will approach 1 from below. At this point, an additional source would cause a violation of the QoS requirements of all sources. The condition is insufficient because decisions regarding cell discarding (whether to discard or not, rather than which to discard) might not be optimal. An example of an algorithm that satisfies this condition but is not optimal is the BCLPR.

■

e. Proof of Optimality

We assume that at decision time no more cells would arrive at the queue. Thus, the objective function can only increase. An optimal algorithm would keep the increase in the objective function (if any) to the minimum possible. We also assume that the system is observed at steady state; however, the number of cells that have arrived thus far is finite. Current time is denoted as T_0 and service slot k ($0 \le k \le L$) relates to the time of the semi-closed period $[T_k, T_{k+1}) = \left[T_0 + k/C_W, T_0 + (k+1)/C_W\right)$, where L is the latest slot during which the last queued cell can be serviced. If $Deadline(N_L\text{-}1)$ denotes the DBS of the last cell, then

$$L = \lfloor [Deadline(N_L - 1) - T_0] \times C_W \rfloor.$$

The STE scheme presents minimum loss in a homogeneous buffer [66]; thus, the queue is sorted according to non-decreasing deadlines (or DBSs). This means that STE does not differentiate between sources having different allowed loss and delay (which implicitly assigns them different priorities in service). Therefore, in the case of contention for service among multiple cells, while the STE approach discards cell(s) regardless of the distinct loss requirements of the contending sources, the STEBR takes these loss requirements into account in order to minimize *f*.

Property V.12: Consider a simple algorithm *Count*. Using this algorithm, the maximum number of available slots for service, N_{SS}, is obtained by summing non-negative integers as follows:

- **Set** $N_{SS} = 0$
- **For** $k = 0$ **to** L
  - **Calculate** $N_{SS} = N_{SS} + \min\{k+1 - N_{SS}, |\{G_k\}|\}$,

where $\{G_k\}$ denotes the set of cells having DBS within $[T_k, T_{k+1})$, $0 \le k \le L$, and $|\{G_k\}|$ denotes the number of cells within $\{G_k\}$.

Suppose that, prior to running *Count*, every slot k ($0 \le k \le L$) is marked as idle or "I". At any stage throughout the algorithm run time, whenever N_{SS} is increased in the loop by some value $a \ge 1$, it implies that the current slot and the previous a-1 "I" slots (if any) can be utilized for service. These slots are then marked as used or "U". ■

Example: Consider the following queued cells with their DBS drawn in Figure V.14 as vertical arrows. The various quantities in the *Count* algorithm are given in Table V.5.

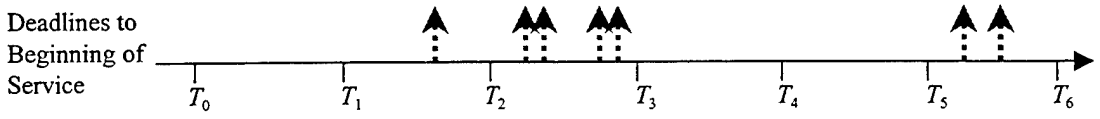


Figure V.14: *Count* Algorithm, Example of Operation

Stage	$\lvert\{G_k\}\rvert$	N_{SS}	Slot Status					
			T_0	T_1	T_2	T_3	T_4	T_5
Begin			**I**	**I**	**I**	**I**	**I**	**I**
$k=0$	0	0+min{1, 0} = 0	I	I	I	I	I	I
$k=1$	1	0+min{2, 1} = 1	I	**U**	I	I	I	I
$k=2$	4	1+min{2, 4} = 3	**U**	U	**U**	I	I	I
$k=3$	0	3+min{1, 0} = 3	U	U	U	I	I	I
$k=4$	0	3+min{2, 0} = 3	U	U	U	I	I	I
$k=5$	2	3+min{3, 2} = 5	U	U	U	I	**U**	**U**
End			**U**	**U**	**U**	**I**	**U**	**U**

Table V.5: *Count* Algorithm, Variable Values along Operation

Result V.13: *The maximum number of slots that can be utilized by any scheduling algorithm is* $N_{SS} \leq L+1$. ■

Corollary V.14: *The minimum number of cells to be discarded by any algorithm is* N_L-$N_{SS} \geq 0$, *where* N_L *marks the number of cells left in the queue after Step 1 of STEBR (discarding of expired cells).* ■

Next, we consider the number of cells that are discarded by all rational scheduling algorithms. A rational scheduler tries to minimize f whenever possible; specifically, if a cell can be serviced at a given slot without increasing f, then a rational scheduler utilizes this slot rather than leaving it idle.

Lemma V.15: *The number of cells to be discarded by all rational algorithms is equal to* N_L-N_{SS}.

Proof: The proof follows a sequence of steps that show that any rational algorithm would use exactly N_{SS} (available) slots for service, according to *Count*, and discard exactly N_L-N_{SS} cells (if any). The proof relies on the argument that if a non-rational algorithm services *additional* cells instead of the ones originally scheduled for service, the objective function, f, is not increased.

From Corollary V.14, in the trivial case where $N_{SS} = N_L$, every scheduler can assign cells to slots without discarding any of them (e.g., service the cells as they are sorted), thus maintaining f at the same value.

When $N_{SS} < N_L$, the *Count* algorithm assigns the slots as shown in Figure V.15. It is guaranteed that slot L is always of type U because the last cell's DBS is greater than or equal to T_L. If the algorithm has produced at least one slot of type I, then *all* cells having DBS greater than or equal to the end of the first such slot (T_{n+1} in Figure V.15) *should be scheduled on time* by a rational algorithm on a subset of slots n+1, n+2, ..., L without any cell discarding. (The case in which the first slot is of type I (n = 0) is a special case, guaranteeing that all queued cells can be assigned slots on time.) An irrational algorithm would either unnecessary discard one of the cells having DBS ≥ T_{n+1} or unnecessary occupy one of the first n slots at the expense of another cell. Following are examples that demonstrate these two cases of unnecessary discarding by an irrational assignment.

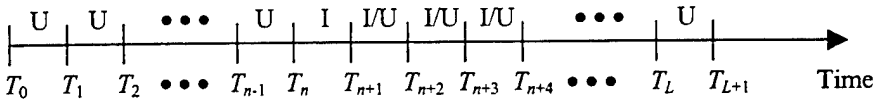


Figure V.15: Slot Assignment by Algorithm *Count*

Example: In Figure V.16, an irrational algorithm services three cells and discards two cells. Since $N_{SS} = 4$, it is possible to service four cells and discard only one without increasing f. The example demonstrates how an irrational assignment unnecessarily discards a cell (indexed 4) that has DBS longer than the end of the first I slot (T_3).

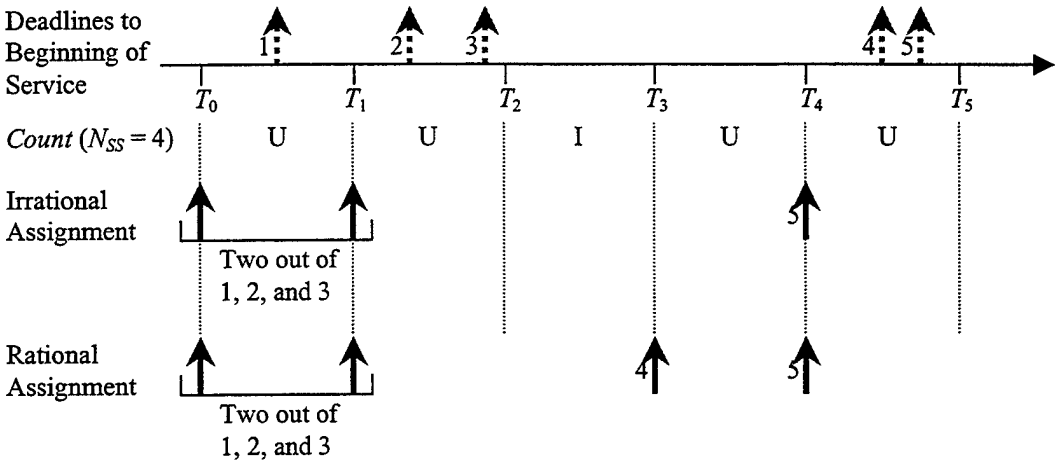


Figure V.16: Irrational Assignment, First Example

Example: In Figure V.17, an irrational algorithm services three cells and discards two cells. Since $N_{SS} = 4$, it is possible to service four cells and discard only one without increasing f. The example shows how an irrational assignment schedules a cell (indexed 4) having DBS longer than the end of the first I slot (T_3) on slot preceding the first I slot, thus unnecessarily discarding one additional cell out of the first three.

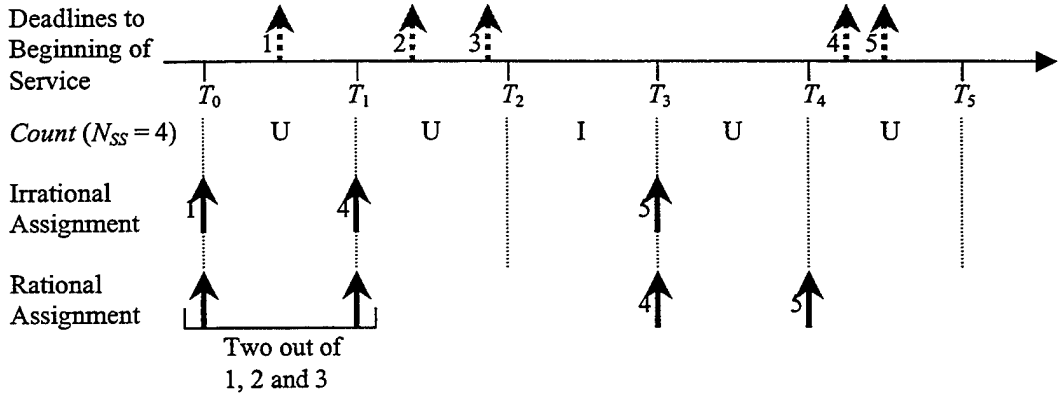


Figure V.17: Irrational Assignment, Second Example

Continuing the discussion about Figure V.15, we now concentrate only on the first n slots (0 through n-1), where cell discarding is unavoidable. The way *Count*

assigns the slot notations I and U ensures that the following properties apply: $|\{G_n\}| = 0$, $|\{G_{n-1}\}| \geq 1$, $|\{G_{n-1}\}| + |\{G_{n-2}\}| \geq 2$, ..., and generally

$$\sum_{k=k_x}^{n-1} |\{G_k\}| \geq n - k_x, \qquad 0 \leq k_x \leq n\text{-}1. \tag{V.8}$$

From the properties of Equation (V.8), we conclude that any rational algorithm should utilize *all* the first n slots. If a slot k ($0 \leq k \leq n$-1) is left idle by a scheduler, it implies that at lease one cell having DBS within $[T_k, T_n)$ is unnecessarily discarded (i.e., the discarded cell could have been serviced on slot k); or this cell occupies one of the slots [0, k-1] thereby causing unnecessarily discarding of a cell having DBS within $[T_0, T_k)$. Repeating this argument for every such non-utilized slot guarantees termination of the process because $n \leq L$+1 (from Result V.13) such that any rational algorithm would utilize all the first n slots. Consequently, exactly N_{SS} slots are utilized and exactly N_L-N_{SS} cells are discarded by rational algorithms. ■

Corollary V.16: STEBR algorithm is rational because for any set of queued cells, it discards no more than N_L-N_{SS} cells. On the other hand, at least N_L-N_{SS} cells must be discarded regardless of the algorithm. ■

Theorem V.17: *Given a set of cells in the queue and no future arrivals at the queue, the STEBR algorithm schedules the cells for transmission optimally, i.e., the objective function is kept minimal.*

Proof: The approach used to prove the optimality of STEBR algorithm is as follows:

- Assume that there exists a (distinct) rational scheduling algorithm Y that outperforms STEBR. Run the two algorithms and obtain the ordered sets of cells to be *serviced* by algorithms STEBR and Y, $\{OS_{STEBR}\}$ and $\{OS_Y\}$, respectively, and the sets of cells to be *discarded* by algorithms STEBR and Y, $\{OD_{STEBR}\}$ and $\{OD_Y\}$, respectively. (The letter O in the set marks the Original problem and the second letter, S or D, marks the cells Serviced or Discarded, respectively, by the algorithm.)

- Transform the original problem into a modified one by eliminating *similar* decisions made by the two algorithms regarding cell service and cell discarding.

Show that the decisions to be made by the algorithms in the transformed problem are the same as those performed in the original problem.

- Show that, in the transformed problem, algorithm Y does not make scheduling decisions as well as STEBR does.

We want to analyze the *differences between decisions* made by STEBR and Y algorithms, regarding which cells to service and which to discard. Let us examine (in any given order) all the cells within $\{OS_{STEBR}\}$ and $\{OD_{STEBR}\}$,[10] and transform the problem using the following two steps:

- For every cell within $\{OS_{STEBR}\}$, if its source is also the source of a cell within $\{OS_Y\}$, delete the two cells from both sets. These two cells do not affect the value of f in either algorithm, and the slot in which they are serviced does not have any importance in this case.

- For every cell within $\{OD_{STEBR}\}$, if its source is also the source of a cell within $\{OD_Y\}$, delete the two cells from both sets and update the cost of that source. Discarding of these two cells increases f to the same value under both algorithms. The "contribution" of a source to the value of f does not depend on the specific cells discarded from the source, but only on the total number of such cells.

We denote $\{TS_{STEBR}\}$, $\{TD_Y\}$, $\{TS_Y\}$, and $\{TD_{STEBR}\}$ as the sets of *sources* of the cells remained in $\{OS_{STEBR}\}$, $\{OD_Y\}$, $\{OS_Y\}$, and $\{OD_{STEBR}\}$, respectively, after the above two steps have been performed. (The first letter, T, in the above four sets marks the Transformed problem.) Outcomes of the steps are $\{TS_{STEBR}\} = \{TD_Y\}$ and $\{TS_Y\} = \{TD_{STEBR}\}$. The number of cells, m, in each of $\{TS_{STEBR}\}$, $\{TD_Y\}$, $\{TS_Y\}$, and $\{TD_{STEBR}\}$ is equal (guaranteed by Lemma V.15); it is obtained by

$$m = \frac{1}{2}\left(\sum_{i \in \{S_S\}} |S_i - Y_i|\right),$$

where S_i and Y_i are the number of cells from source i found in $\{OS_{STEBR}\}$ and $\{OS_Y\}$, respectively. The case $m = 0$ is trivial, since the number of cells to be serviced from each source by both algorithms is the same, thus increasing the objective function to the same

[10] Sets $\{OS_Y\}$ and $\{OD_Y\}$ could be considered as well.

value (STEBR is optimal because f is non-decreasing). The sources within $\{TS_{STEBR}\}$ and $\{TS_Y\}$ have no common elements; otherwise, they would have been eliminated during the process of problem transformation.

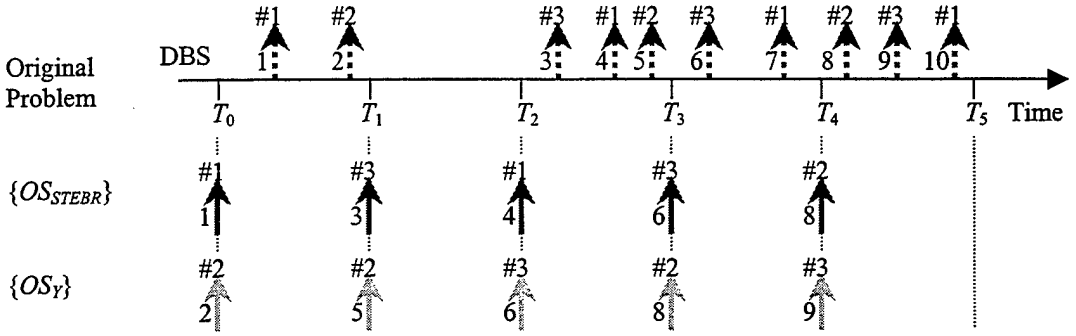
It is convenient to consider the following modified problem. Find an optimal scheduling assignment given that for every l ($1 \leq l \leq m$) the queue contains exactly two cells having DBS within $[T_{l-1}, T_l)$: one cell is originated by the source in the l^{th} position within $\{TS_{STEBR}\}$, and the second is originated by the corresponding source within $\{TS_Y\}$. The source costs are obtained *after* transformation of the problem; each decision to discard a cell from source i increases the source cost *linearly* by Δ_i:

$$\Delta_i = \frac{1}{A[i] \times ACLP[i]}.$$

Example: An example of the outcome of the transformation steps just described is shown in Figure V.18a, where #i marks a cell originated by source i. Figure V.18b uses the example shown in Figure V.18a to demonstrate the transformation of the problem; the black arrows relate to the cells within $\{TS_{STEBR}\}$ while the gray arrows relate to these of $\{TS_Y\}$.

We next show that both algorithms would make the same scheduling decisions for the transformed problem. Consider the costs of all sources within $\{TS_{STEBR}\}$ in the new problem; these are the final (i.e., largest) costs since they reflect all the cell-discarding decisions from these sources made by the algorithm with respect to the original problem. Similarly, the costs of all sources within $\{TD_{STEBR}\}$ are smaller than their final (largest) values obtained after the STEBR has run on the original problem. Equivalent argument can be applied to the costs of the sources within $\{TS_Y\}$ and $\{TD_Y\}$. Each element in $\{TD_{STEBR}\}$ represents a discarded cell by STEBR; for each such cell, the algorithm could discard instead (in the original problem) the corresponding cell in $\{TS_{STEBR}\}$ or any of the cells preceding it, but it chose not to. In the transformed problem, STEBR has the same discarding choices again, thus it is expected to repeat its decisions made in the original problem. The same applies for algorithm Y. In summary, the constellation of the transformed problem allows both algorithms to repeat their exact

(presumably optimal) decisions made in the original problem with respect to the sets (of size m each) of cells to be serviced and discarded.

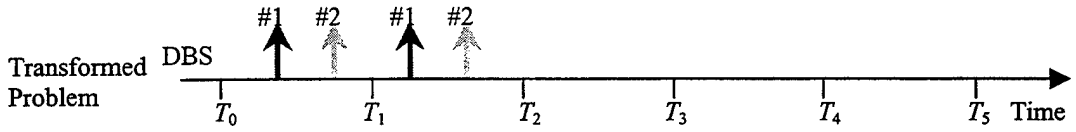


$\{OD_{STEBR}\}$ = {2(#2), 5(#2), 7(#1), 9(#3), 10(#1)}, $\{OD_Y\}$ = {1(#1), 3(#3), 4(#1), 7(#1), 10(#1)},

$m = \left(|S_1 - Y_1| + |S_2 - Y_2| + |S_3 - Y_3|\right)/2 = \left(|2-0| + |1-3| + |2-2|\right)/2 = 2$,

$\{TS_{STEBR}\} = \{TD_Y\}$ = {#1, #1}, $\{TS_Y\} = \{TD_{STEBR}\}$ = {#2, #2}.

(a) Differences in Decision Making between STEBR and *Y* Algorithms



Updated Source Costs:
Cost{Source 1} = Original Cost {Source 1} + $2\Delta_1$
Cost{Source 2} = Original Cost {Source 2}
Cost{Source 3} = Original Cost {Source 3} + Δ_3

(b) Transformed Problem

Figure V.18: Example of Problem Transformation

Out of the $2m$ cells in the transformed problem, m must be discarded. Whenever algorithm Y discards one of m cells from $\{TD_Y\}$ (say, from source $i \in \{TD_Y\}$), and STEBR discards one of m cells from $\{TD_{STEBR}\}$, the following relation holds thereafter:

$$\max_{j \in \{TD_{STEBR}\}} \left[f_{STEBR} \Big|_{\substack{\text{after discarding one} \\ \text{cell from source } j}} \right] \le f_Y \Big|_{\substack{\text{after discarding one} \\ \text{cell from source } i \in \{TD_Y\}}} \le f_Y \Big|_{\substack{\text{after discarding all} \\ \text{cells from sources} \in \{TD_Y\}}}, \quad \text{(V.9)}$$

where f_{STEBR} and f_Y are the values of the objective functions under STEBR and Y algorithms, respectively. The right inequality is satisfied because the CLPR (and the cost)

of the discarded source can only increase (the right term in this inequality states the maximum value over all sources discarded by Y). The left inequality is satisfied because STEBR algorithm always chooses to discard a cell from the source having the least cost (Lemma V.9); since STEBR decides *not to discard* any cell from sources within $\{TS_{STEBR}\} = \{TD_Y\}$, it means that the costs of all the sources within $\{TD_{STEBR}\}$ are *smaller than or equal to* all costs of sources within $\{TD_Y\}$. Equation (V.9) and the previous statement are correct for the discarding of all m cells. Consequently, (in relation to the transformed problem) *all* decisions made by algorithm Y regarding discarding of cells that correspond to sources within $\{TD_Y\}$ are inferior to (or increase f more than) the discarding decisions performed by STEBR. Since the final values of f under both algorithms in the transformed problem will be identical to the corresponding ones in the original problem (all service and discarding decisions are the same as proven earlier), algorithm Y cannot be optimal. ■

f. Satisfaction of Necessary Condition

The STEBR algorithm follows the concept of balancing the CLPRs of all active sources. If it decides to discard a cell, it immediately compensates the source of that cell by increasing its cost. Thus, the condition is satisfied. ■

H. SIMPLIFICATIONS OF STEBR ALGORITHM

This section proposes simplifications to the STEBR algorithm described in Section F.4. An ad-hoc relaxation of the $O(N^2)$ run time, as well as an implementation in linear time are supplied. The section concludes with a discussion on the dynamic nature of the scheduling problem and related future work.

1. Ad-Hoc Relaxation of $O(N^2)$ Run Time

The STEBR algorithm presented earlier runs in time $O(N^2)$, which may be very time consuming when the number of cells in the queue grows to large values (e.g., several hundreds). Empirical results show that, for the source models used in this work, it

is sufficient to run the algorithm during every service slot over only the first 10-30 cells in the queue for loss values that are only about 1% larger than those obtained by scanning the entire queue. Run time of the algorithm can thus be significantly reduced using this ad-hoc relaxation to time O(N).

2. STEBR Implementation in Linear Time

a. Concepts

We propose here a different implementation of the STEBR algorithm that runs in linear time. The outcome of any scheduling technique is the index of a single cell to be serviced at current service time. The previous implementation of the STEBR algorithm considers the queued cells from head to tail, one at a time; for every cell that is presumably scheduled late by the earliest-deadline-first approach, the queue is scanned from that point toward its head. Since the number of cells scanned toward the head of the queue grows with the instantaneous location within the queue, one gets run time that is O(N^2). The idea behind the new approach is to scan the queue only *once* and perform a constant number of calculations durind each scanning. The first two steps of the algorithm, expired cell discarding and queue sorting by cell deadlines, are the same as in the previous approach.

We begin by running the *Count* algorithm to obtain N_{SS}, the number of service slots that can be utilized in the current constellation of cells in the queue (see Property V.12). If $N_{SS} = N_L$, no contention on service slots between the cells will occur, and the first cell in the queue is immediately chosen for service. Otherwise, at most N_{SS} out of N_L cells can be scheduled on time.

While scanning the queue only once, the cells are assigned costs. Consider the costs of the cells as the queue is scanned from the head to the tail; the first cell from a source (say, i) is assigned the value of $CLPR[i]$, given an additional cell discarding from i. In the previous approach, the following cells from source i were assigned larger costs only if former cells had been set for discarding. An equivalent approach is to assign to every *cell* (say, from source i) a cost; the first (oldest) cell from source $i \in \{S_S\}$ is

assigned the value $CLPR[i]+\Delta_i = CLPR[i]+(A[i]\times ACLP[i])^{-1}$ and every newer cell from this source gets a value larger by $\Delta_i = (A[i]\times ACLP[i])^{-1}$ than that of the former cell from i. By doing so, we emphasize the importance given to the cells from the given source: the first one has the least cost; however, if it is discarded, the second has a larger cost; if the first two cells are discarded, the third has a larger cost than the second, and so on.

We now describe key concepts of the new approach. Scan the queue from the tail of the queue to its head (opposite of the $O(N^2)$ STEBR algorithm). On slot T_L, only one out of $|\{G_L\}|$ cells can be scheduled, thus the one having the largest cost is chosen. The remaining $|\{G_L\}|$ -1 cells (if any) continue to contend on the preceding slot(s). Since on slot T_{L-1} only one out of $|\{G_{L-1}\}| + |\{G_L\}|$ -1 cells can be scheduled, the one having the largest cost is chosen and the remaining cells continue to contend on the preceding slot(s). Generally, on slot k_x ($0 \le k_x \le L$), one chooses a single cell out of $\sum_{k=k_x}^{L} |\{G_k\}| - (L - k_x)$ cells having the largest cost. The costs assigned to cells from the same source (i) increase linearly by Δ_i as we go from the tail to the head of the queue (with non-decreasing deadlines). Thus, if multiple cells from the *same* source contend on a given slot, it is sufficient to check the cost of the *newest* one (having the largest cost). Therefore, the costs of at most $|\{S_S\}|$ (a constant number) cells need to be compared for every slot. The cell chosen finally for service at the decision time is the oldest cell from the source that is assigned Slot 0. The overall run time of the algorithm is at most $|\{S_S\}| \times (N_L\text{-}1) \le O(N)$.

We can summarize that there is a conceptual difference between the two versions of the STEBR algorithm. The implementation that runs in time $O(N^2)$ scans the queue from head to tail and concerns which cells *to discard* whenever required. The linear-time implementation uses the fact that a cell having DBS within some $[T_k, T_{k+1})$ can be assigned not only service slot k but any of 0, 1, ...k-1 as well. This means that a cell that is not assigned slot k continues to contend on slots k-1, k-2, ... until it is assigned a slot or Slot 0 is reached. Therefore, the algorithm scans the queue from tail to head and makes decision regarding which cell *to service* at every slot.

b. *Example of Operation*

We demonstrate the conceptual operation of the linear-time implementation of the STEBR algorithm using the example of Figure V.11 and the source costs of Table V.1. In Figure V.19 we reproduce the example, together with the assigned costs of the queued cells. Table V.6 details the contending cells and their source costs throughout the operation of the algorithm, and the source that is assigned each slot. For every slot, the relevant costs are those of the oldest cells in the list of contending cells; they are bolded in Table V.6. Slot T_0 is assigned to Source 2, thus the first cell from this source (2_1) is chosen for service.

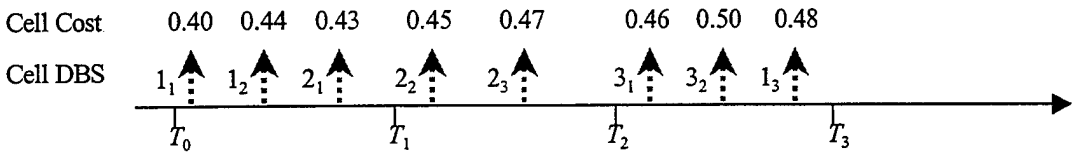


Figure V.19: Linear-Time STEBR Example; Cell Deadlines and Cell Costs

Slot	Contending Cells	Cost of Contending Source			Slot Assigned to Source
		Source 1	Source 2	Source 3	
T_2	3_1,**3_2,1_3**	0.48		0.50	3
T_1	2_2,**2_3,3_1,1_3**	0.48	0.47	0.46	1
T_0	1_1,**1_2**,2_1,2_2,**2_3,3_1**	0.44	0.47	0.46	2

Table V.6: Example of Operation of Linear-Time STEBR Algorithm

c. *Database*

The following variables constitute the database maintained by the algorithm:

- $A[i]$: Array; entry $i \in \{S_S\}$ contains the number of cells arrived thus far from source i.

- *DS*[*i*]: Array; entry $i \in \{S_S\}$ contains the number of cells discarded thus far by the server from source *i*.

- *CLPR*[*i*]: Array; entry $i \in \{S_S\}$ contains the CLPR of source *i*.

- *Cost*[*i*]: Array; entry $i \in \{S_S\}$ contains the modified value assigned to *CLPR*[*i*] if an additional cell from source *i* is discarded.

- *Extra_Cells*[*i*]: Array; entry i ($i \in \{S_S\}$) contains the extra number of cells from source *i* that are not included in the scheduling of the current slot, but may be included later.

- *Cell_Cost*[*j*]: Array; entry j ($0 \leq j \leq N_L$-1) contains a value (cost) assigned to the cell indexed *j* in the queue.

- N_{SS}: Variable; contains the maximum number of cells in the queue (out of N_L) that can be scheduled on time.

- *L*: Variable; contains the index of the latest slot that can be allocated for transmission. Its value is determined by the cell having the largest deadline.

d. Algorithm

All database elements other than *Cost*[*i*] are initialized to zero. Entries *Cost*[*i*] (for every $i \in \{S_S\}$) are set to some negative value (e.g., -1). Every time a cell from source *i* arrives at the queue, the cell is enqueued such that the queue remains sorted in a non-decreasing order of deadlines. Additionally, *A*[*i*] is incremented. Then, at every service slot, the algorithm summarized in Figure V.20 is carried out.

1. **For every** $i \in \{S_S\}$:
   A. **Set** $Cell\_Cost[i] = CLPR[i]$.
2. **Scan** the queued cells from head to tail. **For every** cell (say, from source i):
   A. **Calculate** the ToE of the cell.
   B. **If** the ToE of the cell is smaller than $1/C_W$ (the service time):
      i) **Discard** the cell.
      ii) **Increment** $DS[i]$.
   C. **Else**:
      i) **Calculate** $Cell\_Cost[i] = Cell\_Cost[i] + (A[i] \times ACLP[i])^{-1}$.
3. Suppose that N_L cells are left in the queue after Step 2. **Sort** the remaining cells in the queue in a non-decreasing order of their ToE.
4. **Set** $N_{SS} = 0$.
5. **Calculate** $L = \lfloor (\text{Deadline of last cell in the queue} - T_0) \times C_W \rfloor$.
6. **For** $k = 0$ **to** L:
   A. **Find** the set of cells having DBS within $[T_k, T_{k+1})$, $\{G_k\}$, and their number, $|\{G_k\}|$.
   B. **Calculate** $N_{SS} = N_{SS} + \min\{k+1-N_{SS}, |\{G_k\}|\}$.
7. **If** $N_{SS} = N_L$:
   A. **Service** the first cell in the queue.
8. **Else** ($N_{SS} < N_L$):
   A. **For** $k = L$ **downto** 1:
      i) **For every** cell $j \in \{G_k\}$ (say, originated by source i):
         a) **If** $Cost[i] \geq 0$:
            (1) **Increment** $Extra\_Cells[i]$.
         b) **Else** ($Cost[i] < 0$):
            (1) **Set** $Cost[i] = Cell\_Cost[j]$.
      ii) **Find** the largest $Cost[i]$ ($i \in \{S_S\}$). Suppose that it belongs to source i_x.
      iii) **If** $Cost[i_x] \geq 0$ (a valid source is obtained):
         a) **If** $Extra\_Cells[i_x] = 0$:
            (1) **Set** $Cost[i_x] = -1$.
         b) **Else** ($Extra\_Cells[i_x] > 0$):
            (1) **Decrement** $Extra\_Cells[i_x]$.
            (2) **Calculate** $Cost[i_x] = Cost[i_x] - (A[i_x] \times ACLP[i_x])^{-1}$.
   B. **Service** the first cell in the queue from source i_x.

Figure V.20: STEBR Allocation in Time O(N) in a Wireline System

3. Remarks

Scheduling in communication networks can be divided into two categories: off-line and on-line scheduling. In *off-line* systems, it is assumed that all arrival instants into the queue and their corresponding sources are known; scheduling algorithms utilize this information for their decisions. Because of the non-causal nature of the scheduling problem at hand, it is obvious that solution to the off-line problem can only be used for comparison between a proposed algorithm and an ideal one. In *on-line* systems, scheduling decisions are made at every service instant based, in general, on the history of cell arrivals and the prediction of upcoming cells. Algorithms that solve the on-line problem are supposed to run in servers of dynamic systems, in which cells arrive into the queue between decision instants.

The on-line problem, as in our case, does not feature an optimal substructure. The reason for that is the dynamics of the system: between decision (service) times, cells arrive at the queue and change the costs of the sources on line. Therefore, an algorithm that optimally solves the scheduling in real-time systems does not exist.

The STEBR scheme supplies the optimal scheduling for a 'semi-on-line' case, where in every service slot it is assumed that no more cells enter the queue from that point and on. The decisions the algorithm makes, based on the instantaneous occupancy of the queue and the previous values of the number of cell arrivals and discards, are thus only *locally* optimal. Consequently, overall locally-optimal decisions may result in a suboptimal *global* decision.

Using *a priori* knowledge of the statistics of the sources, one may predict future arrivals of cells from the sources and improve the decision on which cell to serve at each decision time. One may predict, for example, the number of arrivals from each source within the next n service slots to make a decision. Use of prediction as part of the scheduling process is beyond the scope of this work and left for future research.

I. SIMULATION RESULTS

In this section, we discuss the simulation results of the scheduling algorithms for the wireline case. Let N_S, N_V, and N_D be the number of input streams into a single-queue single-server system of type speech, video, and data, respectively. The objective here is to find the boundary of the admissible region and the server normalized throughput.

The admissible region consists of a three-dimensional volume (see Figure II.16) and requires a very large number of simulations in order to obtain its shape completely. In order to reduce the computational effort, our experiments first test two-dimensional cuts of the admissible region, for example, as functions of N_S and N_V, where the value of N_D is held constant. These cuts are then interpolated to construct the entire admissible region. Figure V.21 shows the two-dimensional cut for $N_D = 0$. The results of different algorithms are presented using different levels of gray. Each strip of gray indicates the performance improvement over the preceding algorithm. The STEBR is shown to outperform all other algorithms. By comparing it with the STE scheme, roughly an additional speech connection is allowed for every value of N_V.

Figure V.22 presents thick green plots of the normalized throughputs of the schedulers over the boundaries of the admissible regions. The STEBR algorithm yields a maximum normalized throughput of almost 90% and improves the server throughput over that of the STE by about 4%[11] as can be seen in Figure V.22. For a given scheduler, the maximal value of server throughput is achieved on the boundary of the admissible region; thus, Figure V.22 presents the maximum normalized throughput on that boundary. The maximum theoretical value of the normalized throughput (which can only be obtained for CBR sources using mean-rate allocation), is indicated in Figure V.22 as a yellow curve. The BCLPR algorithm demonstrates a slight improvement over the STE while the performance of the simple static allocation falls behind (normalized throughput

[11] To demonstrate this improvement by scaling the channel capacity up to an OC3 ATM link (155.52 Mbps) carrying 64-kbps digital speech (PCM) streams, the STEBR algorithm accommodates an extra bit rate of 6.2208 Mbps, which is equivalent to 97 more PCM connections.

of about 75%) mainly because of its lack of heterogeneous-class multiplexing gain. The peak-rate allocation exhibits the worst (smallest) region as expected.

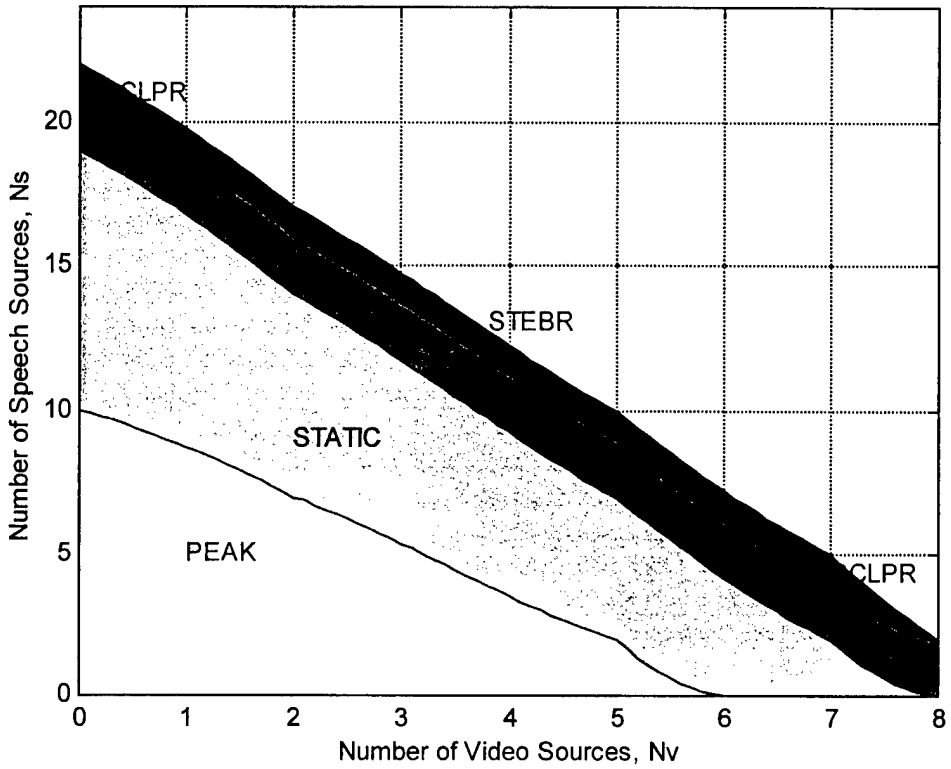


Figure V.21: Cut of the 3D Admissible Region at Zero Data Connections; $N_D = 0$

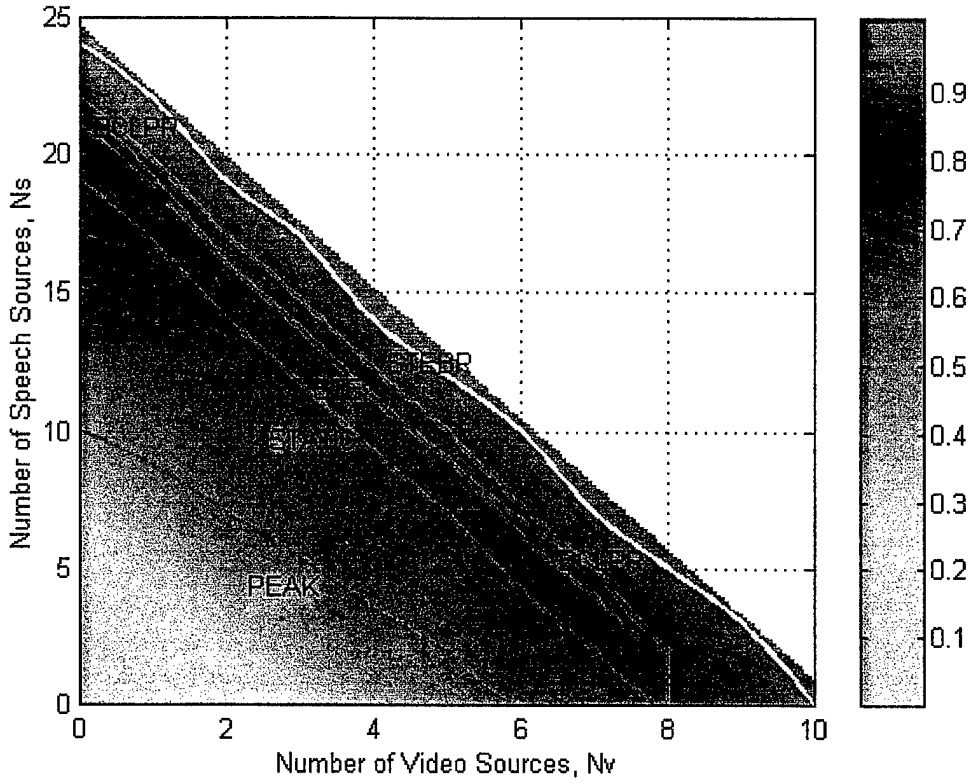


Figure V.22: Normalized Server Throughput at Zero Data Connections; $N_D = 0$

Figure V.23 illustrates a cut of the admissible region for $N_D = 100$. The relative sizes of admissible regions for the various algorithms in this plot are similar to those shown for $N_D = 0$ in Figure V.21. Nevertheless, the absolute differences in performance of the different schemes are somewhat smaller than those in Figure V.21. This is attributed to the reduction in available capacity for servicing speech and video calls due to the existence of 100 data sources. In Figure V.23, no region is plotted for the peak-rate allocation since the algorithm fails to satisfy any of the combination of N_S and N_V.

Figure V.24 presents the normalized throughput for the case of 100 data sources. The improvement of STEBR over STE regarding server throughput varies on the boundaries of their admissible regions between 0 and 4 percent although both techniques (as well as the BCLPR) present normalized throughputs that are greater than 93% throughout the cut boundaries. STEBR achieves the maximum possible admissible region

and normalized throughput as obtained by mean-rate allocation (the mean-rate allocation plot is not shown), other than in a single working point ($N_1 = 6$, $N_2 = 5$), where mean-rate allocation allows one more speech conversation.

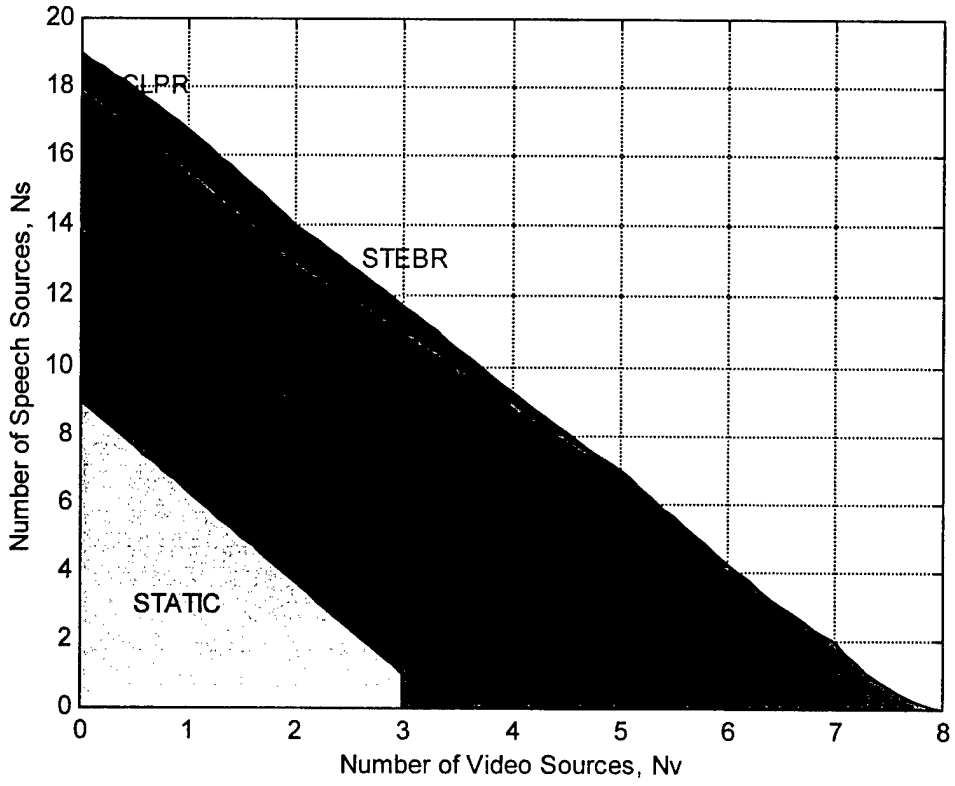


Figure V.23: Cut of the 3D Admissible Region at 100 Data Connections; $N_D = 100$

The maximum number of data sources in the admissible region (i.e., the value of N_D for the case $N_S = N_V = 0$) has been investigated as well. The results for different scheduling algorithms are presented in Table V.7. The normalized server throughputs, computed using Equation (V.1) and verified by simulation results, are also given in this table. The differences in performance become very small while all schemes (other than the peak-rate allocation) exhibit very high throughputs. This could be attributed to the small arrival rate of the data source; this permits the admission of a large number of sources into the system leading to a large multiplexing gain in the buffer. This means that

the mean capacity required by the server for each source approaches (from above) the mean arrival rate of the source.

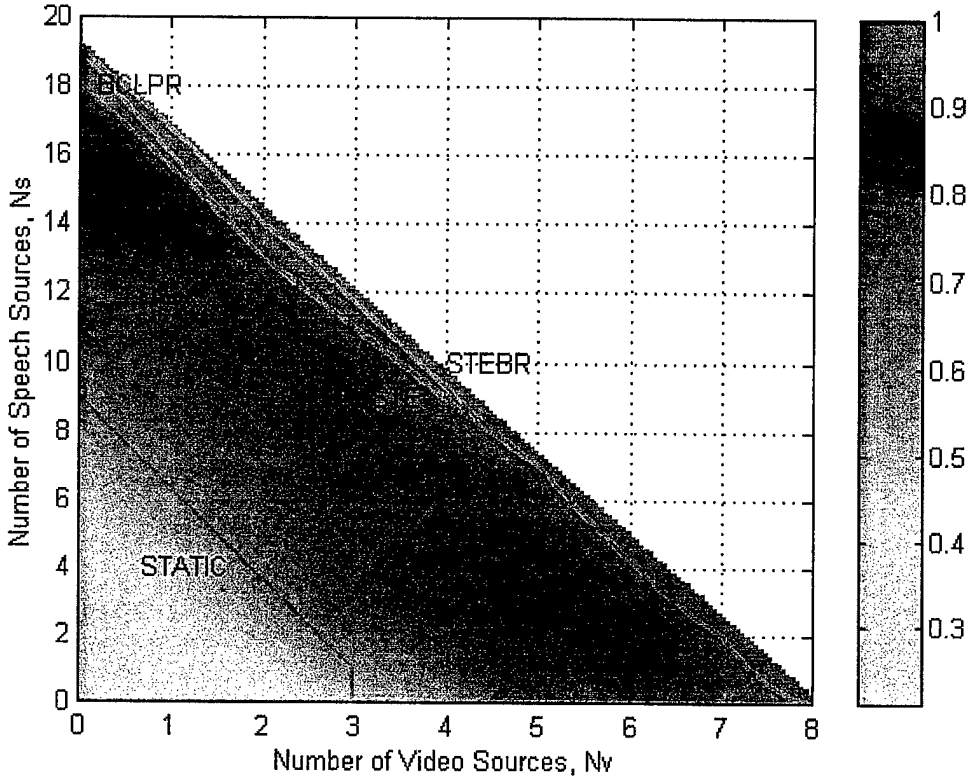


Figure V.24: Normalized Server Throughput at 100 Data Connections; $N_D = 100$

Scheduling Algorithm	Maximum value of N_D (Admissible Region Point (0, 0, $N_D\|_{max}$))	Normalized Server Throughput
Peak Allocation	18	0.0378
Static Allocation	450	0.9450
STE	475	0.9975
BCLPR	475	0.9975
STEBR	475	0.9975

Table V.7: Intersection of the Admissible Region with Axis N_D

Interpolation of the three-dimensional admissible region using the results presented so far does not clearly illustrate the differences among the schemes (the volumes are too close to each other). For a more meaningful illustration of results, Figure V.25 presents the number of admissible data sources as a function of the number of speech conversations and video sources. The figure shows the extra number of data sources that the STEBR scheduling algorithm could admit over the STE algorithm. The scale of the extra data sources admitted into the buffer is shown as a color bar on the right-hand side in Figure V.25. The advantage of the STEBR scheme over the STE is more visible as the number of speech and video sources increase, indicating a larger multiplexing gain achieved by STEBR for heterogeneous traffic comprising all three classes. The figure is believed to present the essence of performance improvement achieved due to the contributions made in this work to channel-allocation algorithms.

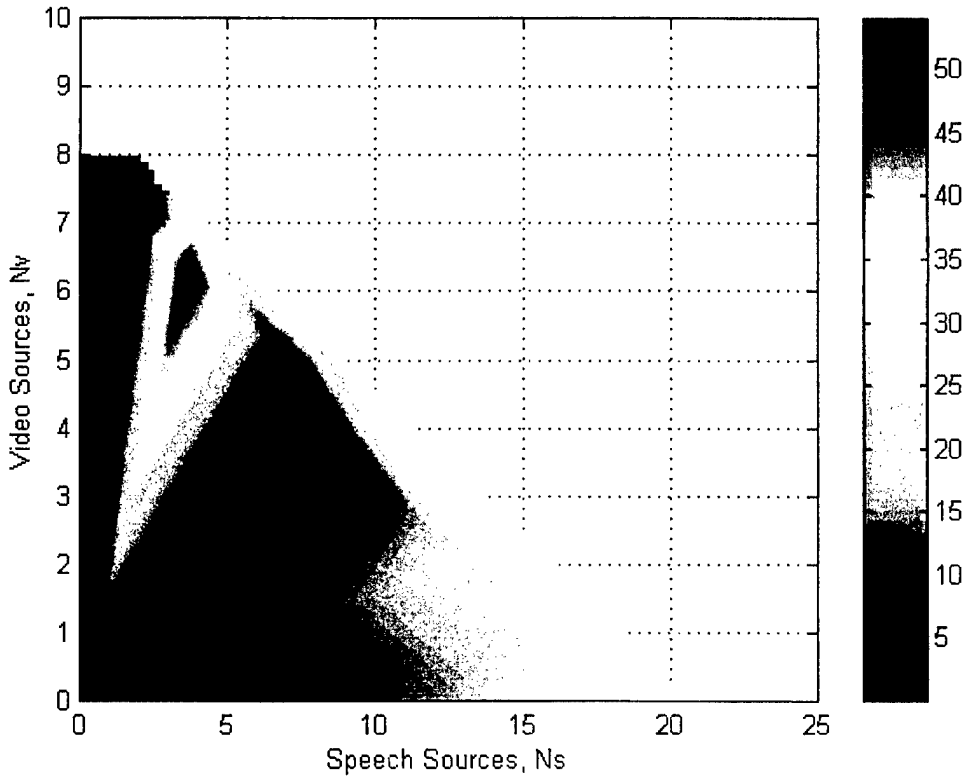


Figure V.25: Additional Data Sources Admitted using STEBR over STE

In this chapter, we have considered the problem of scheduling in wireline integrated services networks. A representative node in such networks consists of a single queue filled by cells from active sources, and a single server obeying a scheduling policy. Several existing scheduling policies were summarized and new algorithms proposed. The static-allocation algorithm assigns fixed pre-determined capacity to the sources; an analytical scheme to obtain the required capacity for homogeneous sources based on the Markov-chain characteristics of their class was provided. The STE algorithm bases its decisions on the cell deadlines only. The BCLPR algorithm ignores the cell deadlines completely; it makes service decisions using only the ratios between loss experienced by the sources and their allowed loss. The STEBR algorithm, proposed here for the first time, utilizes the advantages of STE and BCLPR. Cells are scheduled for service according to their deadlines as in STE, unless loss is expected in the future using this policy; then, the loss experienced by the sources thus far is taken into account to achieve an overall least-cost decision. A proof showing that STEBR makes an optimal decision at each service slot, given that no information about future cell arrivals is available, has been provided. Simulation results were shown to support this theory. Using the traffic classes described in Chapter IV, the admissible region obtained by STEBR is larger than the regions produced by other algorithms. In other words, for a given channel capacity, STEBR admits more sources. STEBR also yields normalized channel throughput, which is larger by up to 4% compared to STE.

VI. CHANNEL ALLOCATION IN MOBILE NETWORKS

In a wireless medium, the network may be represented as a system of distributed queues, where the service times at the remote stations are controlled by a central station (the CP). This chapter aims to design and implement the scheduling schemes, developed in the previous chapter, for use in the mobile channel. We follow the system architecture presented in Chapters II and III, using a TDD-based MAC protocol. In this chapter, we adapt the algorithms discussed in Chapter V for use over the wireless channel, complying with the outlines of the MAC. The traffic classes proposed in Chapter IV are then used in the next chapter to test the performance of the adapted algorithms over the mobile network.

In developing mobile schedulers, we divide the discussion into two parts according to the status information transmitted by the remotes and made available at the CP. In Section B, we consider the case of *partial* information received at the CP as piggyback data included within the cells or via contending control messages. Reception of the status report at the CP in this case is not guaranteed and its amount limited; the overhead required to acquire remote status is low. In Section C, on the other hand, we address the issue of *complete* status information. In every frame, each remote having at least one non-empty source is obligated to transmit some information about all its sources (e.g., number of waiting cells from the source). The complete status report may be crucial for scheduling algorithms that are sensitive to the "perfect" knowledge of all the sources' status. In both cases, channel allocation at the CP is determined frame-by-frame. That is, information slots for the entire frame are allotted at the beginning of the frame, rather than at each slot as in the wireline case. This behavior may cause discrepancies in simulation results, thus Section D is devoted to discussion about possible simulation errors due to the mobile-access-control architecture.

A. INTRODUCTION

Here we introduce the terms and the notation used in the scheduling schemes for the wireless medium. The mobile system includes the CP (numbered Station 1) and $|\{S_N\}|$ -1 registered remotes (numbered Stations 2, 3, ..., $|\{S_N\}|$). From a queueing point of view, the queues within the mobile network are distributed and can be represented as shown in Figure VI.1. The traffic from sources enters the system, and the output of the queues reaches the final or intermediate destinations. Each remote station $i \in \{S_N\}$ multiplexes traffic from n_i local active sources. The CP, on the other hand, multiplexes traffic from the local sources as well as traffic that arrives from external or remote sources, which may be destined to other remotes (external source-to-remote or remote-to-remote connections). We refer to these additional sources of traffic at the CP as *secondary sources*. The multiplexed sources in station i are numbered from 1 to n_i, where at the CP, n_1 includes the external sources as well. Thus, $|\{S_S\}|$, the total number of *incoming* traffic streams into the mobile network is given by:

$$|\{S_S\}| = \sum_{i=1}^{|\{S_N\}|} n_i .$$

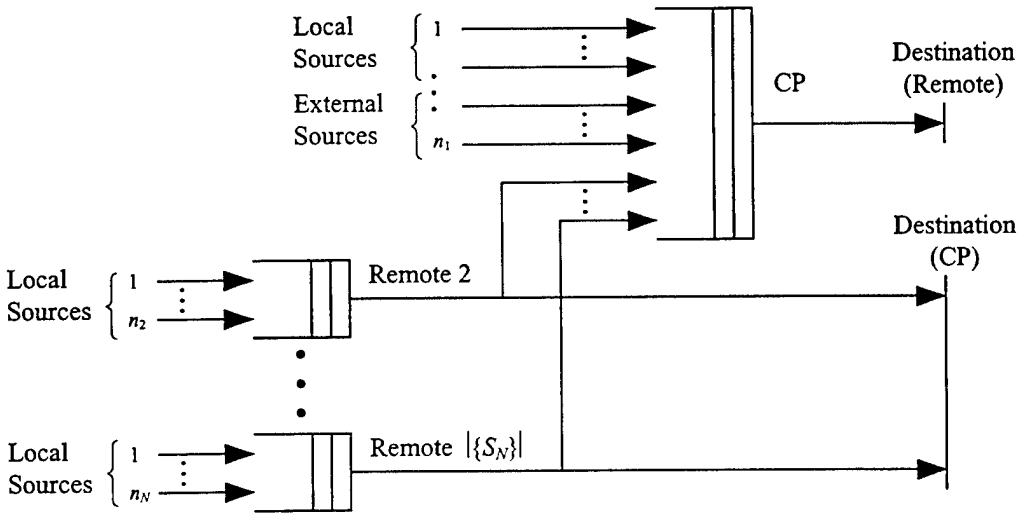


Figure VI.1: Distributed-Queues Representation of the Mobile Network

Figure VI.2 presents a closer look at the remote's queueing system. The secondary MAC (SMAC) in a remote functions as a dependent server of the remote's queue; a queued cell can be serviced only when a slot is allocated by the primary MAC (PMAC) at the CP. This constraint is marked in Figure VI.2 as a dashed line, controlling the outgoing stream from the queue.

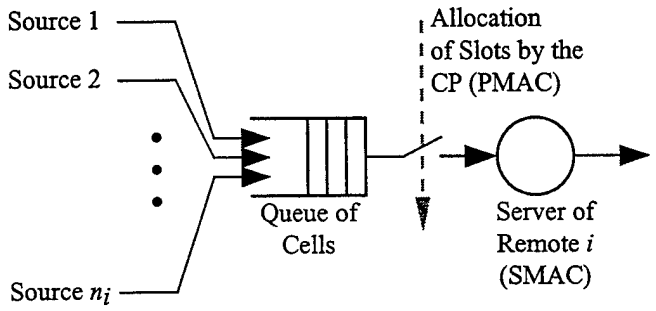


Figure VI.2: Multiplexing n_i Sources at Remote Station i

Throughout this chapter, we denote the number of information slots in the MAC frame as N_{IS}, and the capacity of the mobile channel as C_M (in cells/sec or slots/sec).

B. SCHEDULING BASED ON PARTIAL REMOTE STATUS

In this section, we consider the performance of the scheduling schemes in the mobile network when only partial information on the status of a remote queue is available at the CP. The purpose is to reduce the overhead of the remote status to the minimum possible. In the next section, the same algorithms are analyzed when complete information on the remote queue status is available. While the algorithms are expected to work more efficiently in the latter case, the extra overhead in carrying remote information might not make it worthwhile. This will be investigated for each allocation technique in the next chapter.

We propose schemes for operation of the scheduling algorithms over the wireless channel. The algorithms require some modifications in relation to the wireline case; status information from the remotes is crucial for efficient implementation. These issues

are thoroughly addressed here. Before proceeding with the scheduling schemes, we first discuss multiple-hop connections, commonly found in the mobile network as remote-to-remote calls.

1. Multiple-Hop Connections

Connections within a mobile integrated services network can have multiple hops, where a source has no direct connection (physical or logical) to its destination. In the network under study, for remote-to-remote connections, even if the radio transceivers of the remotes are within communication range, the information passes through the CP before being relayed to the final destination(s).

Cells in a multiple-hop connection experience queueing delays in the originating station as well as in each of the intermediate nodes along the source-destination path. The QoS requirements of such a connection are defined end to end. As a result, the connection requires n capacities (C_1, C_2, ..., C_n) corresponding to the number of queues along the end-to-end path as shown in Figure VI.3.

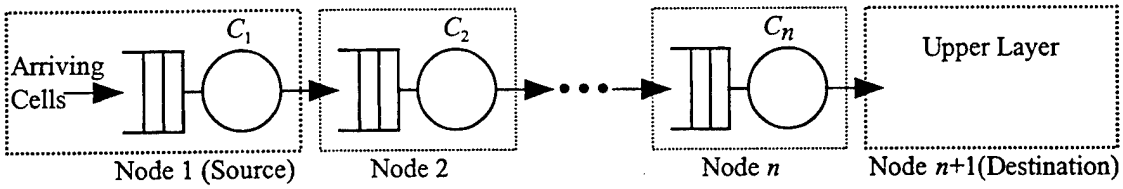


Figure VI.3: Queueing Model of a Multiple-Hop Connection

The problem here is to impose the end-to-end constraints ($CLP_{\text{end-to-end}}$ and $maxCTD_{\text{end-to-end}}$) on every queue along the source-destination path in an optimal fashion such that

$$\sum_{i=1}^{n} CLP[i] \leq CLP_{\text{end-to-end}}$$

$$\sum_{i=1}^{n} maxCTD[i] \leq maxCTD_{\text{end-to-end}},$$

where $CLP[i]$ and $maxCTD[i]$ are the loss and delay, respectively, designed for the i^{th} segment ($1 \le i \le n$).

An analytical solution that accommodates both CLP and maxCTD requirements in a multiple-hop connection does not exist at this time [81]. Adopting the solution proposed by [99] for a Poisson-arrival, non-constrained, two-hop connection, the end-to-end limitations can be equally divided among the queues in order to obtain the same required capacity for each node:

$$CLP[i] = \frac{CLP_{\text{end-to-end}}}{n} \qquad 1 \le i \le n$$

$$maxCTD[i] = \frac{maxCTD_{\text{end-to-end}}}{n} \qquad 1 \le i \le n.$$

In particular, in the remote-to-remote connection case, we set the requirements over the remote-to-CP and CP-to-remote segments to be one half of the end-to-end CLP and maxCTD requirements. Once an analytical solution for the division of the QoS requirements along the end-to-end path becomes available, the new capacities can be easily obtained. This issue is further discussed in Section VIII.B as possible future work.

2. Static Allocation

The static channel allocation is based on the results obtained for a wireline ATM multiplexer. Station i (see Figure VI.1) requires a minimum channel capacity in order to support the n_i sources that are multiplexed in this station. The set of capacities required by all stations in the network governs the way in which the CP needs to allocate the information slots to the sources, in order to satisfy the QoS requirements for all. We use the (known) required capacities obtained based on the wireline multiplexer for allocation of slots on the uplink and downlink information subchannels.

In the static-allocation scheme, the uplink control subchannel *is not* used to request the channel since a constant capacity is allocated to each active source, regardless of its instantaneous traffic-generation behavior and/or its occupancy. Thus, the size of the uplink control subchannel is small compared to other schemes that make use of it to request channel allocation thereby improving the MAC performance. (Nevertheless, a

few control slots are still needed for mobile signaling messages, hence the performance improvement is minor.)

We use the following notation to describe the operation of the algorithm:

- $CRC[i]$: Required capacity by source $i \in \{S_S\}$ in cells/sec.
- CRC_T: Total required capacity by all sources in cells/sec.
- $CRS[i]$: Required capacity by source $i \in \{S_S\}$ in information slots/frame.
- F: Number of MAC frames per second.
- $Z[i]$: Counter for static allocation of source $i \in \{S_S\}$.

a. Policy at the CP

When a new source is activated or released, the number of secondary sources at the CP may be affected as well. The PMAC scheduler must consider the number of secondary sources since they occupy a portion of the channel capacity for proper operation. For example, if a source is originated at the CP, then only the CP will have a modified required capacity. On the other hand, if a source is activated at a remote with another remote as destination, then both the originating remote and the CP will have new required capacities. The stations that have modified capacity requirements due to admission or release of a source are denoted as the *affected stations*.

During the process of static allocation, the actual required capacities by the sources (in slots per frame) are multiplied by a constant factor. The factor is equal to the ratio between the number of information slots in a frame (N_{IS}) and the number of information slots required by all existing sources; the multiplication allows the use of all the slots available in the MAC frame. This avoids waste of slots in the frame that are not allocated to sources. All the available slots are allocated to sources, even if fewer slots are required by the active sources. This leads to an improvement in the QoS experienced by the sources, whenever the channel load is smaller than the available capacity, over the case where only the minimum capacity is allocated.

The static-allocation algorithm is summarized in Figure VI.4.

Initialization:

1. **Calculate** $C_M = F \times N_{IS}$.
2. **Set** $CRC[i]$ and $CRS[i]$ of all sources (and secondary sources) in the system to 0.
3. **Set** $Z[i]$ of all possible sources (and secondary sources) in the system to $-\infty$.

Every time a source j becomes active:

4. **Calculate** $CRC[i]$, the required capacity for every source i within the set of affected stations using the scheme proposed in Section V.C.
5. **Calculate** CRC_T, the total required capacity by all traffic sources:
$$CRC_T = \sum_{i \in \{S_S\}} CRC[i].$$
6. **Calculate** for every $i \in \{S_S\}$:
$$CRS[i] = CRC[i]/F.$$
7. **Calculate** for every $i \in \{S_S\}$:
$$CRS[i] = CRS[i] \times \frac{C_M}{CRC_T}.$$
8. **Set** $Z[j]$ to 0.

Every time a source j is released:

9. **Set** $CRS[j]$ to 0.
10. **Repeat** Steps 4, 5 ,6, 7.
11. **Set** $Z[j]$ to $-\infty$.

Every time a frame is to be transmitted by the CP:

12. **Discard** local cells whose ToE is smaller than the time it takes for the CP to transmit one cell on the downlink channel.
13. **Calculate** for every source $i \in \{S_S\}$:
$$Z[i] = Z[i] + CRS[i].$$
14. **Repeat** N_{IS} times:
    A. **Find** the maximum $Z[i]$ ($i \in \{S_S\}$), say, $Z[j]$.
    B. **Allocate** an information slot to source j.
    C. **Decrement** $Z[j]$.

Figure VI.4: Static Allocation with Partial Remote Status (PMAC)

The algorithm relies on the fact that the admission controller *does not* admit more calls to the network than the scheduler is capable of handling. That is, an admission decision is based on recalculation of CRC_T by taking the new call into account; the call is admitted if the new CRC_T satisfies

$$CRC_T \leq C_M$$

or rejected otherwise.

We wish to remark on some properties of the algorithm. Since we have purposely increased the required capacities to utilize all the available slots in the frame, the values of the source counters $Z[i]$ fall within [-1, 1] immediately after the slot allocation in a frame. If the total required capacity, CRC_T, is greater than C_M, the values of $Z[i]$ are expected to grow without bound since the network cannot satisfy the requests for channel allocation within a finite period.

Example: Assume that, at some point in time, five sources are active with the following capacity requirements: 10, 12, 24, 26, and 28 cells/sec. The channel capacity is given to be 200 cells/sec, and there are 20 frames/sec (i.e., 10 cells/frame or slots/frame). Since the channel capacity is twice as large as the total capacity requirement by all sources (100 cells/sec), the actual capacity requirements are multiplied by 2 to receive $CRS[i]$ of 1, 1.2, 2.4, 2.6, and 2.8 slots/frame for Sources 1 through 5, respectively. Assuming that initially $Z[i]$ is zero for all i, the allocation of slots in the first two frames is given in Table VI.1.

Frame	Stage	Allocation Counters					Allocation for Sources on Frame
		Z[1]	Z[2]	Z[3]	Z[4]	Z[5]	
	Initialization	0	0	0	0	0	
1	Before allocation	1	1.2	2.4	2.6	2.8	
	After allocation	0	0.2	0.4	-0.4	-0.2	5, 4, 3, 5, 4, 3, 2, 1, 5, 4
2	Before allocation	1	1.4	2.8	2.2	2.6	
	After allocation	0	0.4	-0.2	0.2	-0.4	3, 5, 4, 3, 5, 2, 4, 1, 3, 5
....							....

Table VI.1: Static Allocation at the PMAC, Example of Operation

b. Control-Subchannels Size Adjustment

The MAC frame is normally set for default values for the four subchannels within its frame (see Chapter III). If all slots are allocated for transmission of cells, the sizes of the *control* subchannels need to be adjusted. This is required in case the sizes of the downlink and uplink *information* subchannel (N_{ID} and N_{IU}, respectively) are not equal to their default values. The sizes of the downlink and uplink *control* subchannels (N_{CD} and N_{CU}, respectively) are *adjusted* to achieve a total frame size of 12 milliseconds at most. If there are more downlink information slots than the default, more control slots may be added to the frame. In the opposite situation, there is a deficit of time in the frame that should be compensated for by reducing the sizes of the control subchannels as necessary. This means that momentarily the signaling channel responds in the mobile channel more slowly (e.g., slower registration process, slower admission decision response, etc.). The minimum value of N_{CD} is 2 to ensure proper operation of the network (acknowledging control messages and information cells transmitted on the previous frame). The part of the algorithm that adjusts the sizes of the control subchannels, common to all scheduling algorithms operating over the mobile channel, is summarized in Figure VI.5.

1. **If** $N_{ID} = N_{ID}|_{\text{DEFAULT}}$ and $N_{IU} = N_{IU}|_{\text{DEFAULT}}$:
   A. Do nothing.
2. **Else if** $N_{ID} > N_{ID}|_{\text{DEFAULT}}$ (there is unused time in the frame):
   A. **Increase** N_{CU} as much as possible.
3. **Else** ($N_{ID} < N_{ID}|_{\text{DEFAULT}}$, i.e., there is a shortage of time in the frame):
   A. **Set** $N_{CD} = 2$.
   B. **Set** N_{CU} to the maximum possible value up to the limit of a 12-millisecond frame.

Figure VI.5: Size-Adjustment Algorithm for the Control Subchannels

c. Policy at the Remote

The static-allocation policy at the remote is summarized in Figure VI.6.

1. **Schedule** an invocation at the time allocated by the CP for transmission of a cell from one of the remote's sources.
2. At the scheduled time, **scan** the local queue and
   A. **Discard** cells with deadlines smaller than the transmission time of one cell on the uplink (for remote-to-CP connections) or smaller than the minimum time required for the CP to relay one cell on the downlink (for remote-to-remote connections).
   B. **If** a cell from the source receiving the allocation is found:
      i) **Transmit** the first cell from that source.
   C. **Else**:
      i) **Transmit** a cell from the queue having the smallest ToE.

Figure VI.6: Static Allocation (SMAC)

d. Remarks

The static-allocation scheme is based on traffic characteristics known *a priori*. Since only the statistical behavior of the arrival processes rather than the exact arrival instants are known, we have found that the static-allocation scheme provides lower performance than that achieved by a dynamic technique. In other words, the probability that a slot for transmission has been allocated to a remote, but has not been used, is larger.

In order to decrease this waste of potential information slots and increase the MAC throughput, we wish to apply dynamic scheduling schemes that take into account the instantaneous occupancy of the queues at the CP and in the remotes. Such schemes, however, require some knowledge of the number of waiting cells for each MVC and/or their deadlines, in order to accurately allocate no more than the amount of slots required.

From Chapter III, the MAC protocol supports two types of messages, which include special fields to indicate the status of the transmitting remotes (cells and ALLOCATE_REQUEST control messages). The dynamic allocation schemes make use of these fields to keep track of the instantaneous bandwidth requirement of the active sources, thus attempting to minimize the number of wasted slots. If an information slot is allocated to a source, but there are no cells to transmit within that source, the corresponding remote looks for another source with a waiting cell and uses the slot to transmit this cell.

3. STE Allocation

a. Scheduling with Transmission Deadlines

In Chapter V, a class of schemes that make use of the transmission deadlines has been considered. The shortest-time-to-extinction (STE) policy was the scheme proven optimal for the discrete-time G/D/1 queueing system having a unit service time. The mobile network operates under different conditions. The service time is constant; however, the sources do not necessarily generate cells at integer multiples of the service time. We consider the STE technique mainly because of its popularity among researchers in the field of mobile ATM [75].

For use in personal communication systems, Raychaudhuri and Wilson [75] compared the STE (which they denoted time of expiry (ToE)) and the FCFS approaches in mixed time-critical (voice-) and non-time-critical (data-) traffic scenarios. From simulation results, they showed that the ToE policy substantially improves the packet loss probability for time-critical traffic (for example, a reduction by a factor of 3 to 5 with normalized throughputs greater than 0.5). However, in order to achieve this performance improvement, time stamping must be incorporated in the PCN header, requiring reasonable clock synchronization between the base station and the remotes. In order to support the STE scheduling, the deadlines of the cells, which are known only at the remotes, need to be passed to the CP.

In the system under study, deadlines of cells within the remote queues can be passed to the PMAC at the CP in two ways: whenever a request for channel allocation is sent (indicating the deadline of the first cell from the source seeking allocation), and whenever a cell is sent (indicating its deadline). In both cases, a remote uses eight bits to inform the CP about the deadline of the cell in the following way: for speech and video sources, the field includes the deadline of the cell in milliseconds, and for data sources, the field includes the deadline of the cell in quarters of a second.

b. Concepts

Instead of passing the deadlines of all remote-waiting cells to the CP, we propose a different technique. Although the technique does not provide all the remotes' cell deadlines, it still allows the use of the STE scheme with its advantages. The basic idea is to collect at the CP the number of cells that must be serviced on the upcoming frame, rather than the exact values of their deadlines. If these numbers are positive, they are reported to the PMAC by each station that generates traffic (including the CP). At the beginning of every frame, the CP determines the number of cells (from all stations) that must be serviced on that frame; a cell that is not serviced is then expected to be discarded. For CP-to-remote and remote-to-CP connections, the report includes the number of cells per station to expire in the next *two* frames. Figure VI.7 demonstrates the need to consider two frames instead of one. Suppose that a remote has one cell to expire at time $T_{n+2}+\varepsilon$, a short time after the end of frame $n+1$ ($n = 1, 2, \ldots$). In frame n, the remote does not inform the CP that it has any cells to expire in frame $n+1$, thus the CP is not required to allocate a slot for it on the latter frame. However, if no allocation is made for this cell on frame $n+1$, then allocation on the next frame ($n+2$) becomes too late and the cell is discarded.

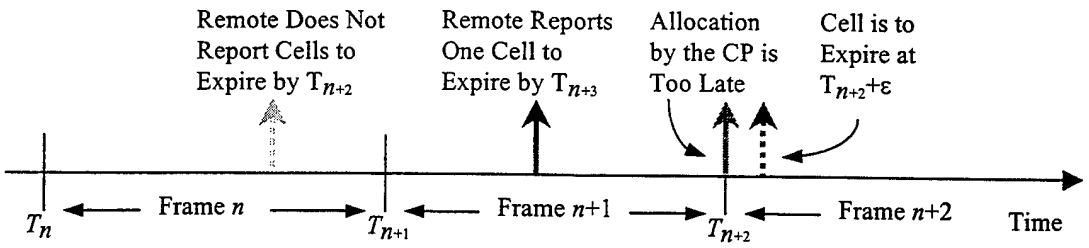


Figure VI.7: Cell Discarding in a Remote-to-CP Connection when the Remote Reports the Cells to Expire by the End of the Next Frame

Remote-to-remote connections require some extra consideration. If the originating remote transmits its deadline information in the same manner as described above, cells might be discarded even in a lightly-loaded network. In Figure VI.8, we demonstrate that by describing the flow of a single cell. Suppose that a remote source has one cell that expires at time $T_{n+2}+\varepsilon$, a short time after the end of frame $n+1$ ($n = 1, 2, \ldots$). The remote informs the CP in frame n that a cell is to expire within the next two frames. The CP then allocates on frame $n+1$ a slot to the remote, which transmits the cell. When the cell arrives at the CP, it needs to be relayed on the downlink; however, no sufficient time is available for this transmission, and the cell is discarded.

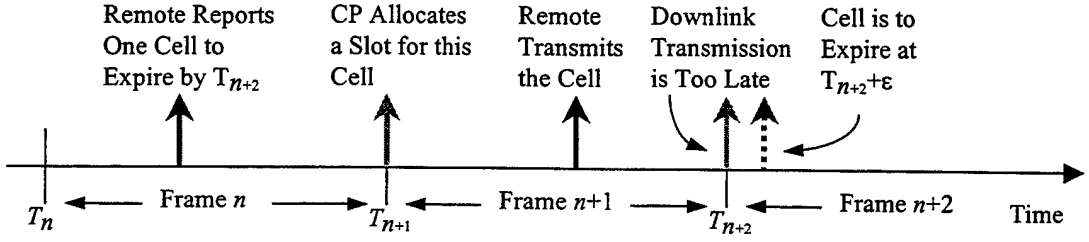


Figure VI.8: Cell Discarding in a Remote-to-Remote Connection when the Remote Reports the Cells to Expire by the End of the Next Two Frames

The proposed solution to this problem is as follows: if a remote is the source station in several remote-to-remote connections, it informs the CP about the number of cells to expire for all connections within the next *three* frames. (In remote-to-CP connections, the report still relates to the expiration of cells in the next two frames.)

This way, the CP has sufficient time to support the transmission of the cells on the downlink.

Following the discussion presented above, the CP obtains information about the number of cells that must be serviced on the upcoming frame from all sources throughout a frame. The CP uses this information at the beginning of the next frame to allocate slots according to the scheduling algorithm described below. The information about the number of required slots is collected by the CP in two ways. When a cell arrives at a remote's MAC with an empty queue, an ALLOCATE_REQUEST message is generated (see Chapter III). The remote utilizes the MATM-signaling-information field within this message to report the number of cells within the remote that must receive allocation on the next frame. The same information is also included by the remote when a cell is transmitted as a piggyback allocation request (see Chapter III). If the connection is of type remote to remote, the sending SMAC also includes its deadline information in the eight reserved bits of the cell header, as discussed earlier.

Unlike the static-allocation scheme, the STE policy is a distributed technique: the SMAC at the remote takes an active role in determining the sources to be serviced. When slot allocation is determined by the CP, each remote decides to utilize it for the transmission of the cell having the smallest deadline (this information is perfectly known locally). Thus, we split the discussion into two parts: one for the PMAC policy at the CP and the other for the SMAC scheme in the remotes.

c. Policy at the CP

The CP maintains a table of the number of cells that need to be serviced on the next frame (otherwise, they might be lost). Every active node (including the CP) has an entry in this table. Let entry j in the table be $RS[j]$ ($j \in \{S_N\}$), where $\{S_N\}$ is the set of active nodes. Let the number of available information slots (downlink and uplink) in one frame be N_{IS}.

Another table to be maintained at the CP is that of the mean CLP ratio (MCLPR) of every station. The MCLPR measures the average ratio between the actual

cell loss and the allowed cell loss of all sources within a node. Each entry in this table contains the value

$$MCLPR[j] = \frac{EC[j]}{E\{A[j]\}}, \qquad j \in \{S_N\}, \tag{VI.1}$$

where $EC[j]$ is the number of cells expired in node j as known at the CP (i.e., due to insufficient slot allocations by the CP), and $E\{A[j]\}$ is the mean amount of traffic that was supposed to arrive from that node from the time its connections have been admitted and to the present time. $EC[j]$ is increased only when the CP cannot allocate slots to a station, thus it knows for sure that a cell is to expire. The value of $E\{A[j]\}$ in Equation (VI.1) is calculated by

$$E\{A[j]\} = \sum_{i \in \{S_j\}} E\{\lambda_i\} \times (t - t_{0i}),$$

where $E\{\lambda_i\}$ is the mean arrival rate of source $i \in \{S_j\}$ (the set of active sources within node j), t_{0i} is the time at which i has been admitted into the network, and t is the current time. The actual number of cells discarded at remote $j \in \{S_N\}$ may differ from $EC[j]$; we nevertheless assume that, most of the time, the information regarding the number of cells to expire during the next frame is reported to the CP.

The STE algorithm is summarized in Figure VI.9.

1. **Discard** local cells whose ToE is smaller than the time it takes for the CP to transmit one cell on the downlink channel.
2. **Find** the number of local cells that must be serviced on the next frame.
3. **Calculate** RS_T, the total number of cells from all nodes that must be serviced on the next frame:
$$RS_T = \sum_{j \in \{S_N\}} RS[j].$$
4. **Calculate** the average number of arrivals so far from every station $j \in \{S_N\}$:
$$E\{A[j]\} = \sum_{i \in \{S_j\}} E\{\lambda_i\} \times (t - t_{0_i}).$$
5. **Calculate** $MCLPR[j]$ for every $j \in \{S_N\}$ using Equation (VI.1).
6. **If** $N_{IS} \geq RS_T$:
   A. **Allocate** $RS[j]$ information slots to a source in node $j \in \{S_N\}$ having $RS[j] > 0$.
   B. **Allocate** the remaining N_{IS}-RS_T slots (if any) to a source in the node having the largest MCLPR.
7. **Else** ($N_{IS} < RS_T$):
   A. **Repeat** RS_T-N_{IS} times:
      i) **Increment** $EC[j]$ of node $j \in \{S_N\}$ (with $RS[j] > 0$) having the smallest MCLPR.
      ii) **Calculate** $MCLPR[j]$.
      iii) **Decrement** $RS[j]$.
   B. **Allocate** the N_{IS} slots in the frame such that $RS[j]$ slots are allocated to a source within node $j \in \{S_N\}$ having $RS[j] > 0$.
8. **Set** $RS[j]$ to 0 for every $j \in \{S_N\}$.

Figure VI.9: STE Allocation with Partial Remote Status (PMAC)

d. Policy at the Remote

The proposed STE policy at the remotes relies on the fact that each remote has exact knowledge of deadlines for local cells waiting to be transmitted. The PMAC at the CP, on the other hand, has only partial knowledge of this information. Hence, its allocation is not optimal. In order to improve the performance of the network, the MAC operates in a distributed fashion, i.e., decisions regarding the cells to be transmitted are

made locally by each SMAC. The scheme gives priority to remote-to-remote connections. In comparison to single-hop connections, the latest time for transmission of waiting cells of two-hop connections is smaller. Thus, the SMAC services the cell with the smallest time to the latest transmission rather than the cell with the smallest deadline. ALLOCATE_REQUEST control message is sent only when a cell is to expire in the next two frames in a remote-to-CP connection or three frames in a remote-to-remote connection, rather than when the queue becomes non-empty. The STE policy at the remote is summarized in Figure VI.10.

1. **Schedule** an invocation at the time allocated by the CP for transmission of a cell from one of the remote's sources.
2. At the scheduled time, **scan** the local queue and
   A. **Find** the cell having the smallest time to latest transmission:
      i) For remote-to-CP connection, the time to latest transmission is the cell deadline minus the transmission delay of the cell on the uplink.
      ii) For remote-to-remote connection, the time to latest transmission is the period until the CP can relay at least one cell on the downlink.
   B. **Discard** cells having negative time to latest transmission.
3. **If** an eligible cell has been found:
   A. **Transmit** the cell.

Figure VI.10: STE Allocation (SMAC)

4. BCLPR Allocation

We now consider the implementation of the BCLPR algorithm for the wireless network. In order to support the BCLPR scheduling, one needs to pass the CLPR of the sources within the remotes to the CP. We propose a technique to supply all the information needed by the CP to implement the BCLPR algorithm over the wireless network. The CLPRs of the remote sources are obtained by the CP in two ways. When a cell arrives at the SMAC and finds there an empty queue, an ALLOCATE_REQUEST message is generated (see Chapter III). The remote utilizes the MATM-signaling-information field within this message to report the ICLP of the source and the number of

waiting cells within the remote. The same information is also included by the remote when a cell is transmitted as a piggyback allocation request (see Chapter III).

Now, we want to further detail the way the CLPR is represented at the CP and in the remotes. A CLPR is represented as a floating-point number in the range [0, 1]. Occasionally, the CLPR values may exceed 1, especially at the beginning of a connection lifetime during which discarding of any cell dramatically increases the CLPR. In order to be able to pass, from the remotes to the CP, very small (positive) as well as very large (positive) values in an 8-bit field, we use the following method:

- Using the allocated 8-bit field, only the value of the CLP is transmitted. The CP uses the allowed CLP, which is known to both sides (according to the connection class) to obtain the CLPR; this reduces the required range of floating-point values to [0, 1].

- Figure VI.11 shows the proposed 8-bit floating-point representation of a number in the range [0, 1], where M and E bits represent the mantisa and the exponent, respectively. The five bits of the mantisa correspond to the CLP value in a way that M_n represents 2^{n-5} for $0 \leq n \leq 4$. The exponent value is the negative power of 10. For example, the 8-bit representation 01100100 stands for a CLP value of $(0.25+0.125)\times10^{-4} = 0.0000375$. This technique allows one to represent a variety of CLP values: from a minimum positive value of $(1/32)\times10^{-8}$ (the minimum allowed CLP among the traffic classes considered here is 10^{-6}) to a maximum value of 0.96875.

M_4	M_3	M_2	M_1	M_0	E_2	E_1	E_0

Figure VI.11: An 8-Bit Floating-Point Representation of CLP

Once the CP receives the CLP values of the remote sources, it calculates the CLPR according to the type of the connection. For remote-to-CP and CP-to-remote connections, the CLPR is just the reported CLP divided by the ACLP. For remote-to-remote connections, the calculation is somewhat involved. Let $DR[i]$ and $DC[i]$ ($i \in \{S_S\}$) be the number of cells discarded from source i by the originating remote and by the CP, respectively. Let $A[i]$ be the total number of arrivals generated at the (remote) station

originating source i. Figure VI.12 illustrates the flow of information and possible locations of cells discards in a remote-to-remote connection.

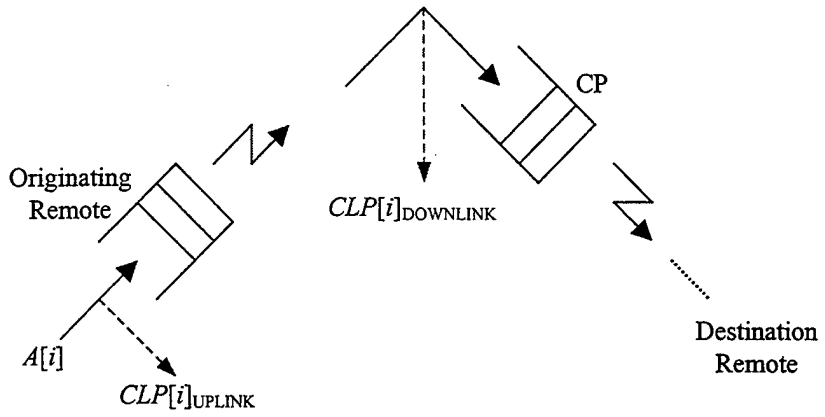


Figure VI.12: Flow of Cells and Possible Losses in a Remote-to-Remote Connection

Now, the losses in the two segments of the connection ($CLP[i]_{UPLINK}$ and $CLP[i]_{DOWNLINK}$) contribute to the total loss of connection i ($CLP[i]$) as follows:

$$
\begin{aligned}
CLP[i]_{UPLINK} &= \frac{DR[i]}{A[i]} \\
CLP[i]_{DOWNLINK} &= \frac{DC[i]}{A[i]-DR[i]} \\
CLP[i] &= \frac{DR[i]+DC[i]}{A[i]} \\
&= \frac{CLP[i]_{UPLINK} \times A[i] + CLP[i]_{DOWNLINK} \times (A[i]-DR[i])}{A[i]} \\
&= CLP[i]_{UPLINK} + CLP[i]_{DOWNLINK} - CLP[i]_{UPLINK} \times CLP[i]_{DOWNLINK} \\
&= CLP[i]_{UPLINK} + CLP[i]_{DOWNLINK} \times (1 - CLP[i]_{UPLINK}) \\
CLPR[i] &= \frac{CLP[i]}{Allowed\ CLP[i]}.
\end{aligned}
\qquad \text{(VI.2)}
$$

The exact value of $A[i]$ is unknown at the CP, and it needs to be estimated by counting the slots used by the source or by using the source's mean arrival rate.

As in STE, the BCLPR is also a distributed technique: the SMAC takes an active role in determining the sources to be serviced. In the following, we describe the different

mechanisms used at the CP and in the remotes in order to implement the BCLPR-allocation algorithm.

a. Policy at the CP

The CP maintains three arrays throughout the operation: $CLPR[i]$ indicates the instantaneous CLPR of source i; $AS[i]$ indicates the number of allocated slots to that source, which contains the approximate value of $A[i]$; and $W[i]$ marks the number of waiting cells from source i. Elements $W[i]$ are obtained by the CP from the local sources and from remote reports over the uplink control and information subchannels.

Two scheduling policies are possible for the case of remote-to-remote connections. Under Policy I, cells waiting at the CP to be relayed to a destination remote are given priority over cells from the originating remote since knowledge about the latter is unknown perfectly at the time of allocation. Under Policy II, on the other hand, priority is given to the originating remotes. From Equation (VI.2), in a remote-to-remote connection, there are allowed losses in both remote-to-CP and CP-to-remote segments. The former has a greater effect on the overall CLP; thus, we want to reduce it by allocating slots to the remote first (given that it has cells from this source). Simulation results indicate minor differences in performance between the two policies.

Prior to summarizing the BCLPR algorithm, we define *func*(i) as a set of operations on the array elements of source i:

func(i):

1. **Increment** $AS[i]$.
2. **Calculate** $CLPR[i]$ according to Equation (VI.2).
3. **Decrement** $W[i]$.
4. **If** $W[i] = 0$:
   A. **Set** source i as being ineligible for service.

The BCLPR algorithm is summarized in Figure VI.13.

1. **Discard** local cells whose ToE is smaller than the time it takes for the CP to transmit one cell on the downlink channel.
2. **Find** the number of waiting cells in the local queue and accordingly update $W[i]$. **Set** sources having $W[i] > 0$ as being eligible for channel allocation.
3. **Repeat** N_{IS} times:
   A. **Find** the source with maximum CLPR (say, i) among the eligible sources. If no eligible sources exist, pick a remote source (say, i) in a cyclic manner.
   B. **If** i is a local source:
      i) **Allocate** a slot on downlink for source i.
      ii) **Call** *func*(i).
   C. **Else** (a remote source):
      i) **If** the connection is of type remote-to-CP:
         a) **Allocate** a slot on uplink for source i.
         b) **Call** *func*(i).
      ii) **Else** (a remote-to-remote connection):
         Under Policy I:
         a) **If** a cell that has not been allocated a slot thus far from source i is found in the local information queue:
            (1) **Allocate** a slot on downlink for source i.
            (2) **Call** *func*(i).
         b) **Else** (no cells without allocated slots have been found in local queue):
            (1) **Allocate** a slot on uplink for source i.
            (2) **Call** *func*(i).
         Under Policy II:
         a) **If** $W[i] > 0$:
            (1) **Allocate** a slot on uplink for source i.
            (2) **Call** *func*(i).
         b) **Else, if** a cell that has not been allocated a slot thus far from source i is found in the local information queue:
            (1) **Allocate** a slot on downlink for source i.
            (2) **Call** *func*(i).
         c) **Else** (no cells without allocated slots have been found in local queue):
            (1) **Allocate** a slot on uplink for source i.
            (2) **Call** *func*(i).
4. **Set** $W[i]$ to 0 for every $i \in \{S_S\}$.

Figure VI.13: BCLPR Allocation with Partial Remote Status (PMAC)

We remark that if there are no more eligible sources, the algorithm allocates slots in a cyclic manner among the *remote* sources only because, at allocation time, the occupancy of the local queue of the CP is completely known while remote sources may have enqueued more cells *after* reporting their status to the CP.

b. Policy at the Remote

The BCLPR policy at the remotes relies on the fact that each remote has exact knowledge of the CLP of its local sources. The PMAC at the CP, on the other hand, might not have this knowledge updated at all times; hence, its allocation is not optimal. The STE policy at the remote is summarized in Figure VI.14.

1. **Schedule** an invocation at the time allocated by the CP for transmission of a cell from one of the remote's sources.
2. At the scheduled time, **scan** the local queue and
   A. **Discard** cells that have already expired.
   B. **Dequeue** the oldest cell from the source having the largest CLPR. If none exists, dequeue a cell from a source having the second-largest CLPR; and so on.
   C. **Transmit** the cell.

Figure VI.14: BCLPR Allocation (SMAC)

5. STEBR Allocation

The STEBR allocation over the mobile network operates in a distributed manner by combining the features of the STE and the BCLPR allocations. For each remote, the CP maintains its MCLPR as the STE does while the local sources are assigned costs based on their CLPs as required by the algorithm. The remotes contend on the uplink control subchannel in the same manner as in the STE case and inform the CP of the number of cells requiring slot allocation on the next frame. When allocation is obtained, they use both the *deadline* and the *cost* in order to determine the cell to be serviced.

a. Policy at the CP

The STEBR algorithm at the CP works similar to the STE with one modification: when allocating slots to local sources, the criterion for service is a combination of earliest-deadline-first and largest-cost concepts rather than just the former. This guarantees that STEBR outperforms STE as obtained in the wireline scheduling case. The CP maintains four tables required for its operation:

- The number of cells that need to be serviced on the next frame, $RS[j]$ ($j \in \{S_N\}$).
- The MCLPR of every station (see Section 3 above for a detailed definition).
- The number of cells expired in node $j \in \{S_N\}$ as known at the CP, $EC[j]$.
- The costs of the local sources, $STEBR\_cost[i]$ for every source $i \in \{S_1\}$.

The STEBR algorithm at the CP is summarized in Figure VI.15.

1. **Discard** local cells whose ToE is smaller than the time it takes for the CP to transmit one cell on the downlink channel.
2. **Find** the number of local cells that must be serviced on the next frame.
3. **Calculate** RS_T, the total number of cells from all nodes that must be serviced on the next frame:
$$RS_T = \sum_{j \in \{S_N\}} RS[j].$$
4. **Calculate** the average number of arrivals so far from every station $j \in \{S_N\}$:
$$E\{A[j]\} = \sum_{i \in \{S_j\}} E\{\lambda_i\} \times (t - t_{0_i}).$$
5. **Calculate** $MCLPR[j]$ for every $j \in \{S_N\}$ using Equation (VI.1).
6. **Calculate** $STEBR\_cost[i]$ for every source i having a cell in the local queue:
$$STEBR\_cost[i] = \left(\frac{\text{Discarded from source } i+1}{\text{Arrived from source } i} \right) \Big/ \text{Allowed } CLP[i].$$
7. **If** $N_{IS} \geq RS_T$:
   A. **Allocate** $RS[j]$ information slots to a source in node $j \in \{S_N\}$ having $RS[j] > 0$.
   B. **Allocate** the remaining N_{IS}-RS_T slots (if any) to a source in the node having the largest MCLPR.
8. **Else** ($N_{IS} < RS_T$):
   A. **Repeat** RS_T-N_{IS} times:
      i) **Increment** $EC[j]$ of node $j \in \{S_N\}$ (with $RS[j] > 0$) having the smallestMCLPR.
      ii) **Calculate** $MCLPR[j]$.
      iii) **Decrement** $RS[j]$.
      iv) **If** j is the CP:
         a) **Increment** $EC[i]$ of source i having the least $STEBR\_cost$ among $\{S_1\}$.
         b) **Calculate** $STEBR\_cost[i]$.
   B. **Allocate** the N_{IS} slots in the frame such that $RS[j]$ slots are allocated to a source within node $j \in \{S_N\}$ having $RS[j] > 0$. At the CP, $RS[1]$ slots are allocated among the sources having the largest values of $STEBR\_cost$.
9. **Set** $RS[j]$ to 0 for every $j \in \{S_N\}$.

Figure VI.15: STEBR Allocation with Partial Remote Status (PMAC)

b. Policy at the Remote

The STEBR implementation at the remotes relies on the fact that each remote has exact knowledge of deadlines of its local waiting cells and the exact loss performance of its local sources. The PMAC at the CP, on the other hand, has only partial knowledge of this information; hence, its allocation is not optimal. In order to improve the performance of the network, the MAC operates in a distributed fashion by making decisions regarding the cells to be transmitted locally within each remote (SMAC). Decision regarding cell transmission takes into account both loss and delay requirements as well as performance thus far. When the PMAC allocates an information slot to an SMAC, the latter calculates the costs of its local sources. If there are some cells requiring immediate service, then it transmits the first (oldest) cell from the source having the largest cost among these cells only. Otherwise, it services the first (oldest) cell from the source having the largest cost among all waiting cells. This scheme is expected to outperform the STE approach since the slot is allocated to the cell that would "cause the most damage" if not serviced (i.e., the cell that would increase the CLPR the most). As in STE, priority is given to remote-to-remote connections by considering the cell *times to latest transmission* rather than their deadlines. An ALLOCATE_REQUEST control message is sent only when a cell is to expire in the next two frames in a remote-to-CP connection or three frames in a remote-to-remote connection rather than when the queue becomes non-empty. The STEBR algorithm at the remote is summarized in Figure VI.16.

1. **Schedule** an invocation at the time allocated by the CP for transmission of a cell from one of the remote's sources.
2. At the scheduled time, **calculate** the costs of all the non-empty sources.
3. **Calculate** the number of cells having deadlines within the next two frames (for remote-to-CP) or three frames (for remote-to-remote connections). Also, **calculate** the time to latest transmission for every cell, and **discard** cells with negative values.
4. **If** there are cells to expire in the next two/three frames:
   A. **Dequeue** among these cells the first from the source having the largest cost.
5. **Else** (no cells to expire in the next two/three frames):
   A. **Dequeue** among all waiting cells the first from the source having the largest cost.
6. **If** a cell has been dequeued:
   A. **Transmit** the cell.

Figure VI.16: STEBR Allocation (SMAC)

6. Summary

In this section, we considered the case where only a partial knowledge of the status of remotes is available to the channel-allocation algorithm. Due to lack of complete remote queues information, the performance of the scheduler in this case is upper bounded by its wireline counterpart.

The advantages of this method are its simplicity and efficiency. The remotes need to contend on the uplink control subchannel only when their queue becomes non-empty or cells approach their deadlines (only if an information slot has not been allocated on current frame because of piggybacking). Since the number of times the remotes are required to do so is relatively small, computations and battery power are saved. Additionally, the scheme adds small overhead (about 2.2%) to the transmission thereby the available channel rate for transmission of information is large.

Among disadvantages, the scheme does not guarantee (this might sometimes even be far from being correct) that the CP has knowledge of the status of the remote queues. For example, consider, in a given frame, a remote that did not get allocation on the

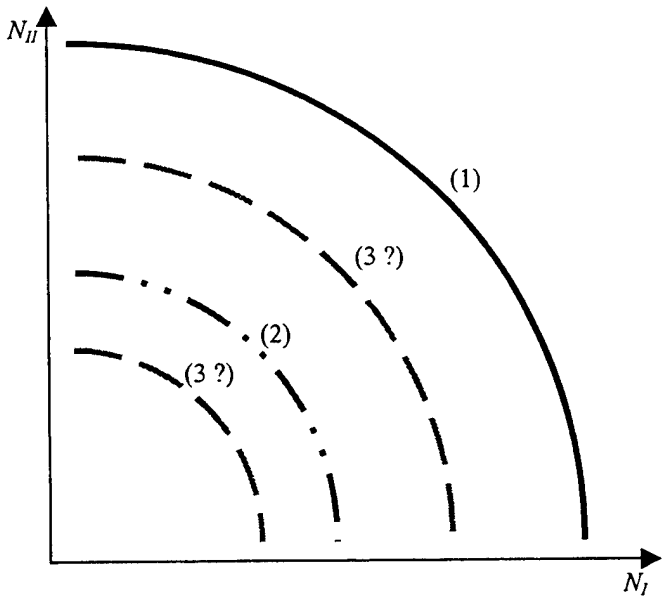
previous frame and whose ALLOCATE_REQUEST message has been lost. A possible outcome of this lack of information is a non-perfect allocation by the CP resulting in low system performance: empty remotes may get allocation while non-empty remotes might not.

C. SCHEDULING BASED ON COMPLETE REMOTE STATUS

In a wireline network, at the time of service, the server has exact knowledge of the status of the queue: the number of waiting cells, their classes, their deadlines, etc. A complete knowledge of the status of the network queues is not available in the wireless network because of its distributed nature.

Here we wish to address the issue of availability of "almost-complete" knowledge at the CP. By having a complete knowledge regarding the remote queues, we know that the network could approach the performance of a similar scheduler in a wireline queue. It is clear that such knowledge must be available at the CP at the beginning of every frame, thus forcing a transmission by all (non-empty) remotes on the previous frame. (Note that the update information is received at the CP from all remotes slightly before the beginning of the frame, hence it is *almost complete*.) Transmission of the update information via control messages may significantly *reduce* the available bit rate for cells. For example, assume that the network consists of 30 remote sources and every source has on average 10 cells waiting for transmission. Using the STE scheme, 30×10×8/12,000 or 20% of the channel capacity is required to transmit the deadlines of all cells in every frame. Moreover, since the instantaneous number of waiting cells from each source is a random process, the frame structure can no longer be constant; thus, synchronization problems between the stations may occur immediately after connections are established or released. On the other hand, a complete knowledge of the remote queue status improves the efficiency of the wireless scheduling and its performance could approach that of the wireline case (under reduced capacity). In this section, we wish to analyze the effect of the almost-complete knowledge on the *overall* performance.

Figure VI.17 presents a two-dimensional area as a function of two types of traffic: N_I and N_{II}. Each curve in the figure *qualitatively* represents the bounds (maximal combinations of N_I and N_{II} in which the network satisfies the QoS requirements of all sources) of an admissible region for a given scheduler; the larger the admissible region, the more efficient the scheduling scheme. Curve (1) in Figure VI.17 represents an admissible region for the scheduler in a wireline queue. Curve (2) bounds the region in the mobile network for the partial-status case. Curve (3) represents the admissible region in the almost-complete remote report case, which may be above or below Curve (2) (but below Curve (1)). The placement of this curve is to be investigated. The bound of the region in a decreased-capacity wireline queue case (not drawn) is expected to tightly bound Curve (3) from above.



(1) Wireline Server (Single Queue)
(2) Mobile Network (Distributed Queues) with Partial Remote-Status Report
(3 ?) Mobile Network (Distributed Queues) with Complete Remote-Status Report

Figure VI.17: Admissible Regions with Partial and Complete Remote Reports

Analysis of the almost-complete-status case can also exhibit the sensitivity of the schedulers to remote-status information. Sensitive schedulers are expected to

significantly improve their performance as more complete queue-status information is provided to them (upper Curve (3)). Less sensitive schemes, on the other hand, would present the lower Curve (3) indicating that the reduction in available capacity degrades the performance, offsetting the benefit of having additional information on the remote queue status.

1. Concepts

In order to support the complete knowledge of the remote queues, some modifications to the MAC are required. First, bandwidth must be allocated for transmission of the update information from the remotes. Since this information must be available at the CP at the beginning of every frame, the uplink control subchannel needs to be enlarged as necessary. This requirement affects the available bandwidth for transmission of cells; thus, the *sizes* of the frame header and subchannels must be adapted. Second, in order to have the *latest* possible status of the remote queues at the CP, the locations of the uplink control and information subchannels are exchanged as illustrated in Figure VI.18 (see Figure III.11 for comparison). The control messages are sent on the *last* portion of the frame; transmission by every non-empty remote on each frame is now guaranteed. Other than that, the structure of the frame remains the same.

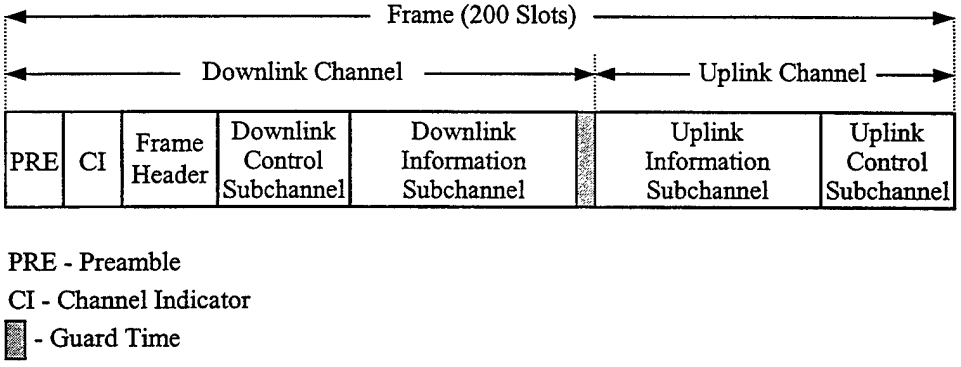


Figure VI.18: A Modified MAC Frame Format

Third, the uplink control subchannel cannot operate in a contention mode any more. The CP must receive the queue status from *all* remotes at all times, thus no

collision is allowed. Fourth, the uplink control channel must support status messages from the maximum number of remote stations (15) and/or sources (60). Even if at the beginning of the operation there are fewer remotes/sources than the maximum allowed in the network, the uplink control subchannel is still required to support the additional station(s) and/or source(s). In order to simplify the management of the control slots among all remotes, slots on the uplink control subchannel are reserved for the worst case (maximum number of remotes and maximum number of sources), even if not all remotes/sources are active. The issue now is to resolve the problem of ordering transmission of the status-messages. Fifth, a unique control message for use by the remotes to inform their queue status should be defined and supported. The values within the fields of this control message may vary according to the scheduling scheme in use. Finally, the PMAC must now make use of all the status information supplied by the remotes in order to better schedule the information slots.

2. Implementation

We begin with a discussion of the bandwidth required for the update process. We consider the worst-case scenario in which all 15 possible remotes are present in the mobile network. Additionally, all possible connections (60) are assumed active. Designing one control message to contain reports of eight sources, the maximum number of status messages from the remotes to the CP in one frame, is then given by

$$14 + \left\lceil \frac{60-14}{8} \right\rceil = 20,$$

which provides the following default values: $N_{CD} = 4$, $N_{ID} = 9$, $N_{IU} = 9$, and $N_{CU} = 25$. The last subchannel is designed for a default of 5 control messages followed by 20 remote-status messages. The number of information slots in a frame is $N_{ID} + N_{IU} = 18$. The available channel capacity for cell transfer becomes $N_{IS} / T_{frame} = 18/0.012 = 1500$ slots/sec, i.e., a reduction of 18.2% compared to the capacity of the case with partial status.

Now, the issue of *ordering* the control message transmissions must be resolved. The uplink control subchannel is partitioned into two portions. The first is for control and

signaling (as before); its size is reported by the CP in the frame header. The second portion is used to transfer the remote queue status. The CP has a bank of indices corresponding to report slots of remotes. Whenever a new call is accepted by the CP, it determines whether or not its source needs a new report slot; if the number of active connections from the originating node before the new call modulus 8 is 0, then a new report slot is needed. If a slot is required, the CP finds the *first* vacant slot from the bank and marks it as being occupied. The remote is then informed about that, using a special field in the ATM "Connect" signaling message.[12] The remote uses this index to determine the time in which it needs to transmit the status message. Whenever a connection is released and the number of remaining active connections within that remote modulus 8 becomes 0, the CP determines that the *largest* index allocated to the remote has become vacant and can be reallocated.

A new REMOTE_STATUS control message is defined in the system as shown in Figure VI.19. The message contains eight status fields, one for each active connection at the originating remote. A remote transmits a REMOTE_STATUS message for every eight active connections that are originated within the node. If there are more than eight active connections, then one message is generated for up to every eight MVCs. The MSI field identifies the sender of the message. Since it is desirable to save as much bandwidth as possible, the REMOTE_STATUS message does not include the MVCI explicitly. Instead, the status fields are related to MVCIs in an increasing order: the status-of-1st-MVCI field relates to the active MVC of this remote having the smallest MVCI, the status-of-2nd-MVCI field relates to the active MVC of this remote having the second from the smallest MVCI, etc. If a remote has more than eight active connections, the messages are referred to sequentially. The status fields contain four bits each, thus allowing to inform the CP about values between 0 and 15. For status values greater than 15, the corresponding status field is set to 15.

[12] This requires some augmentation in the ATM control and MATM control units.

	7	6	5	4	3	2	1	0 (Bits)
1	REMOTE_ STATUS				MSI			
2	MSI		reserved		Status of 1st MVCI			
3	Status of 2nd MVCI				Status of 3rd MVCI			
4	Status of 4th MVCI				Status of 5th MVCI			
5	Status of 6th MVCI				Status of 7th MVCI			
6	Status of 8th MVCI				CRC-16			
7	CRC-16							
8	CRC-16							

Figure VI.19: REMOTE_STATUS Control Message

In some cases, the power consumption of the remotes may be considerately decreased by allowing the remotes to avoid transmissions of REMOTE_STATUS messages at their allocated slot, if there is nothing to report about the corresponding MVCs. A typical scenario for such savings would be a remote with a small number of active connections that are idle most of the time.

Next, we discuss the contents of the status-of-k^{th}-MVCI field ($1 \leq k \leq 8$) within the REMOTE_STATUS message. The contents are dependent on the scheduling scheme; their representative values are summarized in Table VI.2.

Scheduling Scheme	Field Status of k^{th} MVCI in REMOTE_STATUS Message Represents
Static	Number of waiting cells of the k^{th} MVCI
STE	Number of cells of the k^{th} MVCI to expire in the next two frames (*)
BCLPR	Number of waiting cells of the k^{th} MVCI
STEBR	Number of cells of the k^{th} MVCI to expire in the next two frames (*)

(*) …in the next three frames for remote-to-remote connections

Table VI.2: Interpretation of the Status-of-k^{th}-MVCI Field

Within the cells, two fields of four and eight bits are available for piggyback allocation information on the uplink channel (previously denoted as PAR and reserved fields, respectively; see Figure III.17). In remote-to-remote connections, the 8-bit field

contains the deadline of the cell for all scheduling techniques. However, in single-hop connections, it can be utilized to report other useful information (relating to the transmitting source) to the CP; the field represents a value that depends on the scheduling technique as described in Table VI.3. Only remotes that have a slot allocation on a given frame can piggyback such information for use by the CP in the next frame. Thus, the modified scheduling scheme has only a limited use for these fields.

Scheduling Scheme	PAR and Reserved Fields in Single-Hop Connections Represent	PAR and Reserved Fields in Multiple-Hop Connections Represent...
Static	Not Applicable	Deadline of the cell
STE	Number of Waiting Cells from Source	Deadline of the cell
BCLPR	CLP of the MVCI	Deadline of the cell
STEBR	Number of Waiting Cells from Source	Deadline of the cell

Table VI.3: Interpretation of PAR and Reserved Fields

Next, we address the modifications required in the schedulers under complete remote status and their modified algorithms when applicable. The modifications *affect only the policy at the CP* while the remote policies are unchanged.

3. Static Allocation

The static-allocation algorithm assumes that the number of waiting cells within source i, i.e., the occupancy of source i received during the previous frame, is stored at the CP in $W[i]$ ($i \in \{S_S\}$). The modified static-allocation algorithm is summarized in Figure VI.20.

1. **Discard** local cells whose ToE is smaller than the time it takes for the CP to transmit one cell on the downlink channel.
2. **Find** $W[i]$, the number of waiting cells in all local sources i, for every $i \in \{S_S\}$.
3. **Calculate** for every source $i \in \{S_S\}$:
$$Z[i] = Z[i] + CRS[i].$$
4. **Repeat** N_{IS} times:
   A. **If** there is at least one source with $W[i] > 0$:
      i) **Find** maximum $Z[i]$ ($i \in \{S_S\}$, $W[i] > 0$), say, $Z[j]$.
   B. **Else** (no sources with $W[i] > 0$):
      i) **Find** a remote source j in a cyclic manner.
   C. **Decrement** $W[j]$.
   D. **Allocate** an information slot to source j.
   E. **Decrement** $Z[j]$.
5. **Set** $W[i]$ to 0 for every $i \in \{S_S\}$.

Figure VI.20: Static Allocation with Complete Remote Status (PMAC)

4. STE Allocation

In STE, each source informs the CP about the number of cells that must be serviced in the next frame. However, the scheme only makes use of the total number of such cells per station. The number of cells received from individual sources during the previous frame are summed for all nodes and stored in $RS[j]$ ($j \in \{S_N\}$). A subset of the nodes in the network that have transmitted cell(s) during the previous frame also inform their queue occupancies, which are stored by the CP in $W[j]$ ($j \in \{S_N\}$). The scheme contains one difference from the partial-remote-status case; if $N_{IS} \geq RS_T$ (where RS_T is the total number of cells that must be serviced in the next frame), instead of allocating the extra slots (if any) in a cyclic manner, sources having $W[j] > RS[j]$ are allocated up to a maximum of $W[j]$-$RS[j]$ slots whenever available.

5. BCLPR Allocation

The BCLPR algorithm requires no modifications.

6. STEBR Allocation

The information available at the CP for the STEBR algorithm in the case of partial remote report relates to a *remote node* without the possibility to relate to a specific source. In the complete-status case, the number of waiting cells and the number of required slots are available *per source*. The modified STEBR algorithm is summarized in Figure VI.21.

1. **Discard** local cells whose ToE is smaller than the time it takes for the CP to transmit one cell on the downlink channel.
2. **Find** the number of local cells that must be serviced on the next frame.
3. **Calculate** RS_T, the total number of cells from all sources that must be serviced on the next frame:

$$RS_T = \sum_{i \in \{S_S\}} RS[i].$$

4. **Calculate** $E\{\lambda_i\}$, the average number of arrivals so far from all sources $i \in \{S_S\}$.
5. **Calculate** $CLPR[i]$ for every $i \in \{S_S\}$.
6. **Calculate** $STEBR\_cost[i]$ for every source i having a cell in the local queue:

$$STEBR\_cost[i] = \left(\frac{\text{Discarded from source } i + 1}{\text{Arrived from source } i} \right) \Big/ \text{Allowed } CLP[i].$$

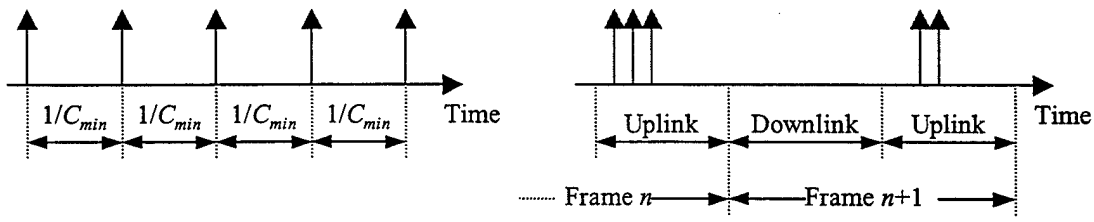
7. **If** $N_{IS} \geq RS_T$:
   A. **Allocate** $RS[i]$ information slots to among sources i having $RS[i] > 0$.
   B. **Allocate** the remaining N_{IS}-RS_T slots (if any) to sources having waiting cells (other than those requiring service on the next frame) as piggybacked in a cell on previous frame (if any).
   C. **Allocate** the remaining slots (if any) to remote sources in a cyclic manner.
8. **Else** ($N_{IS} < RS_T$):
   A. **Repeat** RS_T-N_{IS} times:
      i) **Find** source $i \in \{S_S\}$ (with $RS[i] > 0$) having the smallest CLPR (in the remotes) or minimum $STEBR\_cost$ (at the CP).
      ii) **Increment** $EC[i]$.
      iii) **Calculate** $CLPR[i]$ (and also $STEBR\_cost[i]$, if i is local).
      iv) **Decrement** $RS[i]$.
   B. **Allocate** the N_{IS} slots in the frame such that $RS[i]$ slots are allocated to source $i \in \{S_S\}$, having $RS[i] > 0$.
9. **Set** $RS[i]$ to 0 for every $i \in \{S_S\}$.

Figure VI.21: STEBR Allocation with Complete Remote Status (PMAC)

D. PERFORMANCE COMPARISON BASED ON SIMULATION RESULTS

Simulation results are expected to be inferior to those in the wireline implementation because of the structure of the MAC. At the beginning of a frame, allocation of the service slots must be determined for the entire frame rather than for a single service slot as in wireline cases. Often the total number of waiting cells from all stations is smaller than the number of slots in a frame; this causes the PMAC to perform ineffective decision about allocation of the additional slots, e.g., by allocating the extra slots in an inefficient cyclic manner.

The organization of a relatively-large number of service slots in frames gives rise sometimes to a bursty service fashion. This phenomenon is demonstrated using the static-capacity allocation. In the calculations of the wireline multiplexer, the minimum capacity, C_{min}, required to satisfy the transmission requirements has been obtained such that the server queries the queue every $1/C_{min}$ seconds. In Figure VI.22a, we can see an example for static allocation in a wireline queue, where the vertical arrows mark capacity allocation for source i. When implementing static allocation in the MAC, the allocation to a source is performed on the downlink information subchannel (for CP sources) or on the uplink information subchannel (for remote sources). Allocation to the above source i, in case it is a remote source, is shown in Figure VI.22b. Despite the fact that the source gets the same required capacity of C_{min} cells/sec over time, the allocation appears in bursts. The bursty allocation behavior causes a portion of the cells to be discarded because of timeout violation.



(a) Periodic Allocation in Wireline Case (b) Bursty Allocation in Wireless Case

Figure VI.22: Static Allocation in Wireline and Wireless Cases

Relating specifically to static allocation, another cause for performance degradation is the rounding of the capacity (a floating-point number) to the number of slots per frame (an integer). For example, if the required capacity is translated into 2.5 slots/frame, then the source enjoys three and two slots/frame, alternately. This also creates a bursty service pattern that leads to deviation in results from the theoretical results obtained in the wireline case. This phenomenon is schematically represented in Figure VI.22. On the other hand, when a given station is relatively loaded with heterogeneous traffic, we expect the theoretical results to be conservative. The reason for that, as previously discussed, is that we ignore the multiplexing gain obtained by multiplexing heterogeneous sources. In such cases, the required capacity for all the heterogeneous sources is calculated as the sum of the capacities required for each class individually.

Yet another discrepancy is caused as a result of the frame structure. The downlink subchannel contains several cells packed among other data (e.g., frame header), into one transmission. The CP makes decisions regarding allocation of information slots to sources at the beginning of each frame. Local and relayed cells are then dequeued and organized into a frame. The actual transmission of the frame follows immediately thereafter. The cells within the downlink transmission are decoded by the remotes only after the *entire* transmission is received. Therefore, the transmission delay experienced by the cells on the downlink is actually longer than the time it takes to transmit each single cell separately. As the number of transmitted cells on the downlink becomes larger, so is the transmission delay. This phenomenon is demonstrated in Figure VI.23, where N_1 and N_2 ($N_1 > N_2$) are the number of information slots on the downlink channel in the upper and lower diagrams, respectively. The transmission delay on the former is larger than that of the latter. The transmission delay on the uplink information subchannel is always the same since it involves only a single cell.

In general, the factors mentioned above are expected to degrade the performance of the schedulers in the mobile network in comparison to the wireline network, even when complete remote queue status is available.

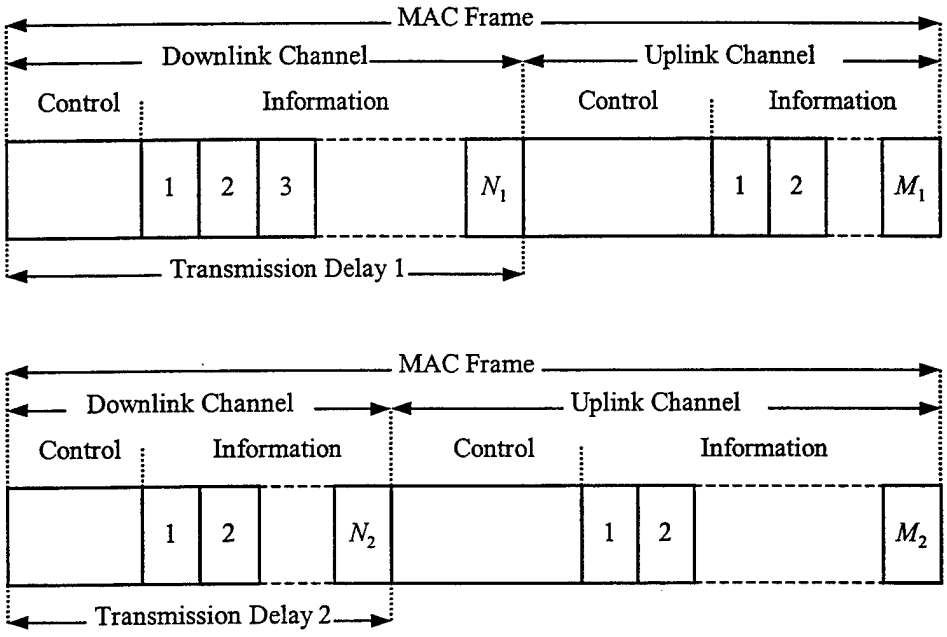


Figure VI.23: Variable Transmission Delays of Cells on the Downlink Channel

E. SUMMARY

The mobile network can be considered in terms of distributed queues at the CP and in the remotes, making the scheduling more involved in relation to wireline systems. The reason for that is the lack of complete knowledge at the network server (the PMAC at the CP) about the loss incurred by remote sources and the status of all queues, namely the occupancies and waiting cell deadlines. The exact status of these sources are known only at their originating stations. Also, the service decisions performed in the mobile network are made frame-by-frame rather than cell-by-cell as in wireline cases. The computational effort in the former system is decreased; on the other hand, the probability of improper channel allocation increases, reducing the possible number of admitted sources and the channel throughput.

In this chapter, for each scheduling algorithm discussed in Chapter V, we have developed a corresponding channel-allocation algorithm for operation in the wireless network. Two cases were considered. In the first case, only partial status of the remotes is

made available to the scheduler. The partial information is obtained whenever the queue becomes non-empty or as piggyback data within a transmitted cell. The overhead required for obtaining partial information is minimal. In the second case, about 20% of the channel capacity is devoted for gathering complete remote-status information during every frame. By doing so, we aim at scheduling decisions that are as close as possible to those made in the wireline queue. A special control message, dedicated for this purpose, allows each source to pass its status using four bits. The availability of almost-complete remote-status information requires modifications to the scheduling algorithms developed for the partial case. In the next chapter, we examine whether the overhead incurred due to complete status leads to improvement in the admissible region and the channel throughput.

VII. PERFORMANCE OF SCHEDULERS IN THE WIRELESS CHANNEL

The purpose of this chapter is to evaluate the performance of the proposed scheduling algorithms in the wireless network. We consider five representative scenarios to study the performance (admissible region and channel throughput) obtained by the various scheduling schemes. Since there is an enormous number of possible scenarios, we use the five scenarios to represent different traffic conditions in the wireless network.

To evaluate the performance of different MAC scheduling policies in each scenario, we use two types of loads: a constant load and a variable load. The constant load is common to all possible combinations of traffic sources within a given scenario. The variable load depends on a set of two parameters, denoted as N_1 and N_2; these values reflect the numbers of sources of Classes I and II, respectively, generated within one or two stations. Section A contains the details of the representative scenarios. In Section B, we detail the implementation of the simulation program. The behavior and performance of the network under the defined scenarios are observed and analyzed in Section C as a function of the variable load (i.e., as a function of N_1 and N_2). We investigate both cases of partial remote status and complete remote status. Section D concludes the chapter with a discussion of the results.

A. SCENARIOS

The details of the five sample scenarios to be used in evaluating the performance of different MAC schedulers over the wireless network are presented here. We use the following notation in describing these scenarios:

- Low load: Traffic from a single station, occupying up to 5% of the channel capacity.
- Medium load: Traffic from a single station, occupying 5% to 10% of the channel capacity.
- High load: Traffic from a single station, occupying more than 10% of the channel capacity.

1. Scenario 1

The first scenario represents a small-scale operation, managed and coordinated by the CP: the number of active stations is small and most traffic load flows between the CP and the remotes and vice versa. Two of the stations generate most of the load in the network: the CP and one of the remotes. The other stations contribute low to moderate load. Table VII.1 details the load distribution among the stations in the network. The variable load in Scenario 1 comprises the number of data sources within the CP and Remote 4.

Station Generating Traffic	Speech Sources		Video Sources		Data Sources	
	Quantity	Destination	Quantity	Destination	Quantity	Destination
CP	1	Remote	1	Remote	N_1	Remote
Remote 2	1	CP	--	--	--	--
Remote 3	--	--	1	CP	1	CP
Remote 4	--	--	1	CP	N_2	CP
Remote 5	--	--	--	--	1	Remote

Table VII.1: Scenario 1 Characteristics

2. Scenario 2

The second scenario represents a large-scale operation, managed and coordinated by the CP: the number of active stations is large, and most traffic load flows between the CP and the remotes and vice versa. The CP, at the heart of the network, generates a large portion of its traffic: speech conversations and data transfers (no video connections are active in the network in this case). In other stations, a small number of data sources are active, thus (each) contributing a very low load. Table VII.2 details the load distribution among the stations in the network. The variable load in Scenario 2 comprises the number of speech and data sources within the CP.

Station Generating Traffic	Speech Sources		Video Sources		Data Sources	
	Quantity	Destination	Quantity	Destination	Quantity	Destination
CP	N_1	Remote	--	--	N_2	Remote
Remote 2	--	--	--	--	3	CP
Remote 3	--	--	--	--	2	CP
Remote 4	--	--	--	--	2	CP
Remote 5	--	--	--	--	1	CP
Remote 6	--	--	--	--	1	CP
Remote 7	--	--	--	--	1	CP
Remote 8	--	--	--	--	1	CP
Remote 9	--	--	--	--	1	CP
Remote 10	--	--	--	--	1	CP
Remote 11	--	--	--	--	1	CP
Remote 12	--	--	--	--	1	CP
Remote 13	--	--	--	--	1	CP
Remote 14	--	--	--	--	1	CP
Remote 15	--	--	--	--	1	CP
Remote 16	--	--	--	--	1	Remote

Table VII.2: Scenario 2 Characteristics

3. Scenario 3

The third scenario represents a balanced load: the number of active stations is average, and all active stations generate traffic of low to medium load. Traffic flows between the CP and the remotes as well as between remotes. Table VII.3 details the load distribution among the stations in the network. The variable load in Scenario 3 consists of the number of data sources within the CP and Remote 5.

Station Generating Traffic	Speech Sources		Video Sources		Data Sources	
	Quantity	Destination	Quantity	Destination	Quantity	Destination
CP	--	--	--	--	N_1	Remote
Remote 2	1	CP	--	--	--	--
Remote 3	--	--	1	CP	--	--
Remote 4	--	--	--	--	2	CP
Remote 5	--	--	--	--	N_2	CP
Remote 6	--	--	--	--	1 1	CP Remote
Remote 7	--	--	--	--	1	CP
Remote 8	--	--	--	--	1	Remote
Remote 9	--	--	--	--	1	Remote
Remote 10	1	CP	--	--	1	CP

Table VII.3: Scenario 3 Characteristics

4. Scenario 4

The fourth scenario aims to test the network with moderate- and high-load sources. All connections are of type speech and video only, and the load is somewhat equally divided between the downlink and the uplink transmissions. Table VII.4 details the load distribution among the stations in the network. The variable load in Scenario 4 consists of the number of video sources within the CP and the number of speech sources within Remote 2.

Station Generating Traffic	Speech Sources		Video Sources		Data Sources	
	Quantity	Destination	Quantity	Destination	Quantity	Destination
CP	--	--	N_1	Remote	--	--
Remote 2	N_2	CP	--	--	--	--
Remote 3	1	CP	--	--	--	--
Remote 4	1	CP	--	--	--	--
Remote 5	1	CP	--	--	--	--
Remote 6	--	--	1	CP	--	--
Remote 7	--	--	1	CP	--	--
Remote 8	--	--	1	CP	--	--

Table VII.4: Scenario 4 Characteristics

5. Scenario 5

The fifth scenario attempts to emulate the behavior of the mobile network under uni-directional traffic, on the uplink. A (silent) CP and three remotes are present in the network. For simplicity, all active connections are of type video. Table VII.5 details the load distribution among the stations in the network. The variable load here consists of the number of video sources within Remotes 2 and 3.

Station Generating Traffic	Speech Sources		Video Sources		Data Sources	
	Quantity	Destination	Quantity	Destination	Quantity	Destination
CP	--	--	--	--	--	--
Remote 2	--	--	N_1	CP	--	--
Remote 3	--	--	N_2	CP	--	--
Remote 4	--	--	1	CP	--	--

Table VII.5: Scenario 5 Characteristics

B. SIMULATION

1. Program Flow

The wireless network is simulated using the OPNET™ package.[13] The network includes the CP and 15 remote stations (the number of active remotes however is scenario dependent), operating over a multiple-access channel. Each station (see Figure VII.1) includes three modules that generate traffic (one for each traffic class) and send it to the DLC in format of ATM cells, or receive ATM cells from the DLC (i.e., AAL/ATM layers are an implicit part of these modules). The station also includes a MAC, a physical layer, and a mobile control unit (at the CP only), which simulates a mobile admission controller.

The simulation, which starts to run at time zero, assumes that all the active remotes have already been registered in the network. Each node contains a module that generates traffic from multiple sources as per the scenario. Prior to traffic generation, each source passes through a network admission-control procedure via message exchange with the CP (also performed at time zero). For remote sources, the SMAC sends a special control message while at the CP; this is done locally involving no transmissions. The admission controller at the CP dynamically assigns a unique MVCI[14] for each source. Since no more then 16 nodes are modeled in the network, the IDVC for each party is determined prior to the beginning of simulation. In this simulation, no source is rejected during admission because it is desirable to identify the boundaries of the admissible regions for various schedulers.

---

[13] OPNET is a registered trademark of MIL 3, Inc.

[14] In some circumstances, working points within the admissible region contain more than 60 active users at the same time. In the case of partial status report, the simulation allows MVCI values larger than 63 while maintaining the addressing mechanism built in the MAC layer (see Chapter III). In real systems, however, if more than 60 sources are active simultaneously, the size of the MVCI field needs to be expanded beyond six bits. In the case of complete status report, the number of dedicated uplink control slots for remote reports is not expanded beyond 20. Same report slots are assigned to different remotes; however, the simulation program assumes that no collisions have occurred and that all messages can be recovered. In real systems, if more than 60 sources are active simultaneously, the number of dedicated uplink control slots for remote reports must be increased accordingly.

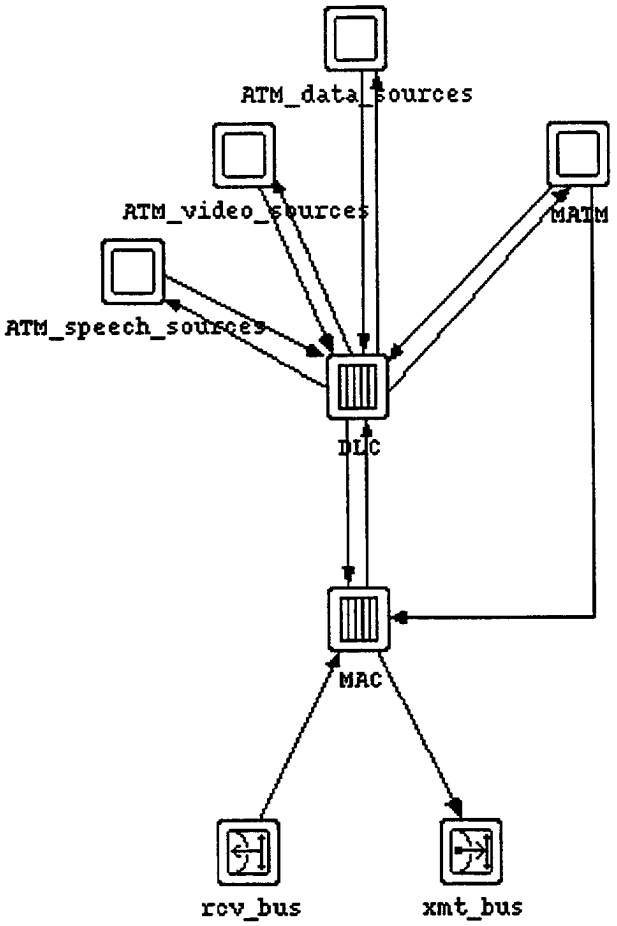


Figure VII.1: Structure of a Mobile Station in OPNET Simulation

Every set of octets generated by the source modules (47 for speech, 45 for video, and 44 for data) is padded arbitrarily to 48 octets, representing an ATM-cell payload, and an appropriate 5-octet cell header is built. The cells are transferred to the DLC and from there to the MAC of the station; the latter transforms the cells into mobile cells (see Figure III.17) and enqueues them into the buffer.

The length of a MAC frame is 12 milliseconds; each frame is divided into four subchannels. At the beginning of each frame, the PMAC sends a downlink transmission, which includes the frame header, control messages, and information cells. The sizes of the four subchannels are indicated in the frame header. The sizes of the uplink and

downlink control subchannels are determined according to the algorithm described in Figure VI.5 such that the frame lasts the longest possible within the limitation of 12 milliseconds. The channel allocation for cells on the uplink is also reported by the PMAC via the frame header, using the MVCI/IDVC notation. The first two downlink control messages are of type LAST_FRAME_ACK, reporting the transmission results of control messages and cells on the uplink control and information subchannels, respectively, in the previous frame.

Remotes seeking channel allocation contend on the uplink control slots using a slotted-ALOHA algorithm with parameter $p \in (0,1]$. At the beginning of every control slot, the SMAC decides whether a transmission of a control message would take place based on a standard uniform randomization using p. An ALLOCATE_REQUEST message is created and transmitted by the SMAC in case a positive decision has been made.

Remotes with slot allocation on the uplink information subchannel, transmit cell(s) in the appropriate time instant(s). Based on the scheduling algorithm utilized, the SMAC decides which of its queued cells to send. Piggyback information is added to each such cell as required by the scheduling technique being used. Cells received at the CP from sources participating in remote-to-remote connections are enqueued in the PMAC's buffer together with local cells for transmission on the downlink.

The wireless channel is assumed to be error free. The propagation delay in the channel is considered negligible. Each transmission in the channel starts with a 40-bit preamble followed by an 8-bit channel indicator, and ends with a 6 microsecond guard period during which no information is sent. The simulation assumes synchronization acquisition is always obtained by all possible transmitter-receiver pairs. Thus, the only origin of cell loss is due to expiration of cell deadlines.

2. Simulation Inputs and Outputs

A simulation of a given scenario requires the following parameters (see Appendix D for a detailed description of all simulation inputs):

- The number of active stations.

- The number of active sources within every station.
- The traffic class of every source.
- The type of each connection.
- The scheduling scheme.

The simulation reports the following output information (see Appendix D for an example):

- The number of cells generated by source $i \in \{S_S\}$ (waiting for transmission at the station's MAC), $N_{GS}[i]$.
- The number of cells discarded from source $i \in \{S_S\}$ by the MAC in the originating station due to expiration of deadline (i.e., prior to transmission), $N_{DS}[i]$.
- The number of cells from source $i \in \{S_S\}$ discarded by its destination(s), $N_{DD}[i]$.
- The total number of cells transmitted on the downlink, N_{TD}.
- The total number of cells transmitted on the uplink, N_{TU}.

In a remote-to-remote connection, cells may be discarded at the CP as well; thus, in this case, the simulation also reports the number of cells discarded by the CP. We refer to this number, associated with remote source i, as $N_{DCP}[i]$. Two factors contribute to the value of $N_{DCP}[i]$: cells that reached the CP late from the SMAC of the originating station and cells that were discarded by the PMAC at the destination remote due to late transmission.

Based on the simulation outcomes, the CLP experienced by source i (a QoS measurement), $CLP[i]$, is calculated according to the connection type as follows:

$$CLP[i] = \begin{cases} \dfrac{N_{DS}[i] + N_{DD}[i]}{N_{GS}[i]}, & \text{for CP-to-remote or remote-to-CP connections} \\ \dfrac{N_{DS}[i] + N_{DCP}[i] + N_{DD}[i]}{N_{GS}[i]}, & \text{for remote-to-remote connections.} \end{cases}$$

The normalized channel throughput is calculated as follows:

$$\overline{S}_M = \frac{(N_{TD} + N_{TU})/T_{SIM}}{C_M},$$

where C_M is the channel capacity and T_{SIM} the simulation duration.

C. SIMULATION RESULTS

The performance results of schedulers obtained from simulation are presented in the form of two-dimensional color graphs. The admissible regions of the channel-allocation algorithms are plotted as a function of the variable load over colored surfaces representing different values of normalized channel throughput. For a given scenario, this representation allows presentation of admissible region and normalized throughput performance on a single graph. Both cases of partial and complete remote status are considered.

1. Partial Remote Status

Figures VII.2 through VII.6 present the admissible regions of the mobile network for different scheduling algorithms under Scenarios 1 through 5, respectively. The figures present the performance of the schedulers using partial remote status. The axes in these plots designate the two independent parameters, N_1 and N_2, forming the variable load in the channel. The boundaries of the admissible regions are plotted as thick green lines. The color bar on the right-hand side of each figure indicates the level of the normalized throughput in the admissible region.

The STEBR algorithm is shown to outperform all other algorithms. The relative performance of the scheduling algorithms is maintained for all examined scenarios in the following (descending) order: STEBR, STE, BCLPR, static allocation, peak-rate allocation. This consistency increases the confidence in the simulation results reported here.

Comparing the performance of the (new) STEBR with that of the (known) STE exhibits an improvement in cell rate transmitted through the wireless channel for all scenarios. In addition to applying the earliest-deadline-first concept at the CP and in the remotes, STEBR also considers the loss thus far occurred for each source, thus providing

improved performance over STE. The improvements are summarized in Table VII.6. STEBR allows the admissible region to grow, in comparison with that achieved by STE, such that the channel throughput is increased by up to 10 percent. These results agree with those obtained in the wireline network case.

Scenario	Improvement Obtained using STEBR over STE		
	Additional Sources	Additional Cell Rate	
		Cells/sec	Percentage
1	30-45 Data Sources	115 - 173	6% - 9%
2	3-22 Data Sources	12 - 85	1% - 5%
3	25-30 Data Sources	96 - 115	5% - 6%
4	1-2 Speech Sources	75 - 150	4% - 8%
5	0-1 Video Sources	0 - 178	0% - 10%

Table VII.6: Cell-Rate Improvement using STEBR over STE

Both STEBR and STE outperform BCLPR and static allocation in all scenarios; as the numbers of sources and remotes in the channel increase, so is the difference in performance. The main reason for that lies in the mechanisms used by the algorithms to request channel allocation and the use of this information by the PMAC. STEBR and STE use a mechanism in which channel-allocation requests by the remotes are submitted at the latest possible instants, containing the exact number of information slots required on the next frame. Such an approach guarantees that the requesting remotes indeed *have cells* for transmission. The PMAC implementing STEBR or STE bases its channel allocation directly on the information provided by the SMACs, resulting in high probability that allocated slots for requesting sources are in fact utilized and not wasted. This mechanism also ensures that the number of transmissions in the uplink control subchannel is low. Each cell invokes a transmission of at most one allocation request message because failure in receiving the first message at the PMAC causes discarding of the cell. Usually, one allocation request message relates to several waiting cells, or cells

requiring immediate service are reported via piggyback information. Both lead to a small number of collisions. The percentage of uplink control slots utilized for transmission of allocation request messages has not exceeded 0.5% of the total available slots (on the boundaries of the admissible regions). Almost none of these messages have collided; as a result, the only cell loss is due to insufficient number of information slots for all the requesting sources.

The BCLPR and static-allocation algorithms use a different mechanism for channel-allocation requests by SMACs. Whenever a remote is non-empty and no allocation on a given frame is made for any of its sources, the SMAC contends on the uplink control subchannel to inform the PMAC about the number of its waiting cells. This causes the number of transmissions and collisions in the contention channel to increase in relation to their counterparts in STEBR and STE, resulting in a higher loss rate of allocation request messages. Typical values of the percentage of uplink control slots utilized for transmission of allocation request messages are 2% to 15% of the total available slots (on the boundaries of the admissible regions). Collisions in these slots occurred in up to 0.5% of the available uplink control slots.[15] As a result, BCLPR and static allocation make only limited use of the remote occupancies reported via the allocation request messages. In the wireline case, BCLPR makes its decisions based on the CLPRs of the sources. If the source having the largest CLPR is empty, the algorithm looks for cells from the source having the second largest CLPR and so on. In the wireless network, if the source having the largest CLPR is a remote source that happens to be empty, an information slot allocated for it is wasted (unless the remote has other non-empty sources). Due to lack of perfect remote occupancies information, the performance of BCLPR is somewhat poor. Static allocation operates in a manner similar to BCLPR; the values of a static-allocation counter for each source instead of a CLPR are considered. In summary, STEBR and STE are superior to BCLPR and static allocation because they

[15] The percentages of utilized and collided uplink control slots is a function of the transmission probability, p, of the slotted ALOHA protocol. Modifying the value of p may improve the performance obtained by the algorithms; however, this issue has not been investigated in this work.

make an extensive use of the information provided by the remotes. This information is sent relatively infrequently, thus increasing its chances to be successfully received by the PMAC.

The normalized throughput decreases with an increase in the number of remotes in the scenario. As the number of remotes increases, the multiplexing gain achieved in each remote drops, thus reducing the number of admitted sources and hence the normalized throughput. We demonstrate this phenomenon with STEBR; in Scenarios 1, 4 and 5, each which includes up to 7 remotes, the normalized throughput is larger than 0.80 throughout the boundaries of the admissible regions. In Scenario 3 (9 remotes), the maximum normalized throughput drops to about 0.62, and in Scenario 2 (15 remotes) it reaches 0.43-0.58. Static allocation is especially sensitive to the number of remotes. In Scenario 2, the maximum normalized throughput is only about 0.20 due to a large number of remotes, each having a small number of data sources requiring almost-peak-rate allocation. Peak-rate allocation in Scenario 2, is insufficient to satisfy even the constant load ($N_1 = N_2 = 0$) alone.

An interesting observation can be drawn from the results of Scenarios 2 and 4, where N_1 and N_2 represent sources from *different* traffic classes (regardless of the originating stations). The normalized throughput changes throughout the boundaries of the admissible regions of the scheduling algorithms. Generally, if the total number of active sources is not very large, a network provides better performance when handling sources from traffic classes having less-stringent CLP requirements. Thus, as the number of such sources increases to the maximum possible (on the boundary of the admissible region), a larger throughput is obtained compared to the case of sources having more-stringent CLP requirements. In Figure VII.3, as the number of speech sources (N_1) increases at the expense of the number of data sources (N_2), the normalized throughput increases. This is attributed to the fact that speech sources have a required CLP of 10^{-3}, which is less restrictive than that of data sources (10^{-6}). In Figure VII.5, N_1 and N_2 represent video sources (having a CLP of 5×10^{-5}) and speech sources, respectively. An increase in the level of the normalized throughput can be observed as N_2 increases. In

Scenarios 1, 3 and 5, where N_1 and N_2 represent sources from the same traffic class, the normalized throughputs remain at the same level along the boundaries of the admissible regions.

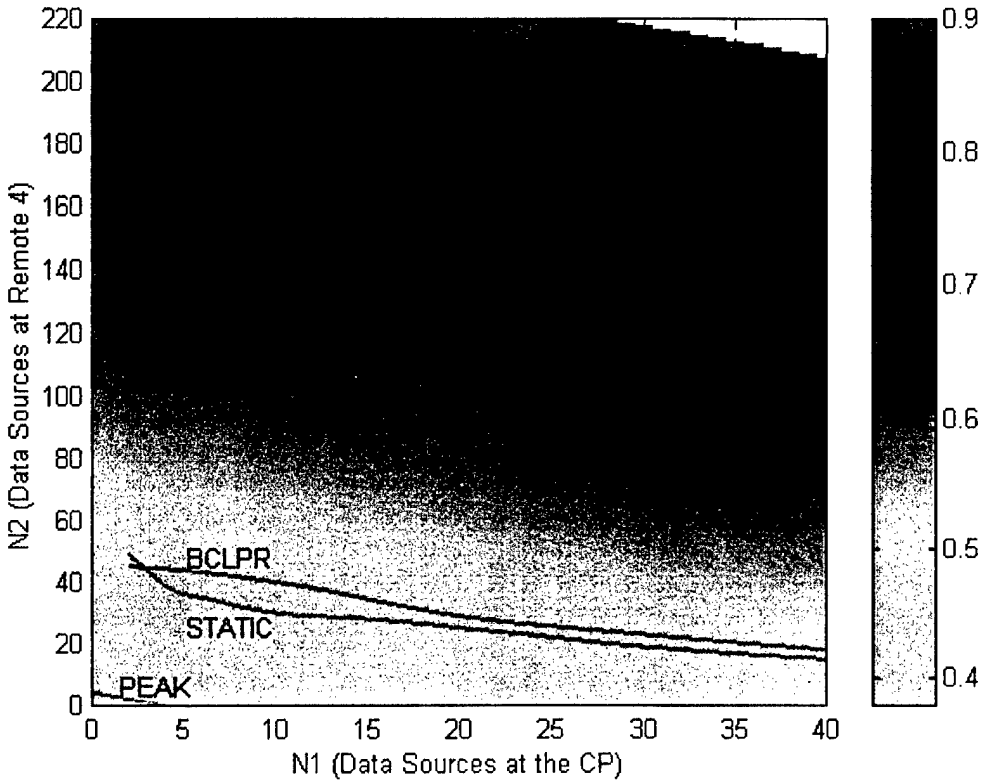


Figure VII.2: Admissible Region and Normalized Throughput for Partial Remote Status, Scenario 1

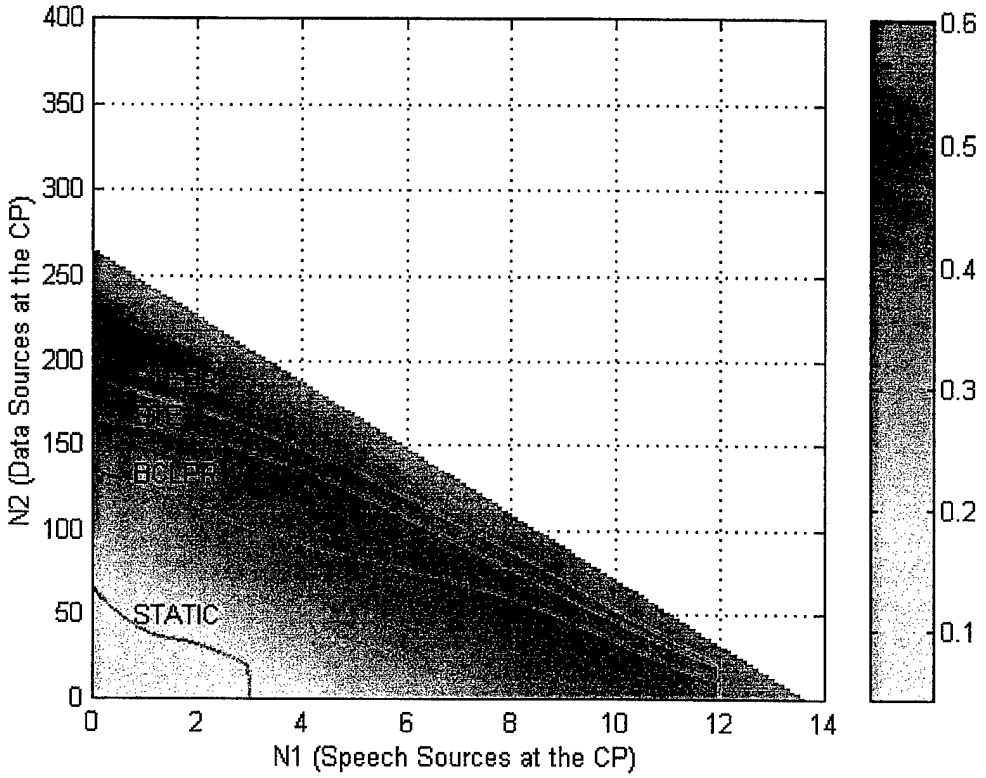


Figure VII.3: Admissible Region and Normalized Throughput for Partial Remote Status, Scenario 2

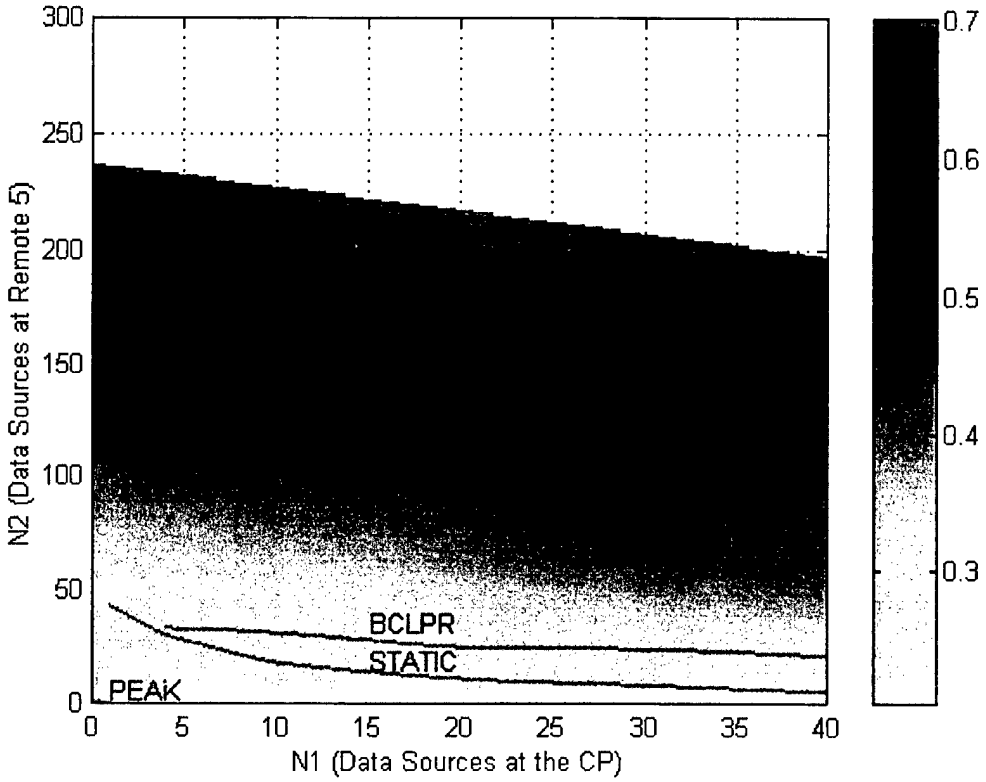


Figure VII.4: Admissible Region and Normalized Throughput for Partial Remote Status, Scenario 3

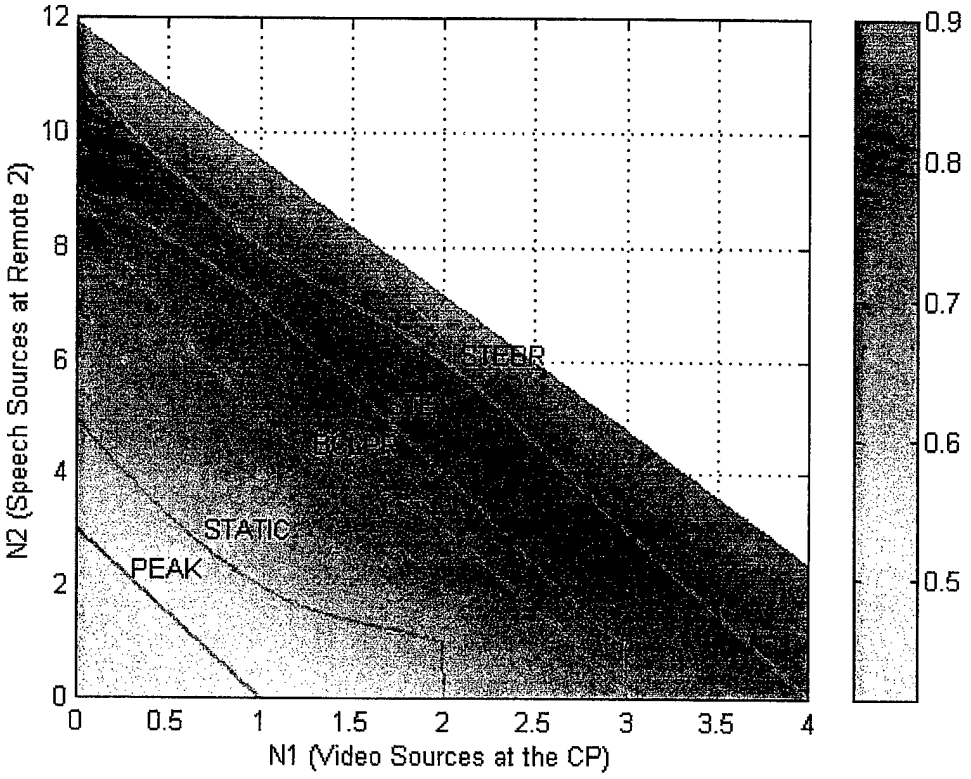


Figure VII.5: Admissible Region and Normalized Throughput for Partial Remote Status, Scenario 4

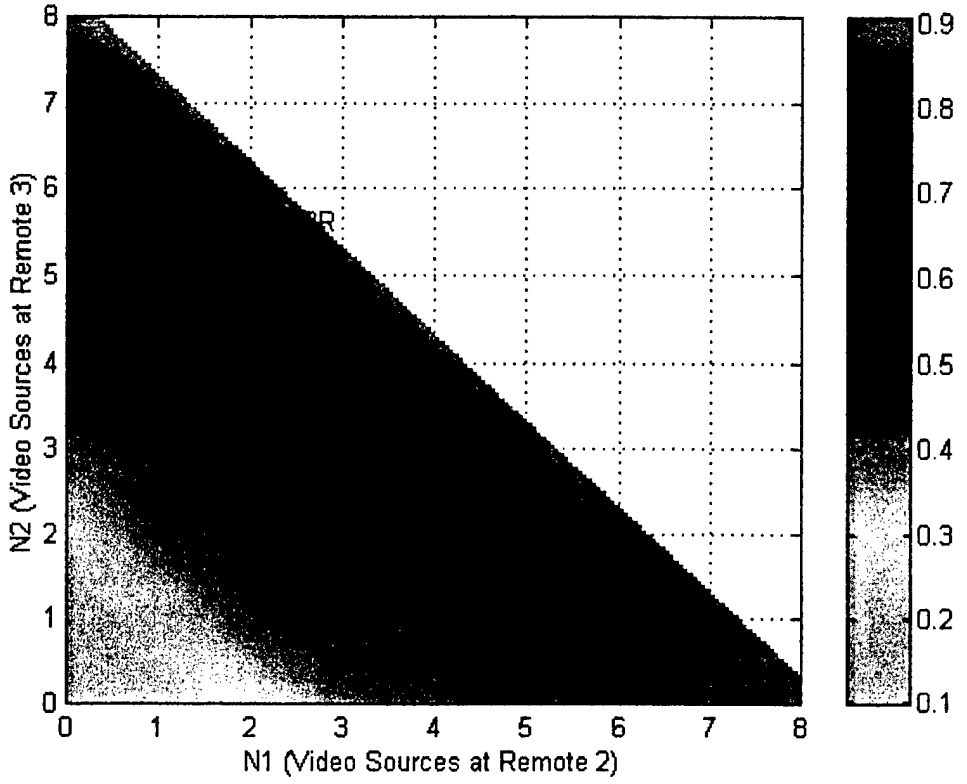


Figure VII.6: Admissible Region and Normalized Throughput for Partial Remote Status, Scenario 5

2. Complete Remote Status

Figures VII.7 through VII.11 present plots of the admissible region (thick yellow lines) and the normalized throughput for different scheduling algorithms in Scenarios 1 through 5, respectively, using complete remote status.

STEBR provides upper bounds on the performance of the schedulers in all scenarios. The relative order of performance is maintained as in the partial-status case except in Scenario 5, where BCLPR slightly outperforms STE.

The advantage of STEBR over STE is shown to be minor (if at all). The main reason for that is the availability of the number of waiting cells in a given frame from sources that transmitted in the previous frame (see Table VI.2). This turns out to be a dominant factor which leads to equal performance by the two algorithms. The small performance improvement obtained by STEBR is attributed to the better decisions it makes regarding cell discarding at the CP and in the remotes, where the exact local source costs are known to the local MAC.

Both STEBR and STE present high level of normalized throughput in the range of 0.70 to 0.82. A normalized throughput of 0.82 is the maximum possible in the complete-status case due to the 18.2% overhead required for complete status reports. The highest levels of normalized throughput are obtained in cases where the number of data sources becomes very large (hundreds) as in Scenarios 1, 2, and 3; the performance of the algorithms approaches, as is also the case in the wireline network (see Table V.7), that of a mean-rate-allocation algorithm, which achieves the maximum multiplexing gain possible. As the total number of sources in the scenario decreases as in Scenarios 4 and 5 (Figures VII.10 and VII.11, respectively), the maximum normalized throughput does not exceed 0.70. Scaling this value to the available channel capacity in the complete-status case gives normalized throughput of 0.85 (= 0.70/0.82), which is only slightly smaller than the values obtained in the wireline case (see Figure V.22).

The BCLPR algorithm performs better in Scenarios 1, 4, and 5 than in Scenarios 2 and 3. The algorithm is sensitive to both the CLPRs of the sources and their corresponding occupancies. In the complete-status case, the remote CLPRs are perfectly known in every frame; the remote occupancies are known whenever a cell is received at the CP. Thus, as the number of remotes is relatively small, it becomes easier for the CP to be informed about the number of waiting cells of most remote sources, leading to better channel-allocation decisions. This outcome is valid for static allocation as well.

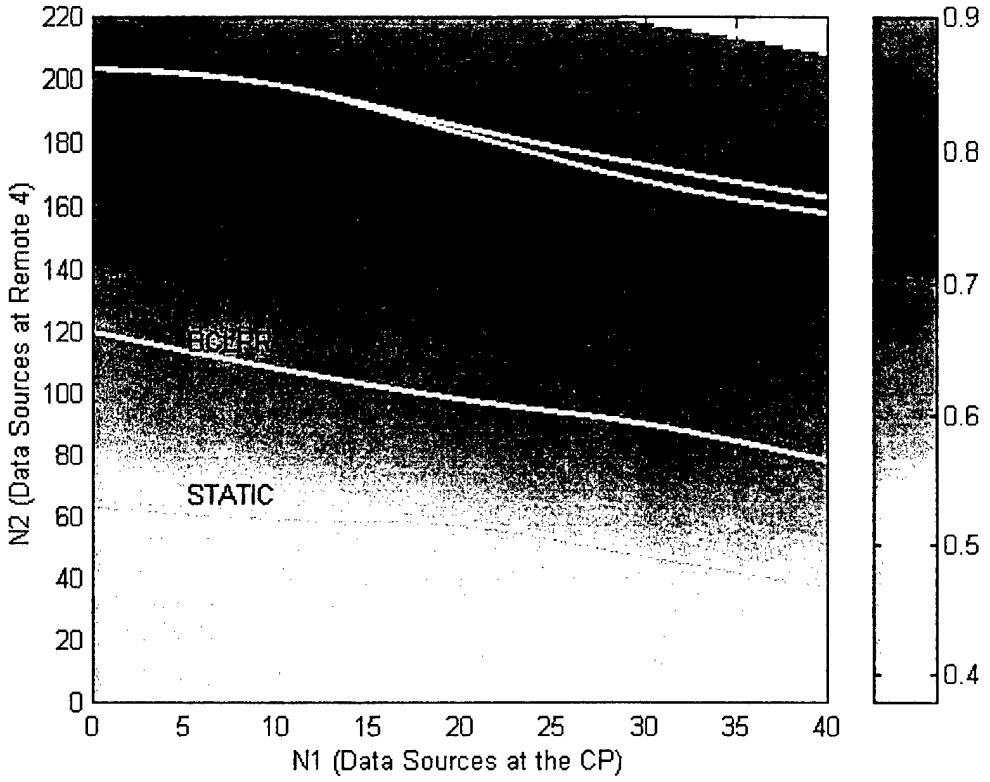


Figure VII.7: Admissible Region and Normalized Throughput for Complete Remote Status, Scenario 1

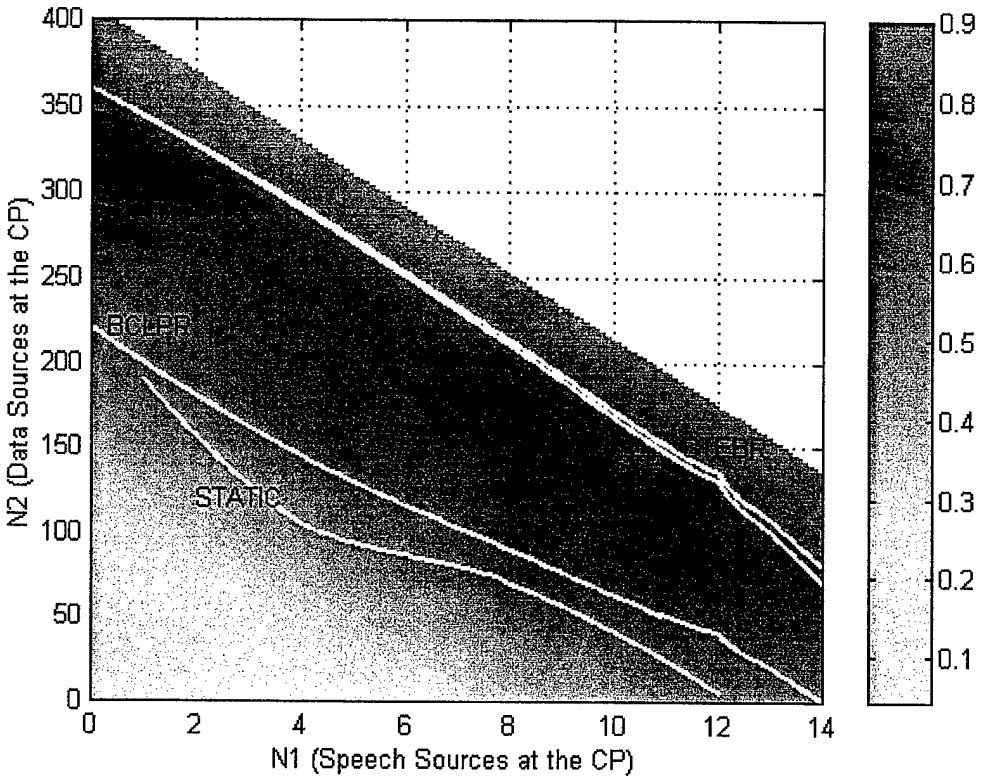


Figure VII.8: Admissible Region and Normalized Throughput for Complete Remote Status, Scenario 2

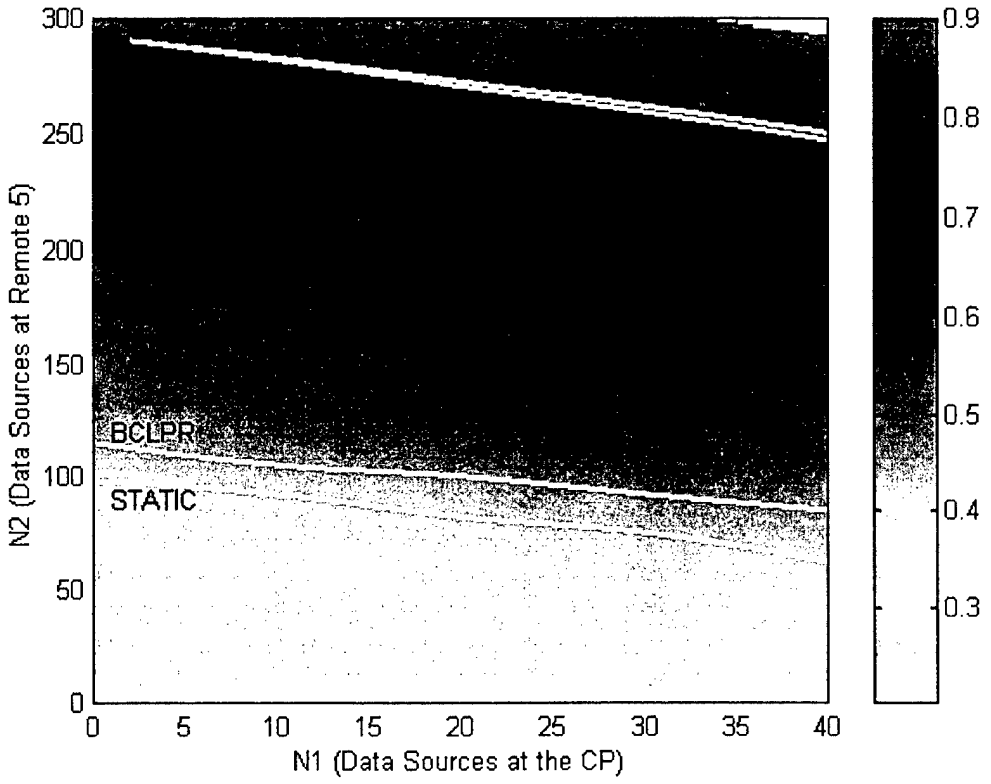


Figure VII.9: Admissible Region and Normalized Throughput for Complete Remote Status, Scenario 3

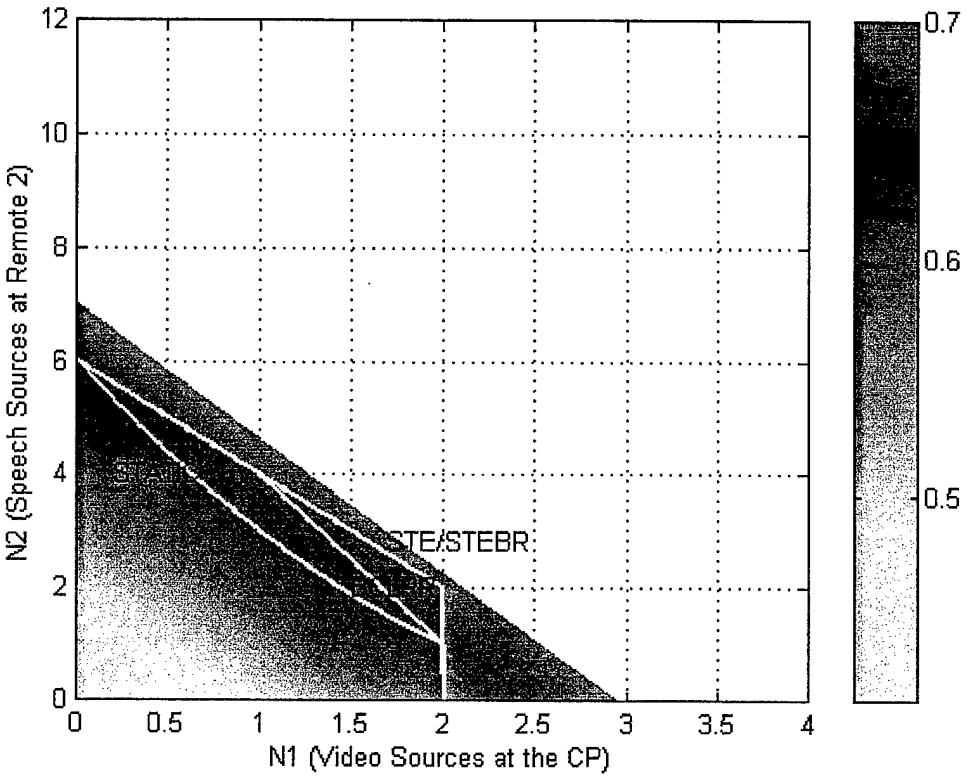


Figure VII.10: Admissible Region and Normalized Throughput for Complete Remote Status, Scenario 4

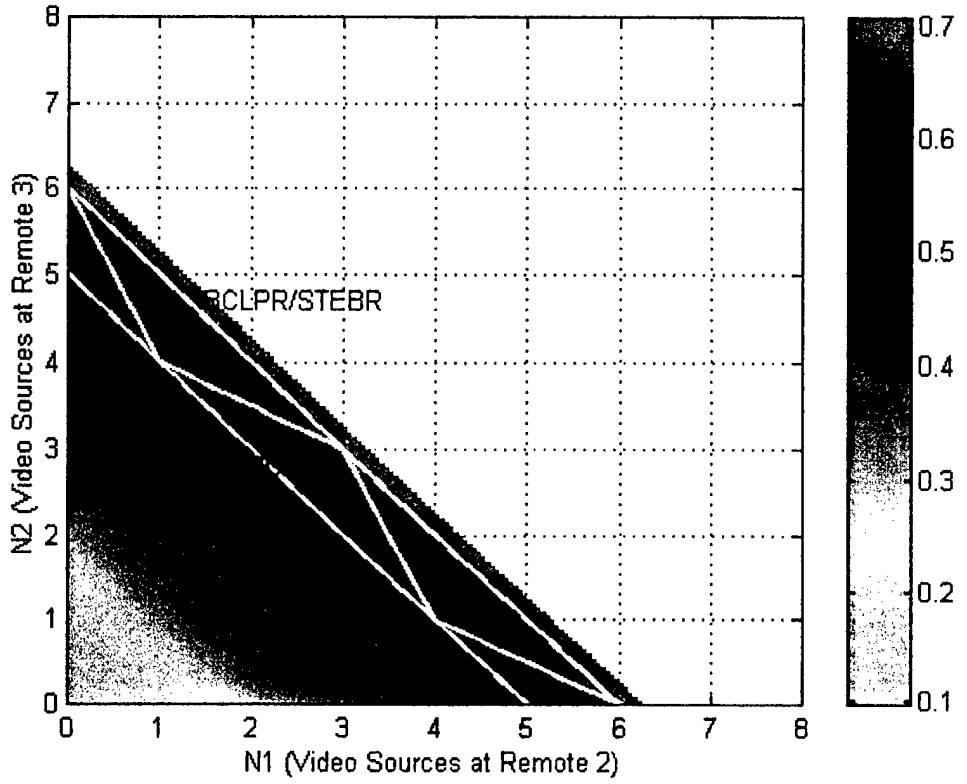


Figure VII.11: Admissible Region and Normalized Throughput for Complete Remote Status, Scenario 5

3. Comparison between Partial- and Complete-Status Cases

In Figures VII.12 through VII.16, we present the results of partial- and complete-status cases together by combining the corresponding individual plots. The subscripts P and C are used to denote the cases of partial and complete status reports, respectively. The plots of peak-rate allocation apply to the channel capacity available in the partial-status case only since clearly no status reports from the remotes are required.

Performance improvements are obtained by all algorithms with complete remote status in Scenarios 1, 2, and 3, where the normalized throughput (in the partial-status case) is smaller than about 0.70-0.75. In Scenarios 4 and 5, where the normalized

throughput (in the partial-status case) is larger than 0.70-0.75, the overhead due to complete status reports by remotes degrades the size of the admissible region and the normalized throughput.

In the wireline case, we have witnessed maximum normalized throughput levels of at least 0.75 for static allocation and 0.85 for other schemes given any constellation of sources. The performance in the wireless network is upper bounded by the values obtained in the wireline case due to discrepancies in allocation as explained in Section VI.D. Whenever the normalized throughput in the mobile channel is small, the allocation decisions made by the schedulers in the partial-status case are far from being optimal. This is attributed mainly to the lack of remote-status information, e.g., queue occupancy or the number of cells discarded thus far at the PMAC, which is essential for efficient operation of the scheduler. In such cases, the overhead due to complete status reports by remotes is considered worthwhile. On the other hand, if the network achieves a large normalized throughput (in the partial-status case), allocation of 18.2% of the channel capacity for complete remote-status transmissions becomes non-beneficial. In some cases, where the normalized throughput (in the partial-status case) is very large (0.80 or more) as in Scenario 5, the overhead due to complete status reports leads to degradation in performance, even below that obtained by peak-rate allocation. (Compare the curve of peak-rate allocation in Figure VII.6 to the curves of the other schedulers in Figure VII.11.)

Let us further examine these conclusions. From Chapters V and VI, the available channel capacities in the partial- and complete-status cases are 1833 and 1500 cells/sec, respectively. Refer to the normalized throughput in a wireline network having a capacity of 1833 cells/sec as $\overline{S}_W\big|_{1833}$, and to the normalized throughput in a wireless network having capacities of 1833 and 1500 cells/sec as $\overline{S}_M\big|_{1833,C}$ and $\overline{S}_M\big|_{1500,C}$, respectively, using complete status. Typical values for $\overline{S}_W\big|_{1833}$ as seen in Section V.I are in the range of 0.90 to 1.00. Due to the nature of the scheduling problem, for the same scheduler, the normalized throughput in a mobile network is upper bounded by its counterpart in a

wireline network for the same capacity. We approximate the former to be about 95% of the latter:

$$\bar{S}_M\big|_{1833,C} \cong 0.95 \times \bar{S}_W\big|_{1833} = [0.86, 0.95].$$

As the available channel capacity increases, it is known that the normalized throughput can only increase due to larger multiplexing gain. Thus,

$$\bar{S}_M\big|_{1500,C} \leq \frac{1500}{1833} \times \bar{S}_M\big|_{1833,C} = 0.82 \times \bar{S}_M\big|_{1833,C}.$$

We then have

$$\bar{S}_M\big|_{1500,C} \leq 0.82 \times [0.86, 0.95] = [0.70, 0.78].$$

Consequently, if the normalized throughput in the network operating with partial remote reports is larger than a value in the range 0.70 to 0.78, applying the complete-status technique would only harm the performance. For smaller values, the complete-status scheme is likely to improve the performance.

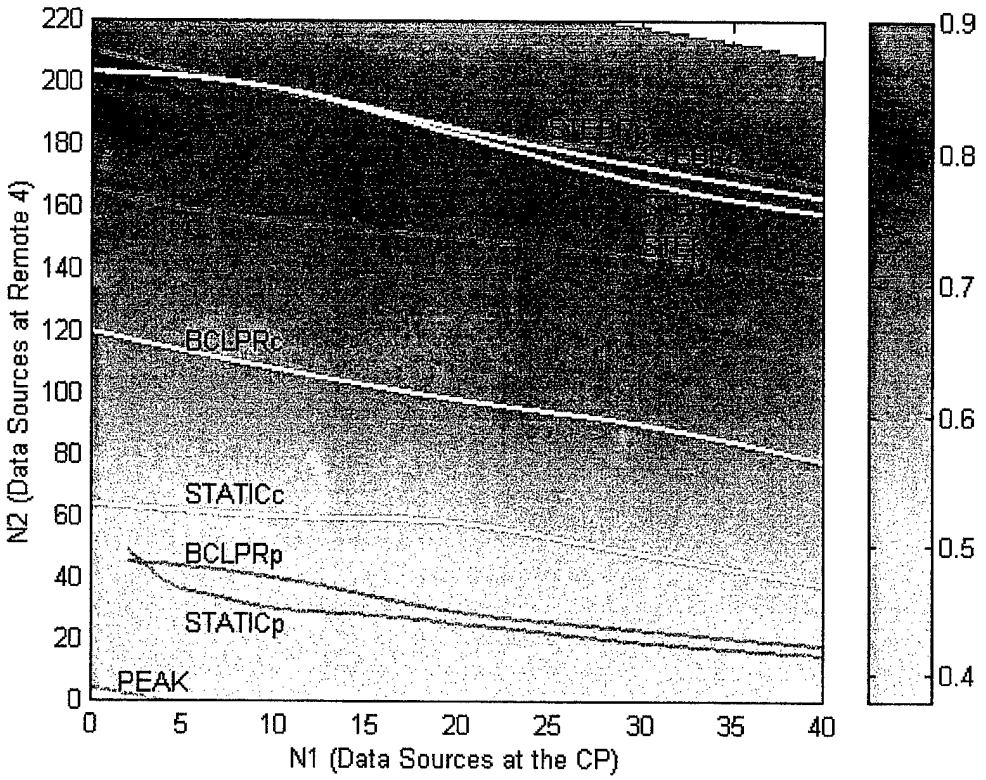


Figure VII.12: Admissible Region and Normalized Throughput for Partial- and Complete-Status Cases, Scenario 1

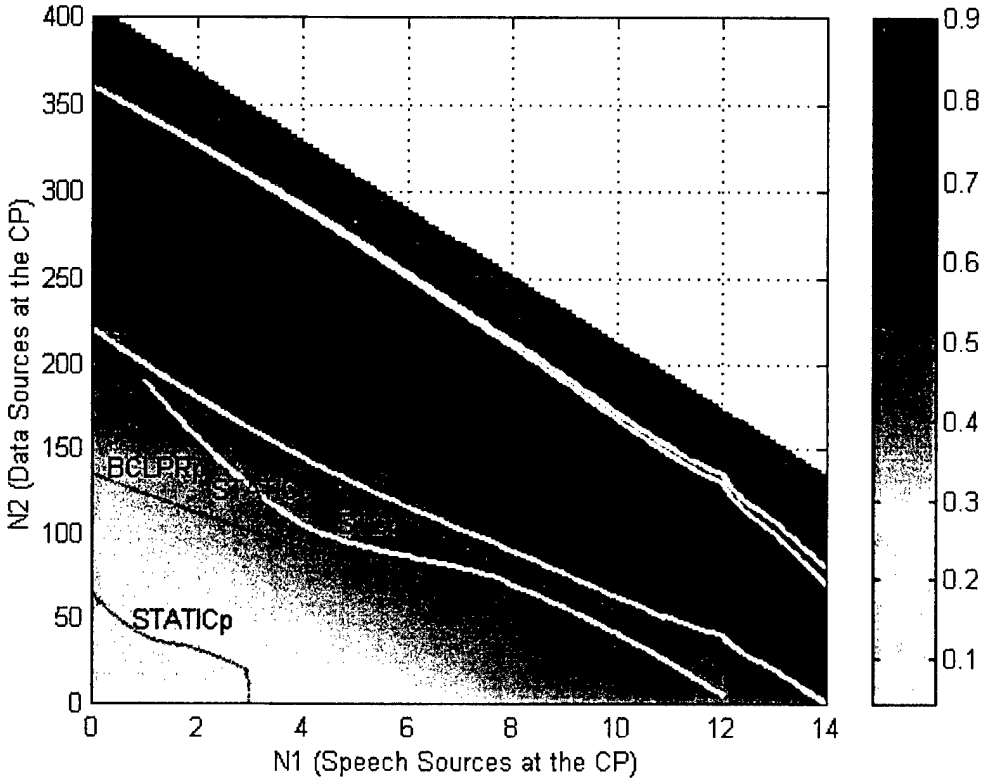


Figure VII.13: Admissible Region and Normalized Throughput for Partial- and Complete-Status Cases, Scenario 2

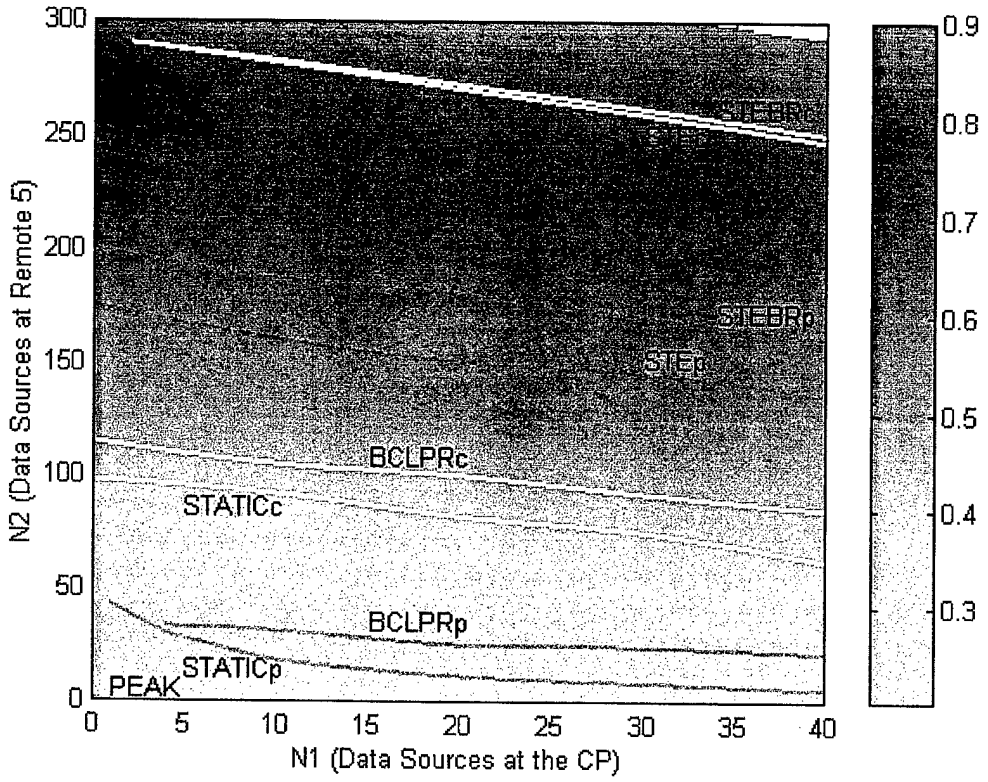


Figure VII.14: Admissible Region and Normalized Throughput for Partial- and Complete-Status Cases, Scenario 3

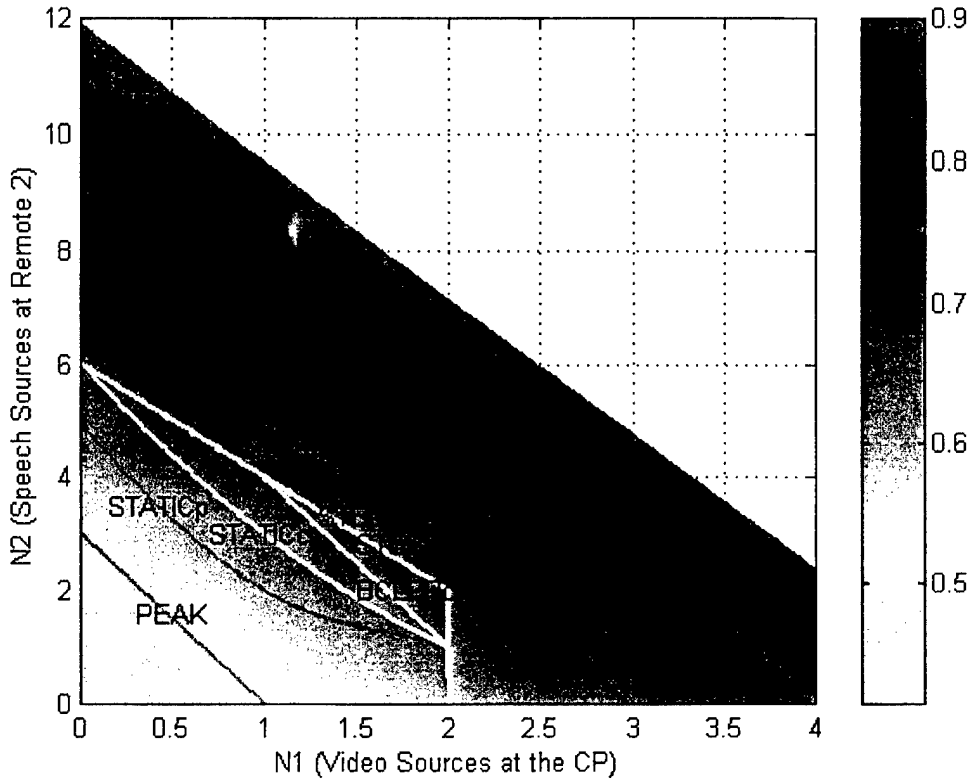


Figure VII.15: Admissible Region and Normalized Throughput for Partial- and Complete-Status Cases, Scenario 4

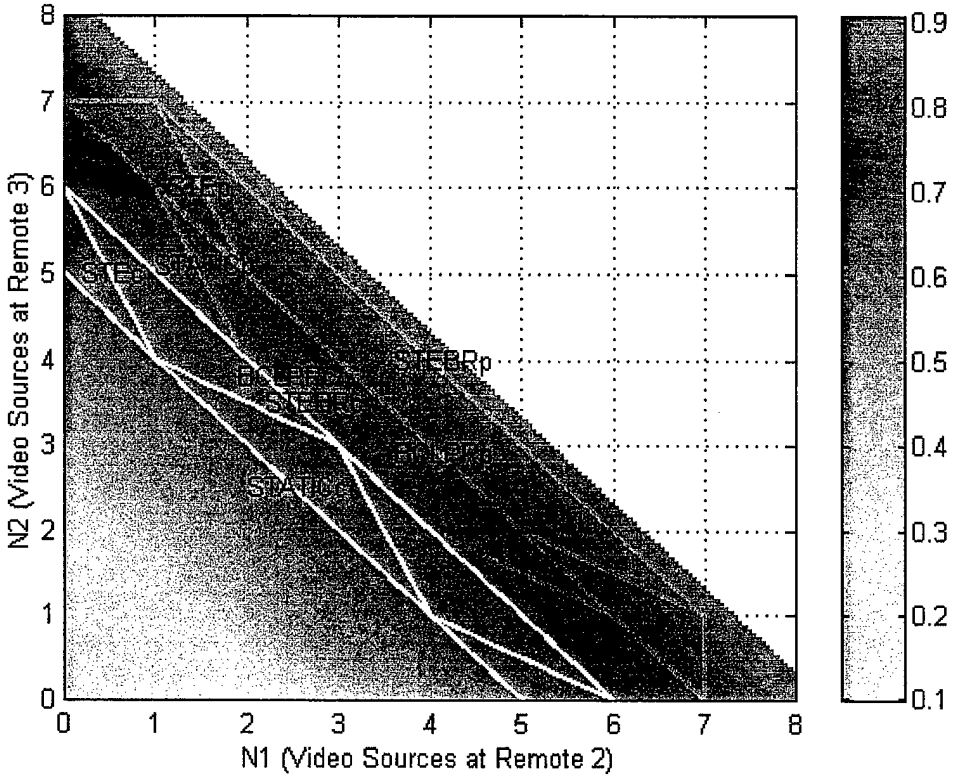


Figure VII.16: Admissible Region and Normalized Throughput for Partial- and Complete-Status Cases, Scenario 5

D. DISCUSSION

We have considered five different scenarios, representing sample cases in the battlefield having different network configurations and/or active traffic sources. The behavior of the scheduling algorithms was investigated as a function of the network load under partial and complete information provided by the remotes to the CP. The load input to the network comprised a constant load and a variable load generated by two independent stations and/or traffic classes.

The STEBR algorithm was shown to outperform all other algorithms under both remote report techniques in all five scenarios. This outcome reinforces the results obtained for the wireline network. The STE algorithm provided improved performance over BCLPR in most cases, unlike the wireline case, where BCLPR was slightly better. This is attributed to the quality of information made available to the scheduler and the mechanism used in the remotes to provide this information. In STEBR and STE, remotes reported the exact number of information slots required by their sources on the next frame. No cell was included in such a report more than once; thus, the number of transmissions in the contention channel became very small leading to very few collisions. As a result, a large portion of the slots allocated by the CP on the uplink was utilized by remotes for transmission of their waiting cells. In BCLPR (and static allocation), remotes contended on the channel whenever they were non-empty or no allocation on a given frame was made for any of their sources. The numbers of transmissions and collisions in the contention channel thus increased, resulting in higher loss rate for allocation request messages. Consequently, BCLPR and static allocation made only limited use of the remote occupancies reported via the allocation request messages, leading to inferior performance.

In the partial-status case, as the number of remotes in the scenario increased, the multiplexing gain achieved in each remote dropped, causing a reduction in the normalized throughput. In the complete-status case, the dominant factor was the number of active sources in the network; as this number became large (several hundred data sources), almost-maximum-possible normalized throughput was obtained due to high level of multiplexing gain. The normalized throughput varied along the boundaries of the admissible regions for scenarios with variable load comprising sources from different traffic classes. As the number of sources having less-stringent CLP requirements increased at the expense of sources having stricter constraints, larger normalized throughput was achieved. In scenarios where the sources of the variable load were of the same traffic class, the normalized throughput remained at the same level along the boundaries of the admissible regions.

Comparison between the performance of the schedulers using partial and complete status reports showed dependence on the level of the normalized throughput. For partial-status cases, where it was larger than about 0.70, the overhead incurred to transmit complete status of all remote sources degraded the performance. On the other hand, when the normalized throughput in the partial-status case was smaller than 0.70, this overhead helped to increase the normalized throughput.

A hybrid approach can be used to best utilize the available channel capacity in mobile integrated services networks. Whenever a source requests admission (release), the scheduler at the PMAC calculates (using Equation (III.1)) the expected normalized throughput in the channel, assuming that the source is admitted (released). The mode of operation is then set according to the following policy. The network starts by using complete status updates from the remotes. As more sources become active and the normalized throughput exceeds $0.72+\Delta$ (see Figure VII.17, where Δ is 0.03), a transition is made to partial updates. The networks returns to complete status reports when the normalized throughput drops below $0.72-\Delta$. This hysteresis behavior reduces fluctuations between the two operation modes near the "knee" value (0.72) of the normalized throughput. A disadvantage of the hybrid scheme is the increased complexity in network management and control; transitions between the operation modes must be performed smoothly to avoid harsh degradation in the QoS provided to existing sources. The performance of the network using the hybrid approach is expected to be the maximum achieved by the partial- and complete-status cases individually.

To validate the hybrid technique, Figure VII.18 presents the normalized channel throughput in the complete-status case as a function of that in the partial-status case. The levels of the normalized throughput in both cases are marked in Figure VII.18 as colored blocks corresponding to the simulation results using different schedulers. The diagonal line represents the case in which the normalized throughputs in both cases are the same. We can observe that whenever the normalized throughput in the partial case is larger than about 0.70, a performance improvement is obtained in the partial-status case compared to

the complete-status case and vice versa. Ideally, no colored blocks would be located in the shaded regions; this is almost the case here.

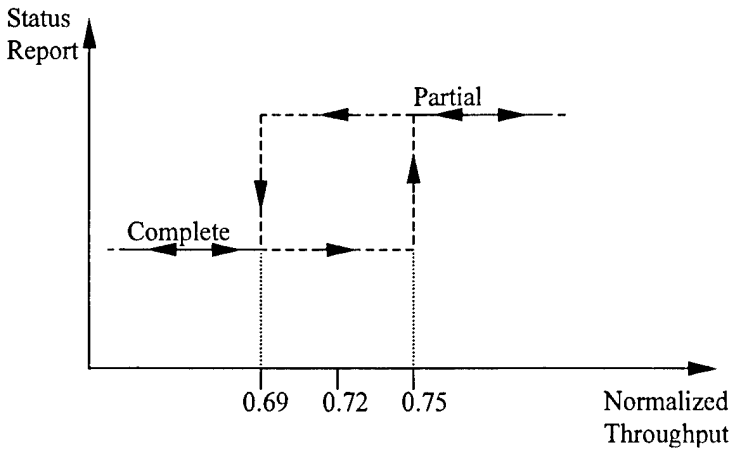


Figure VII.17: A Hybrid Scheme for the Scheduler Operation

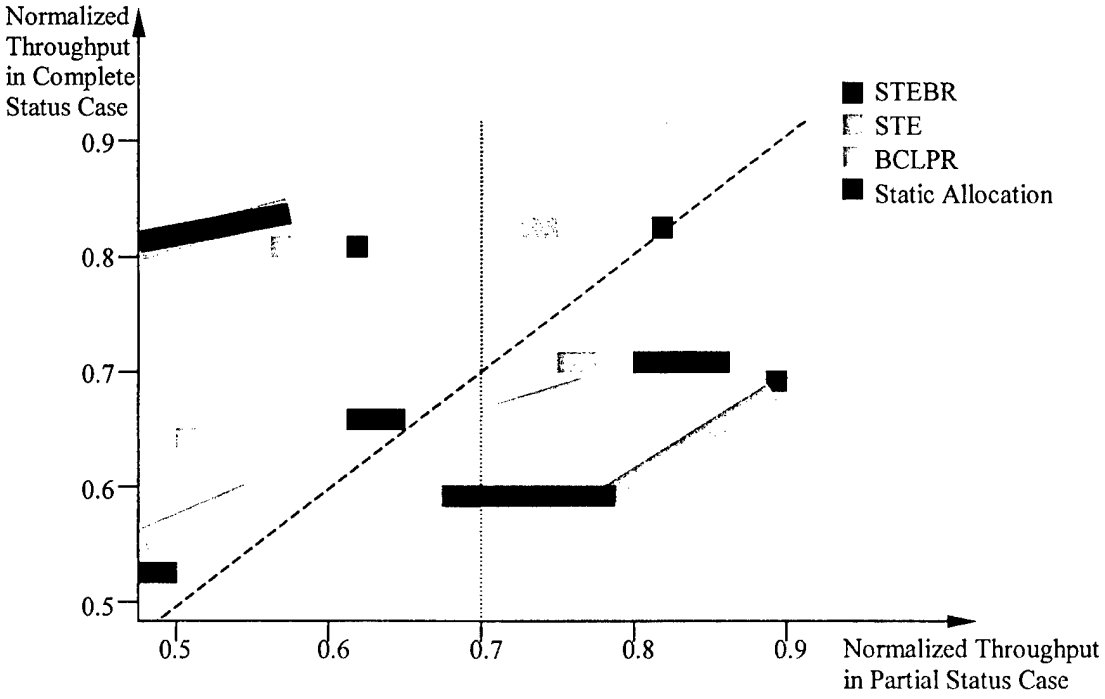


Figure VII.18: Normalized Throughput in Partial and Complete Status Reports

VIII. CONCLUDING REMARKS

A. CONCLUSIONS

The goal of this work is to address issues related to the design of a wireless integrated services network with emphasis on a tactical framework. Such a network is expected to seamlessly integrate with a wireline network via a line-of-sight or satellite link to enable exchange of traffic with the external world. We have proposed an ATM-like protocol architecture for the mobile network; this protocol architecture is an extension of schemes proposed in the literature. A mobile control unit handling the signaling relevant to the wireless media was thoroughly described. This controller manages call admission and release, registration and disconnection to and from the network, distinct user priorities, and translation between different addressing notations.

A MAC scheme for mobile integrated services networks was proposed. The protocol segments the time into contiguous frames, each of which includes downlink (CP-to-remotes) and uplink (remotes-to-CP) channels. The proposed scheme requires slot reservations prior to allocation by the CP for cell transmissions. Together with a piggyback mechanism, where future allocation is requested in the cell header, the scheme guarantees a very small number of collisions in the contention portion of the uplink channel. The reservation scheme is essential to support multimedia services having distinct QoS requirements.

Traffic models for low-bit-rate applications, suitable for low-capacity channels, such as a multiple-access (macrocell) wireless network, were presented. A new speech model based on measured statistics of a two-directional conversation was proposed. A histogram-based model having at least eight bins, which was proven in the literature to accurately represent a variable-bit-rate, real-time, video stream, was used to model a 64-kbps video source. A new hybrid model for data sources was proposed as well. The model represents the bursty nature of two typical multimedia applications: frequent text

transfers and less frequent image transmissions. Expansion from a single source to the case of multiple sources was discussed for all traffic classes.

The problem of scheduling in wireline integrated services networks was thoroughly addressed and new algorithms proposed. The static-allocation algorithm assigns fixed, pre-determined capacity to the sources; an analytical scheme to obtain the required capacity for homogeneous sources based on the Markov-chain characteristics of their class was provided. We have found a necessary condition required by an optimal algorithm: for all sources, the ratios between the experienced loss and the allowed loss must be balanced over a long period to a value approaching 1 from below. The BCLPR algorithm satisfies this condition but ignores the cell deadlines completely; it makes service decisions based only on the CLPRs. The STEBR algorithm, proposed here for the first time, utilizes the advantage of the earliest-deadline-first concept while satisfying the necessary condition. Cells are scheduled for service according to their deadlines unless loss is expected in the future using the STE policy; then, the loss experienced by the sources thus far is taken into account to achieve an overall least-cost decision. A theorem stating that STEBR makes an optimal decision at each service slot given that no information about future cell arrivals is available was proved. Simulation results were shown to support the theorem. For an outgoing link capacity of 1833 cells/sec (~775 kbps), STEBR admitted more sources and yielded larger normalized channel throughput (by up to 4%) than STE. The run-time complexity of STEBR is $|\{S_S\}| \times O(N)$ compared to $O(N)$ for STE.

The mobile network presents a case of distributed queues at the CP and in the remotes, making the scheduling more involved in relation to wireline systems. The exact status of the remote sources, essential for efficient channel allocation by the CP, is known only at their originating stations. This increases the probability of inefficient channel allocation, resulting in reduction of the number of admitted sources and the channel throughput. Based on the schedulers discussed for the wireline network, we developed corresponding algorithms for operation in the wireless network. Two cases were considered. In the first case, only partial status reports by the remotes (SMACs) were

made available to the central scheduler (PMAC). The partial information was obtained whenever a remote queue became non-empty or as piggyback data within a transmitted cell. The overhead required for obtaining partial information was less than 2%. In the second case, about 20% of the channel capacity was devoted for gathering complete remote-status information during every frame. The availability of almost-complete remote information required modifications to the scheduling algorithms developed for the partial case.

The behavior of the scheduling algorithms was investigated as a function of the mobile-channel load in five representative scenarios. The STEBR algorithm was shown to outperform other algorithms under both partial and complete status reports in all scenarios, strengthening the results obtained in the wireline queue. STE provided improved performance over BCLPR in most cases, and BCLPR was better than static allocation. BCLPR and static allocation performed better in both partial- and complete-status cases as the number of remotes in the scenario increased. For STEBR and STE, this outcome was valid in the partial-status case only.

As more sources having less-stringent CLP requirements were active at the expense of sources having stricter constraints, larger levels of normalized throughput were achieved. Performance of the schedulers using partial or complete status reports depended on the value of the normalized throughput. Complete-status mechanism was preferred in all cases in which this value was smaller than 0.70-0.75; partial status was sufficient for values larger than 0.70-0.75. A hybrid approach that makes use of this outcome was proposed to best utilize the available channel capacity under all possible levels of network load.

We summarize here the contributions made in this dissertation to the topics of wireless integrated services (ATM-like) network architecture, MAC design for this network, and channel-allocation schemes for (wireline and) wireless networks:

- Detailed design of a MAC protocol for mobile ATM networks, including support of in-band control and signaling channel.
- Novel, low-bit-rate, Markovian, traffic models for bi-directional speech conversations and mixed short- and long-burst data sources.

- Analytical method to obtain the minimum required capacity (static allocation) in a single-queue single-server homogeneous system with guaranteed loss and delay QoS requirements.

- Statement of a necessary condition for optimal scheduling in wireline ATM networks; development of the BCLPR algorithm that satisfies the condition but not optimal.

- Development of a new scheduling algorithm (STEBR) that makes optimal intermediate decisions in the wireline case in time $O(N^2)$, where N represents the number of cells in the queue at the time of decision; proposed implementation in time $O(N)$.

- Design and implementation of channel-allocation schemes for wireless channels.

B. FUTURE RESEARCH

This section details the issues for future research related to the work presented in the dissertation.

1. Mobile Architecture Improvements

For remote-to-remote or multiple-remote connections, the MAC protocol presented here can be adapted to utilize direct radio connections among the remotes. The CP relays information cells on all these connections. No direct remote-to-remote information transfers are allowed in this work. This constraint is conservative because radio connections between some remotes are likely to exist. Channel throughput can be improved by eliminating cell relaying by the CP in remote-to-remote connections, if the remote destination could successfully receive the information directly. For example, using the proposed MAC structure in this work, if a destination remote realizes that the CP has not yet relayed a cell that the remote has successfully received via a direct link, it can mark this fact on the uplink control subchannel to avoid future transmission of the cell by the CP.

In Chapter II, we proposed two possible architectures for the mobile network. A configuration in which the CP contains both mobile network coordinator (MNC) and

local traffic handler (LTH) has been chosen for use in the dissertation. The network configuration in which the MNC and LTH are separate units can be investigated. Given a central coordinator, where all stations (CP and remotes) have equal importance from a communications point of view:

- Establish a MNC node with a radio transceiver, independent of the CP.
- Equip the MNC with a LOS/satellite radio transceiver and establish a LOS/satellite link between the MNC and the CP.
- Develop a channel-allocation algorithm at the MNC to take the delays incurred on the LOS/satellite link into account for connections involving external sources.

Different levels of priorities, representing users in various hierarchical ranks or having instantaneous significance in the battlefield, are necessary in a military network. High-priority users can affect the existing connections in two major ways. First, active lower-priority users might be disconnected in order to reallocate channel resources to higher-priority users, in case the necessary resources are not available for all connections. Second, high-priority users may enjoy improved QoS, such as shorter delays or larger bandwidth allocations, although low-priority users will still be guaranteed their QoS requirements. The issue of priorities must be investigated in order to achieve a suitable scheme for military mobile integrated services networks. After defining the priority functionality in such networks (i.e., the number of priority levels, metrics for priority quality of services, etc.), several priority schemes need to be investigated and the priority control unit developed.

2. Static Allocation in Wireline Networks

We have defined three different classes of traffic (speech, video, and data). More than one source belonging to more than one service class can be active within a given station simultaneously. Thus, the problem of (static) capacity allocation may be expanded to include the multiplexing of heterogeneous-class sources in a queue. The problem is to find the minimum required capacity for a single-queue single-server system such that the (distinct) QoS requirements of the sources from different classes are maintained.

Some of the connections in the mobile network are of remote-to-remote type. In these cases, a cell that is transmitted from a station experiences queueing delay in two queues (in the remote and at the CP) in addition to the propagation delays. The QoS requirements (CLP and maxCTD) are usually specified on an end-to-end basis. The problem is to find optimal capacity allocation for the two links (C_1 and C_2) such that the end-to-end QoS requirements are met, and the sum C_1+C_2 is minimized (see Figure VIII.1). (We wish to minimize the sum since both capacities share the same resource, the network transmission time.) Actually, the existence of a tandem of queues in the network virtually doubles the number of traffic characterizations: the original three classes with their QoS requirements, and the same three classes with QoS requirements that are one half of the original demands. A comprehensive solution to the tandem-of-queues problem is not currently available in the literature [81].

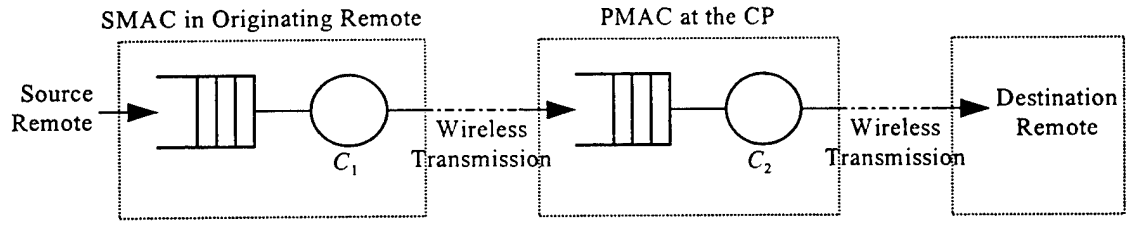


Figure VIII.1: Remote-to-Remote Connection Represented as Two Serial Queues

A discussion of a Poisson arrival process to a tandem of two queues is presented in [99]. If one assigns $C_2 > C_1$, then the second server is idle over a large portion of the time. The optimal solution provided in [99], which minimizes the mean end-to-end delay of the cells, is $C_2 = C_1$. However, the analysis does not involve any QoS constraints (it assumes infinite-size queues), thus can serve as an intuitive solution only. Since a specific loss is allowed end to end, the arrival rate into the first queue along the source-destination path is larger than that into the second queue. Thus, an expected result would be $C_1 > C_2$.

3. Predictive STEBR Algorithm

A global optimal scheduling algorithm for a (real-time) single-queue single-server system does not exist due to the dynamic nature of the system. Cells arrive into the queue on line, between service instants. The STEBR algorithm supplies the optimal scheduling for the case where no information about the future arrivals is available (or equivalently, no more cells are allowed to enter the queue after the point of decision).

Using *a priori* knowledge of the statistics of the sources, one can predict future arrivals of cells from the sources and improve the decision on which cell to serve at each decision time. One can predict, for example, the number of arrivals from each source within the next n service slots to make a decision.

4. Operation in CDMA Networks

Code division multiple access (CDMA) networks are of particular interest in the military community. Such networks typically include separate forward (downlink) and reverse (uplink) channels. The modem (physical layer) in each station of a CDMA network comprises a spreading element (pseudo-noise (PN) generator) in the transmitter and a despreading element in the receiver. The network is designed such that several PN sequences are available for use; thus, if multiple transmitters transmit at the same time using different PN codes, no collision occurs (depending on factors, such as physical distances, transmitter powers, antenna gains, etc.).

The CP in a CDMA network assigns distinct PN codes to the remotes within the network. Several PN-code-allocation schemes are possible. The simplest would be static allocation of a PN code per remote at the beginning of the operation: when the remote joins the network, it receives a code from a PN-code bank. When all the codes in the bank are exhausted, multiple use of codes is allowed. It is clear that, as the number of remotes increases beyond the number of available distinct PN codes, collisions may occur in the channel. More sophisticated allocation techniques can involve dynamic allocation of the PN codes, based on the information known at the CP. A MAC architecture demonstrating such a technique is shown in Figure VIII.2. Both forward and reverse channels are slotted into frames and shifted in relation to each other. Each frame

contains control and information subchannels. At the beginning of every frame on the forward channel, the CP informs the remotes (just prior to the beginning of the frame on the reverse channel) of sources that are assigned a slot for transmission on the upcoming frame and the corresponding assigned PN codes. Clearly, the allocation techniques developed in the dissertation for the TDMA-TDD network can be adapted for channel (PN-code) allocation in the CDMA network. One limitation to be taken into account is that a given transmitter cannot transmit more than one cell at a given time.

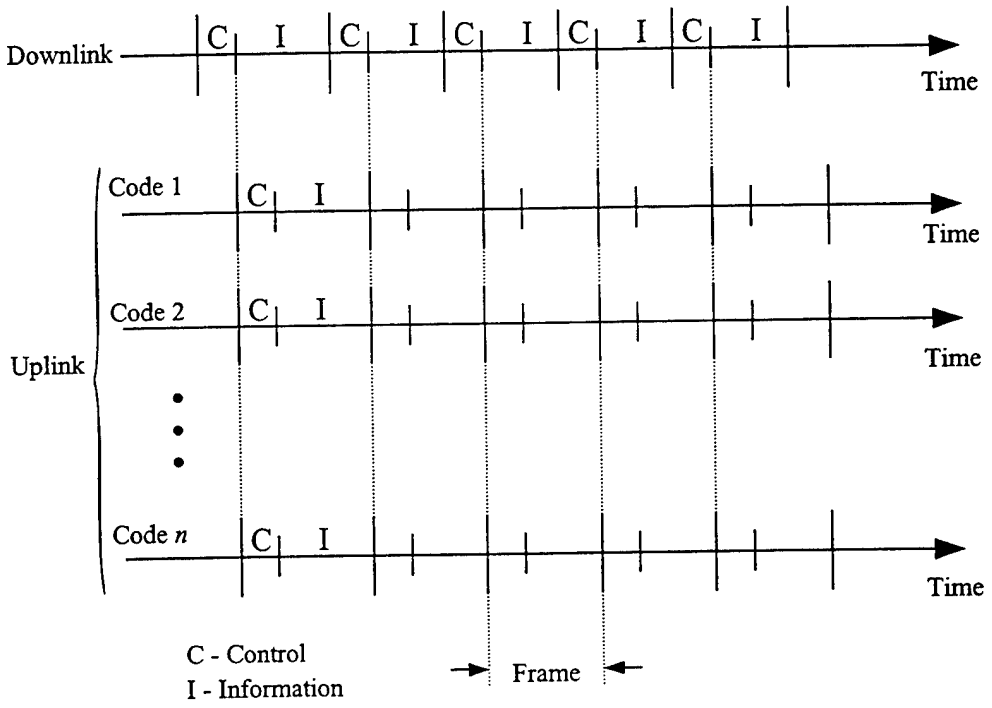


Figure VIII.2: MAC Architecture for Dynamic Allocation of PN Codes

5. Operation under Noisy Channel Conditions

A physical radio channel is characterized by several impairments that affect the quality and strength of the received signals. The impairments affect a frame of data transmitted over a mobile network in one of two forms: lack of reception due to a long burst distorting the preamble sequence or corruption of the frame due to noise.

A DLC is required on the top of the MAC in order to cope with realistic noisy channel conditions. It requires defining the mobile radio channel impairments relevant to the network and developing a channel model [26] [51]. Investigation of the effects of the impairments then follows in terms of reduction in quality of service for single-hop and two-hop connections. Development of an error-control mechanism as part of the DLC sublayer is essential. Error-control protocols may include segmentation of cells into frames, addition of FECC, transmission of positive and negative acknowledgements, etc. Simulation of the network with the channel model would then determine the boundary of the admissible region and the channel throughput under practical operational conditions.

Analytically, the queueing system at every remote node under noisy channel conditions can be simplified as shown in Figure VIII.3. A transmitted cell over the wireless channel may be considered as either successfully received by its destination or corrupted and lost due to channel noise. A corrupted cell reenters the queue with a probability P_{ERROR}, which may be transmission-length dependent and/or time dependent. Such a cell is reenqueued and retransmitted according to the DLC/MAC disciplines. Since the QoS requirements remain the same, the reenqueued cells require allocation of larger capacities to each source/node than that in the error-free channel. Consequently, performance degradation is expected.

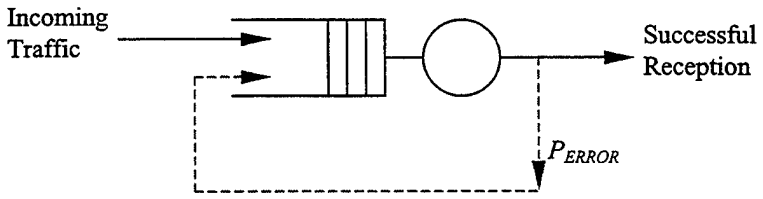


Figure VIII.3: Queueing Model of a Mobile Node in a Noisy Channel Environment

APPENDIX A. REPRESENTATIVE MOBILE DATABASE AND PROCESSES

This appendix extends the discussion developed in Chapter III in relation to the MAC protocol over mobile integrated services networks. The appendix details a representative mobile-station database, intra processes within the station, and inter processes between distinct stations.

A. REPRESENTATIVE MOBILE DATABASE

This section aims at describing a possible database require to be held at the CP and in the remotes for mobile operation. Note that what is detailed here is *in addition* to any other tables, lists, variables, etc., which are required for the regular operation of a (wireline) ATM network.

1. Database at a Remote Station

A remote in the mobile network needs to maintain the following elements:

- *Remote_Operational_ID*: This variable contains the operational ID of the remote and uniquely identifies it. This identifier is assigned to the remote throughout the lifetime of its operation. (In some cases, though, the structure of units change and forces may be assigned other identifiers.)

- *Unit_Addressing_List*: The list details all the possible operational ID's of units, in which the remote may communicate with, plus their corresponding (ATM-oriented) addresses. The latter contain only the addresses of the nodes and not the sources within the nodes; generally, the source addresses are determined by the applications at call setup rather than off line. Each entry within the list, as shown in Table A.1, contains an operational ID and an ATM address. After the remote is turned on, this list should be available for use, usually by means of a file on a hard drive or a hard copy (the translation in the latter case is then made manually by the user in real time, prior to call setup).

Nº	Operational ID	ATM Address
...		
i	*Operational ID$_i$*	*ATM Address$_i$*
...		

Table A.1: *Unit_Addressing_List* Structure

- *Remote_MSI*: This variable contains the mobile signaling identifier of the remote within a given network. The remote is assigned this number after a successful registration within the network, and it is used for ATM and mobile signaling identifications.

- *Keep_Alive_Timeout*: This variable serves as a software timer. It measures the time left for the remote to send a KEEP_ALIVE message. The timer is set whenever the remote transmits any type of message. If the timer expires, it is reset and a KEEP_ALIVE message is generated and sent from the station.

- *Local_Connection_Table*: This table contains information about all the connections that the remote takes part in. It holds data about the current identifiers of the remote within each connection. Note that the information stored is of the remote only, even if the connection has several other parties. Each entry in the table (see Table A.2) includes (ATM-oriented) VPI/VCI and their corresponding (MATM-oriented) MVCI/IDVC. To be completely accurate, we had to add to each entry also a call-reference field, which is used for call-setup/call-release synchronization between the CP and the remotes. However, since the field has only a local importance and actually is nothing but another representation form of the MUI notation, we disregard it in the discussion on the stations database and the communication processes thereafter.

Nº	VPI/VCI	MVCI/IDVC
...		
i	*VPI$_i$/VCI$_i$*	*MVCI$_i$/IDVC$_i$*
...		

Table A.2: *Local_Connection_Table* Structure

2. Database at the CP

The CP maintains the following database elements:

- *Unit_Addressing_List*: A similar list to the one held by a remote, though it usually contains more entries (the CP often communicates with more units than a remote).

- *Registration_Table*: The CP manages a list of remotes within its network. In a given time, this list contains the registered remotes in the network with their corresponding MSIs. Each entry in the table, as listed in Table A.3, contains the operational ID of a remote, its MSI, and the time at which a transmission must be received from it before the remote is considered as disconnected. The latter is necessary for improper disconnection recognition purposes, as discusses in Chapter III.

№	**Operational ID**	**MSI**	**Disconnection Time**
...			
i	*Operational ID$_i$*	*MSI$_i$*	*Disconnection Time$_i$*
...			

Table A.3: *Registration_Table* Structure

- *Available_MSI_List*: A list of available mobile signaling identifiers to be used for ATM and mobile signaling in the mobile network. When a remote registers in the network, the CP allocates for it one of the available MSI from this list.

- *Available_MVCI_List*: A list of available mobile connection identifiers to be used for new connections in the mobile network. When a new call is set up in the network, the CP allocates for it one of the available MVCI from this list.

- *Local_Connection_Table*: This table is similar to the one held by a remote and includes the local connections in which the users/sources within the CP participate.

- *Network_Connection_Table*: The table contains information about *all* the connections that take part in the mobile network, i.e., connections that involve at least one remote, in a given time. It holds data regarding the current identifiers of all the parties within each connection. An entry in the table, shown in Table A.4, includes the class type of the connection and, for each party, the (full) ATM address, the associated MVCI/IDVC, and the port in which the source can be reached with. (However, we require the MVCI/IDVC field to be sufficiently large to be able to include ATM-oriented VPI/VCI as well. The reason for that, is the case of a connection that involves external source(s) as explained later.) The port field may get the values LOCAL for sources within

the CP, REMOTE for remote sources or BACKBONE for external sources. The class-type field may get one of the values SPEECH, VIDEO, and DATA. This value implies the QoS requirements of the connection for the parties on the mobile and/or stationary segments of the connection, which are pre-defined in the system; thus, the field is used by the MAC for appropriate channel allocation. As mentioned earlier, the database reflects only the connections in the mobile network. (An additional field per party per connection, called status, may be added; it receives one of the values – SETUP, ACTIVE, and RELEASED – according to the current stage of the party. The purpose of the status field is to allow flow of information cells (not ATM signaling cells) only for active connections. We, however, do not include the field in the table and in the discussion.)

Nº	**Class Type**	**Party 1**			**Party 2**			**...**
		ATM Address	**MVCI/ IDVC**	**Port**	**ATM Address**	**MVCI/ IDVC**	**Port**	
...								
i	*Class Type*$_i$	*ATM Address*$_{i1}$	*MVCI*$_{i1}$/ *IDVC*$_{i1}$	*Port*$_{i1}$	*ATM Address*$_{i2}$	*MVCI*$_{i2}$/ *IDVC*$_{i2}$	*Port*$_{i2}$	
...								

Table A.4: *Network_Connection_Table* Structure

- *Class_Type_Table*: The table is used for translation between a limited number of pre-defined service classes and their corresponding QoS requirements (see Table A.5). Each entry in the table contains the class type of a connection and the CLP and maxCTD associated with it for remote and external sources (these appear in Table A.5 with subscripts R and E, respectively). The remote parameters relate to the QoS requirements over the radio channel, and the external parameters relate to those on the path from the CP to the source outside the mobile network (i.e., over the backbone link).

Nº	**Class Type**	**Mobile Source**		**External Source**	
		CLP_R	**$maxCTD_R$**	**CLP_E**	**$maxCTD_E$**
...					
i	*Class Type*$_i$	CLP_{Ri}	$maxCTD_{Ri}$	CLP_{Ei}	$maxCTD_{Ei}$
...					

Table A.5: *Class_Type_Table* Structure

B. INTRA- AND INTER-STATION PROCESSES

This section aims to combine the components of the MAC discussed in Chapters II and III: the architecture described in Chapter II and the representative database of the previous section. We wish to describe in detail the intra- and inter-station processes in the mobile network. This summaries the whole operation of the different layers within the mobile station, including (internal) exchange of information between layers in the station and (external) between stations, under various situations. For simplicity, we assume an error-free communication channel.

We describe possible individual scenarios within the network and their effect on the various layers of each of the remote stations involved. This mainly involves the internal processes within a node (update of local variables, etc.) and the inter-node messages that follow thereafter. It is believed that such a description provides a clear understanding of the communication layers in the mobile architecture and their integration with the (partially extended) ATM layers. The section concludes with a description of communication processes, which involve various connections between a remote source and an external source.

1. CP Power-On

When the CP is turned on, it performs the following:

- The ID-assignment controller initializes its *Unit_Addressing_List*.
- It sets in the *Available_MSI_List* the numbers 1, 2, 3, ..., 63.
- It sets in the *Available_MVCI_List* the numbers 4, 5, 6, ..., 63.
- The MAC starts transmitting the MAC frame headers, inviting remotes to register in the network.

2. Remote Registration

When a remote is turned on or moves to a new network, it performs the following:

- The ID-assignment controller initializes its *Unit_Addressing_List*.
- It also initializes its *Remote_Operational_ID*.

- The registration controller sets a timer to the *Keep_Alive_Timeout*.

- It passes a REGISTER_REQUEST message to the DLC for transmission. The *Remote_Operational_ID* is used to identify the remote.

- The MAC listens to the radio channel until a frame structure sent by the CP (using the channel indicator and the frame header) is recognized. It synchronizes the remote's timing mechanisms to the beginning of the frame.

- The MAC sends the REGISTER_REQUEST message using contention on the uplink control subchannel.

When the message is successfully received at the CP and the registration is approved by the registration controller, the CP performs the following:

- The ID-assignment controller assigns an available MSI to the remote from its *Available_MSI_List*. This MSI is marked as occupied.

- It adds an entry to the *Registration_Table* with the remote's operational ID and the allocated MSI. The disconnection-time field in the table is set to a pre-defined value.

- It passes a REGISTER_REPLY message to the MAC with the operational ID and MSI of the remote and its allocated MSI.

When the remote successfully receives the message, the ID-assignment controller updates its *Remote_MSI* about the received allocated MSI.

3. Remote Proper Disconnection

When a remote wishes to leave the network, it performs the following:

- The ID-assignment controller releases all its active connections (if any), appearing in the *Local_Connection_Table*.

- The registration controller passes an EXIT_REQUEST message to the MAC for transmission using *Remote_Operational_ID*.

The CP, after a successful reception of the message, responds as follows:

- The ID-assignment controller erases the entries of the remote from the *Network_Connection_Table* for all participating connections (in case any are still indicated as active). If any of the connections is left with one party only, the corresponding entry at the *Network_Connection_Table* is deleted and the MVCI

associated with the connection becomes available in the *Available_MVCI_List.* (Additionally, a "Release" message is sent to this last party.)

- The registration controller erases the remote from the *Registration_Table* and de-allocates its MSI in the *Available_MSI_List.*

- It sends an EXIT_REPLY message to the DLC with the node's operational ID.

4. Keep-Alive Procedures

The ongoing existence of a registered remote in the network is being continuously tracked by the CP. The registration controller at the remote sets *Keep_Alive_Timeout* to a pre-defined value after power-up and whenever any message is being transferred to the DLC for transmission. Every constant period (say, one second), the controller subtracts this period from the timer, and when it hits zero, the controller passes a KEEP_ALIVE message to the DLC with its *Remote_Operational_ID* and resets the timer.

Whenever the DLC at the CP receives a message of any type from a remote, it so informs the registration controller, and the disconnection-time field of that remote in the *Registration_Table* is reset. In parallel, throughout the operation of the CP, every constant period (say, one second), the registration controller subtracts this period from the timer, and when it hits zero, the controller follows the procedure of remote proper disconnection (but without sending the EXIT_REPLY message). If a MATM signaling message (other than REGISTER_REQUEST) is received by the DLC and the sending remote does not appear in the *Registration_Table*, the DLC responds by a REJECT message to that remote using its operational ID as appears in the received message.

If the DLC at a remote receives such a REJECT message from the CP, it passes the message to the ID assignment and registration controllers. The former clears the *Local_Connection_Table*, while the latter starts with a reregistration process as discussed above.

5. Call Establishment

We describe here the process of a call establishment between two arbitrary remote sources, which we denote as *A* and *B* (assuming *A* initiates the call). New call-setup

processes in other possibilities for the calling and called parties may be obtained from this case since it is relatively more complicated. The case of a remote-to-external source call setup is considered separately later.

The ATM control unit in *A* passes (through the ATM layer) a "Setup" signaling message to the DLC for transmission. The "Setup" message contains (as part of the payload) the calling and called ATM addresses that are read partially or completely from the *Unit_Addressing_List*. The VPI and VCI fields of the message are set to zero. The DLC, using *Remote_MSI*, passes the message to the MAC. The latter contends on the uplink control subchannel and requests a channel allocation.

The MAC at the CP allocates the channel to *A* using its MSI.

The MAC in *A* sends the "Setup" message on the allocated information slot(s).

When the "Setup" message is successfully received by the CP:

- The message is passed on the path from the MAC to the DLC, ATM, and ATM control. Every signaling message is passed to the ATM control unit of the receiving node, even if this node is not the final destination (according to the message content, the ATM control unit decides what action to perform, e.g., pass it to another node, respond by a message to the sender, etc.).

- The ATM control unit exchanges messages with the mobile admission controller, regarding the acceptance of the call into the mobile network. The decision is based on the called address (mobile or not), the traffic descriptors, and the QoS requirements (that are included in the "Setup" message, but must have pre-defined traffic class parameters described in *Class_Type_Table* under CLP_R and $maxCTD_R$ columns).

- If the call is accepted:

    - The ID-assignment controller allocates a MVCI to the call from the *Available_MVCI_List* and assigns IDVCs for *A* and *B*. This information as well as the ATM addresses of *A* and *B* are copied onto the *Network_Connection_Table*. The port fields of *A* and *B* are marked as REMOTE.

    - The MAC is informed by the mobile admission and ID-assignment controllers about the allocated MVCI to the new call and its class type (i.e., implicitly its traffic descriptors and QoS parameters).

- ♦ The ATM control unit passes a "Proceed" message, destined to *A*, to the DLC. The VPI field of the message is set to zero and the VCI contains the MSI of *A*. This will be used by the MAC at the CP to identify the MAC address of *A* in the mobile network.

- ♦ It also passes a "Setup" message, destined to *B*, to the DLC. The VPI field of the message is set to zero and the VCI contains the MSI of *B*. This will be used by the MAC at the CP to identify the MAC address of *B* in the mobile network. The message is a modified version of the standard ATM "Setup" message and includes the MVCI/IDVC allocated to *B*.

The "Proceed" and "Setup" messages are recognized by *A* and *B* using the MSI notation. When the "Setup" message arrives at *B*, it is passed sequentially from the MAC to the DLC, ATM, and ATM controller. If the ATM control unit of *B* accepts the call:

- The ATM control unit assigns a local VPI/VCI to the connection.

- The ID-assignment controller updates the *Local_Connection_Table* with the MVCI/IDVC allocated by the CP and the VPI/VCI allocated locally.

- The ATM control unit generates a "Connect" message for transmission to the CP (with VPI/VCI fields set to zero).

- The message flows through the ATM and DLC to the MAC. The MAC contends on the uplink control subchannel requesting channel allocation using *Remote_MSI*.

The MAC at the CP allocates the channel to *B* using the latter's MSI.

The MAC of *B* sends the "Connect" message on the allocated information slot(s).

When the message is successfully received by the CP:

- The MAC passes the message through the DLC and ATM layer to the ATM control unit.

- The ATM control unit passes an "Ack" message for transmission to *B* and a "Connect" message (with the assigned MVCI/IDVC) to *A*, setting VPI to zero and the VCIs to the MSIs of *B* and *A*, respectively.

When the "Connect" message is successfully received by *A*, it is passed sequentially from the MAC to the DLC, ATM, and ATM control unit. Then:

- The ATM control unit assigns a local VPI/VCI pair to the connection.

- The ID-assignment controller updates the *Local_Connection_Table* about the MVCI/IDVC allocated by the CP and the VPI/VCI allocated locally.

- The ATM control unit generates an "Ack" message for transmission to the CP (with VPI/VCI fields set to zero).

- The message flows through the ATM and DLC to the MAC. The MAC contends on the uplink control subchannel and requests channel allocation using its *Remote_MSI*.

The MAC at the CP allocates the channel to *A* using the latter's MSI.

The MAC of *A* sends the "Ack" message on the allocated information slot(s).

6. Call Release

Here we describe the process of a call release, given that the call is active between two remotes *A* and *B*, assuming *A* releases the call. Call-release processes in other cases can be obtained from this case since it is relatively more complicated. The case where a connection release involves an external source is considered separately later in the section.

The ATM control unit of *A* passes (through the ATM layer) a "Release" signaling message to the DLC for transmission. The VPI and VCI fields of the message are set to zero. The DLC, using *Remote_MSI*, passes the message to the MAC. The latter contends on the uplink control subchannel requesting a channel allocation.

The MAC at the CP allocates the channel to *A* using the latter's MSI.

The MAC of *A* sends the "Release" message on the allocated information slot(s). The message is a modified version of the standard ATM "Release" message and contains the MVCI and IDVC of the source seeking to be released.

When the "Release" message is successfully received by the CP:

- The message is passed on the path from the MAC to the DLC, ATM, and ATM control unit.

- The ATM control unit sends a "Release" message for transmission with VPI set to zero and VCI set to the MSI of *B*. This message contains the MVCI/IDVC of the source in *B* to be released, which were searched and found in the *Network_Connection_Table* (in the same entry the MVCI/IDVC of *A* have appeared).

When the "Release" message is successfully received by the *B*:

- The message is passed from the MAC to the DLC, ATM, and ATM controller.
- The ATM control unit sends a "Release Complete" message for transmission with the VPI and VCI set to zero. The message contains the MVCI/IDVC of the source in *B* to be released (for verification).
- It de-allocates the VPI/VCI assigned to the connection (searching over the *Local_Connection_Table*).
- The ID-assignment controller deletes in the *Local_Connection_Table* the entry in which the MVCI/IDVC pair appears.
- The "Release Complete" message flows through the ATM and DLC to the MAC. The MAC contends on the uplink control subchannel requesting channel allocation using its *Remote_MSI*.

The MAC at the CP allocates the channel to *B* using the latter's MSI.

The MAC of *B* sends the "Release Complete" message on the allocated information slot(s).

When the message is successfully received by the CP:

- The MAC passes the message through the DLC and ATM layer to the ATM control.
- The ATM control unit passes a "Release Complete" message for transmission to *A* (with the released MVCI/IDVC), setting VPI to zero and VCI to *A*'s MSI.
- The ID-assignment controller deletes from the *Network_Connection_Table* the parties *A* and *B* (in the entry in which the MVCI/IDVC pair of *B* appears). If no more parties are left in that entry, the corresponding MVCI is de-allocated.
- The "Release Complete" message flows toward the MAC that transmits it on the downlink information subchannel.

When the "Release Complete" message is successfully received by *A*:

- The message is passed on the path from the MAC to the DLC, ATM, and ATM control unit.
- The ATM control unit de-allocates the VPI/VCI assigned to the connection (searching over the *Local_Connection_Table* for the received MVCI/IDVC).

- The ID-assignment controller deletes in the *Local_Connection_Table* the entry in which the MVCI/IDVC pair appears.

7. Inter-Station Flow of Information

Here we describe the process of information flow when a remote is the originator of an ATM cell. The following describes the procedures in cases the CP, an external source and other remotes are the destinations of that cell. Processes in other cases can be obtained from this case. Nevertheless, the case of an external source-to-remote flow is considered separately later.

In a wireline ATM network, when a source sends a cell for transmission, it uses the ATM-oriented VPI/VCI to mark the (next-hop and thus end to end) destination. We use the same concept for remote sources. At the sender (marked as *A*), the ATM cells arrive at the DLC that translates the VPI/VCI addresses into MVCI and IDVC (using the *Local_Connection_Table*). The receiving DLC performs the opposite operation.

The ATM layer of *A* passes a cell to the DLC for radio transmission with appropriate VPI/VCI. The DLC converts the cell into a mobile ATM cell. The identifiers are translated into MVCI and IDVC using the *Local_Connection_Table* and the cell is passed to the MAC. The MAC of *A* contends on the uplink control subchannel requesting a channel allocation.

The MAC at the CP allocates the channel to *A* using the MVCI of the connection and the IDVC of *A* within this connection.

The MAC of *A* sends the cell on the allocated information slot(s).

When the cell is successfully received by the CP:

- The MAC passes it to the DLC.
- The DLC looks for the received MVCI in the *Network_Connection_Table*. If a source within the CP takes part in this connection, i.e., the port field value of some party is LOCAL, the PMAC converts the cell into an ATM cell and passes a copy to the ATM layer (after a translation of the MVCI/IDVC into VPI/VCI using *Local_Connection_Table*).
- If there are other remotes that participate in the connection, i.e., their port field value in the *Network_Connection_Table* is REMOTE, the DLC modifies the

cell header such that the bitwise-IDVC field contains "1" in the bits corresponding to all these remotes. The modified cell is then sent to the MAC for transmission on the downlink information subchannel.

- If there are other external parties within this connection, i.e., their port value in the *Network_Connection_Table* is BACKBONE, the DLC follows the procedure of local sources described above. That is, the cell is transferred to the ATM layer that routes it toward the backbone link.

A remote that successfully receives a cell, performs the following checks:

- If the cell has arrived on the downlink information subchannel (only) according to the CI field, and
- If the received MVCI is found in its *Local_Connection_Table*, and
- If the remote is not the source of the cell (received source-IDVC field is not equal the remote's IDVC in the *Local_Connection_Table*), and
- If the remote's corresponding bit in the received bitwise-IDVC field is "on"

then the cell is converted by the DLC into a standard ATM cell and passed to the ATM layer (after a translation of the MVCI/IDVC into VPI/VCI using the *Local_Connection_Table*).

8. Involvement of External Sources

The cases in which an external source (denoted as *E*) and a remote source (denoted as *R*) are involved in a connection need a somewhat special consideration. Here, we detail the changes in the databases and procedures at the various nodes required, when an external source is involved. We consider the cases of call establishment and call release in a remote-to-external source connection. The case of an information flow in an external source-to-remote connection is described as well (flow on the opposite direction has been discussed above).

a. Call Establishment from R to E

The transmission procedure of the "Setup" signaling message from the remote is the same as detailed earlier. If the ATM control unit decides to accept the call

(i.e., sufficient backbone link capacity to satisfy the QoS requirements of that call defined in *Class_Type_Table* is available), it assigns VPI/VCI to the connection on the backbone link destined to *E*. The ID-assignment controller at the CP registers the ATM address of *E* and the allocated VPI/VCI at the *Network_Connection_Table*, while the port field gets the value BACKBONE. A "Setup" message is sent then by the ATM control unit to *E*. The rest of the call-setup procedure described above follows the wireline ATM one (on the wireline segment) and the remote one described above (on the mobile segment).

b. Call Release from R to E

The transmission procedure of the "Release" signaling message from the remote is similar to the one detailed above. When the "Release" message is successfully received by the CP, it is passed to the ATM control unit. This searches for the MVCI/IDVC fields (that are part of the message) in the *Network_Connection_Table* and sends a "Release" message to the ATM address whose corresponding port field has the value BACKBONE (i.e., to *E*). Once a "Release Complete" signaling message has been received at the CP from *E*, the ID-assignment controller deletes parties *R* and *E* from the *Network_Connection_Table* (using the MVCI/IDVC which are part of the message). If no more parties are left in that entry in the table, the corresponding MVCI is de-allocated. The ATM control unit then sends a "Release Complete" message to the remote with the de-allocated MVCI/IDVC. The completion of the release process at *R* is similar to the one in a remote-to-remote connection case.

c. Information Flow from E to R

We describe here the flow of a cell from an external source to a remote. Source *E* passes a cell to the ATM layer using the VPI/VCI assigned at call setup. This cell flows in the wireline network using VPI/VCI notation until it reaches the ATM layer at the CP. The latter, using an appropriate connection matrix (not part of the mobile database), passes the cell to the DLC using VPI/VCI (assigned at call setup). The DLC uses the *Network_Connection_Table* to build the mobile cell with the MVCI/IDVC of *R*.

The cell is then passed to the MAC that transmits it on the downlink information subchannel. The reception procedure is the same as described earlier.

APPENDIX B. SELF-SIMILAR STOCHASTIC PROCESSES

A. DEFINITION

A random process $X = \{X_t: t = 0, 1, \ldots\}$ is *second-order self-similar* if the corresponding aggregated processes, $X^{(m)}$, become indistinguishable from X in their autocorrelation, $r(k)$, where [56]

$$X_k^{(m)} = \frac{1}{m} \times \left(X_{km-m+1} + \ldots + X_{km-1} + X_{km}\right), \quad k \geq 1, \quad m = 1, 2, \ldots.$$

B. PROPERTIES

The sum of the autocorrelation function of a self-similar process goes to infinity; this is called *long-range dependence* since each autocorrelation function decays hyperbolically. Poisson-based processes have an autocorrelation function that decays exponentially, thus $\sum_k r(k) < \infty$, and $r^{(m)}(k) \to 0$ as $m \to \infty$.

The *Hurst parameter*, H, measures self-similarity and burstiness of a source; the larger the value of H, the larger are these factors. Typical values of H for a self-similar process are in the range 0.7-0.9, versus 0.5 for a Poisson process. Table B.1 summarizes the differences between self-similar and Poisson processes [34].

Property	Self-Similar Process	Poisson Process
Autocorrelation Decay	Less than exponentially $r(k) \sim k^{-\eta}$, $0 < \eta < 1$, $k \to \infty$	Exponentially fast
Variance of Sample Mean	Less than 1/(sample size) $\mathrm{VAR}\{X^{(m)}\} \sim k^{-\eta}$, $0 < \eta < 1$, $m \to \infty$	1/(sample size)
Spectral Density at the Origin	Diverges (1/*f*-noise) $S(f) \sim f^{\theta}$, $0 < \theta < 1$, $f \to 0$	Finite

Table B.1: Comparison between Self-Similar Process and Poisson Processes [34]

The index of dispersion for count (IDC), defined as the ratio between the variance and the mean of the number of arrivals during a given time interval, is also a commonly used characteristic of a self-similar process [56]. For Poisson processes, the IDC is constant or converges to a fixed value whereas, for self-similar processes, it increases linearly on a log-log scale.

C. MODELS

Two formal mathematical models are presented for self-similar processes: fractional Gaussian and fractional autoregressive integrated moving average [56]. The former is a stationary Gaussian process with mean μ, variance σ^2, and autocorrelation function $r(k) = \frac{1}{2}\left(|k+1|^{2H} - |k|^{2H} + |k-1|^{2H}\right)$, $k > 0$. The latter is a generalization of a class of time series models called Box-Jenkins [15]; the process has three independent variables and it presents more flexibility than the fractional Gaussian model. A self-similar process is generated using a sequence of independent and identically-distributed integer random variables U_0, U_1, U_2,... (inter-renewal times) with a *heavy tail*, i.e., with the property

$$\lim_{u\to\infty} P\{U \geq u\} \sim u^{-\alpha} h(u),$$

where h is slowly varying at infinity and $0 < \alpha < 2$.

A slightly different definition is proposed by [28]. A fractional Brownian motion with Hurst parameter in the range [½, 1) can serve in generating a self-similar traffic stream. A fractional Brownian motion is a zero-mean Guassian random process, $z(t)$, with stationary increments and covariance structure:

$$\mathrm{COV}\{z(t), z(s)\} = \left(t^{2H} + s^{2H} - |t-s|^{2H}\right) \times \frac{\sigma^2}{2}.$$

In the general case $H = ½$, and $z(t)$ is a standard Brownian motion.

One of the simplest heavy-tail distributions (also called *power law*) is the *Pareto* distribution, which describes the statistics of event interarrivals. It has several forms, one of which [34] is

$$P\{X \geq x\} = 1 - F_X(x) = \frac{\beta^\alpha}{(x+\beta)^\alpha},$$

where α and β are the parameters of the Pareto distribution; α, $\beta > 0$. The probability density function, $f_X(x)$, associated with $F_X(x)$ is

$$f_X(x) = \frac{\alpha\beta^\alpha}{(x+\beta)^{\alpha+1}}.$$

Finite moments ($k = 1, 2, \ldots$) exist for $k < \alpha$ only. Table B.2 presents the characteristics of Pareto interarrival process for different ranges of α.

Region	Interarrival Mean	Interarrival Variance	Process
$2 < \alpha$	Finite	Finite	Non-self-similar
$1 < \alpha < 2$	Finite	Infinite	Self-similar
$0 < \alpha < 1$	Infinite	Infinite	Self-similar

Table B.2: Pareto Interarrival Process as a Function of α

A different definition of the Pareto process [94] is

$$F_X(x) = P\{X \leq x\} = 1 - \left(\frac{x}{\varepsilon}\right)^{-D}, \qquad x \geq \varepsilon, \qquad \varepsilon, D > 0,$$

where ε is called the location parameter and D the shape parameter. The n^{th} moment of X exists only if $n < D$ and is given by

$$E\{X^n\} = \frac{D\varepsilon^n}{D-n}.$$

APPENDIX C. MATLAB PROGRAMS

A. STATIC ALLOCATION OF SPEECH SOURCES

```
%--------------------------------------------------------------------------
% Program:  Find the cell loss probability in the burst region for
%           Ns speech conversations.
%
% Speech Model of One Conversation:
%
%                  +-----+   w   +-----+   y   +-----+
%                  |     |------>|     |------>|     |
%                  |  0  |       |  1  |       |  2  |
%                  |     |<------|     |<------|     |
%                  +-----+   x   +--+--+   z   +--+--+
%                                     |             |
%                                     V             V
%                                 Rs(cells/sec)  2Rs(cells/sec)
%
%--------------------------------------------------------------------------
% simulation results
sim_result = [.16  .070  .043   .038   .032   .027   .023   .017   .013
              .0091  .0054  .0015 7.5e-5 0;
              .12  .051  .026   .019   .012   .0061  .0015  2.6e-4 1.4e-4
              5.1e-5 1.6e-5 0     0      0;
              .10  .044  .021   .011   .0036  8.4e-4 3.0e-4 8.5e-5 1.8e-5
              0      0      0     0      0;
              .080 .030  .011   .0037  8.2e-4 2.6e-4 1.7e-5 0      0
              0      0      0     0      0;
              .069 .024  .0072  .0014  1.7e-4 1.7e-5 0      0      0
              0      0      0     0      0;
              .061 .020  .0053  .0010  2.3e-4 5.1e-5 0      0      0
              0      0      0     0      0;
              .058 .017  .0033  4.0e-4 2.7e-5 0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
              0      0      0     0      0;
              .039 .0097 .0014  1.0e-4 1.1e-6 0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
              0      0      0     0      0;
              0    0     0      0      0      0      0      0      0
```

```
            0      0      0      0      0;
            0    0      0      0      0      0      0      0      0
            0      0      0      0      0;
            0    0      0      0      0      0      0      0      0
            0      0      0      0      0;
            .024 .0021 1.6e-5 0      0      0      0      0      0
            0      0      0      0      0];
w = 3.09;
x = 0.65;
y = 0.25;
z = 4.72;
Rs = 85.1;
CTDs = 40E-03;      % Cell Transfer Delay
CLPs = 1E-03;       % Cell Loss Probability
lamda_1 = [0 Rs 2*Rs];
pi_1 = [0.1665 0.7916 0.0419];
mean_rate_1 = sum(pi_1.*lamda_1);
Ns = 20;            % maximum number of conversations to be checked
Nc = 125;           % number of capacities to be checked
MAX_K = 300;        % maximum allowed buffer size
NUM_OF_STATES = 3;  % number of stases in Markov chain
r = 10;             % number of stages of Erlang service process
max_buffer_size = zeros(Ns,Nc);
Cs = zeros(Ns,Nc);
Cs_simulation = zeros(Ns,14);
dominant = zeros(Ns,Nc);
K0_MM1K = zeros(Ns,Nc);
K0_MD1K = zeros(Ns,Nc);
buffer = 1:MAX_K;
DD1K_loss = zeros(Ns,Nc);  % histogram CLP (D/D/1/K queue)
DD1K_fluid_loss = zeros(Nc,MAX_K);
DD1K_fluid_loss_at_max_buffer = zeros(Ns,Nc);
MM1K_fluid_loss_at_max_buffer = zeros(Ns,Nc);
MD1K_fluid_loss_at_max_buffer = zeros(Ns,Nc);
pi = pi_1;
for i = 1:Ns,
    % find the aggregate arrival rate
    for j = 1:(NUM_OF_STATES-1)*i+1,
        lamda(j) = (j-1)*Rs;      %in cells/sec

    end
    % find the mean arrival rate
    mean_rate = i * mean_rate_1;
    % find the link capacities, in relation to the mean arrival rate
    % They are 1.01, 1.02, 1.03 ...., and 2.25 of the mean_rate
    for j = 1:Nc,
        Cs(i,j) = (1 + j/100)*mean_rate;
        max_buffer_size(i,j) = floor(Cs(i,j)*CTDs)-1;
    end
    % find the link capacities, which have used in simulation
    Cs_simulation(i,:) = [1.01 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2 2.1 2.2
                          2.25]*mean_rate;
    % find the CLP using D/D/1/K system (histogram approximation)
    for j = 1:Nc,
        for k = 1:(NUM_OF_STATES-1)*i+1,
            if lamda(k) > Cs(i,j),
                DD1K_loss(i,j) = DD1K_loss(i,j) +
                                 (1/mean_rate)*pi(k)*(lamda(k)-Cs(i,j));
            end
        end
```

```
    end
    % find the dominant eigenvalue of the fluid model
    for j = 1:Nc,
        D = zeros(NUM_OF_STATES,NUM_OF_STATES);
        for k = 1:NUM_OF_STATES,
            D(k,k) = lamda_1(k)-Cs(i,j)/i;
        end
        M = [-w w 0; x -(x+y) y; 0 z -z];
        M_prime = M*inv(D);
        eigval = eig(M_prime);
        dominant(i,j) = -10000;
        for k = 1:NUM_OF_STATES,
            if eigval(k) < -1E-06,
                if eigval(k) > dominant(i,j),
                    dominant(i,j) = eigval(k);
                end
            end
        end
    end
    % find the CLP for D/D/1/K system
    for j = 1:Nc,
        DD1K_fluid_loss_at_max_buffer(i,j) = DD1K_loss(i,j)*exp(dominant(i,j)*
                                             max_buffer_size(i,j));
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulations of M/M/1/K and M/D/1/K systems                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if 0,
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Find the beginning of the burst region in a M/D/1/K system              %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for j = 1:Nc,
        rho = zeros(1,i*(NUM_OF_STATES-1)+1);
        for k = 1:i*(NUM_OF_STATES-1)+1,
            rho(k) = lamda(k)/Cs(i,j);
        end
        MM1K_loss = zeros(1,MAX_K);
        for K = 1:MAX_K,
            for k = 1:i*(NUM_OF_STATES-1)+1,
                MM1K_loss(K) = MM1K_loss(K) +
                               (1-rho(k))*rho(k)^K/(1-rho(k)^(K+1))*
                               pi(k)*lamda(k)/mean_rate;
            end
        end
        lg_MM1K_loss = log(MM1K_loss);
        d_dK_lg_MM1K_loss = diff(lg_MM1K_loss);
        for k = 1:MAX_K,
            if d_dK_lg_MM1K_loss(k) > dominant(i,j),
                K0_MM1K(i,j) = k;
                break;
            end
        end
        if K0_MM1K(i,j) < max_buffer_size(i,j),
            MM1K_fluid_loss_at_max_buffer(i,j) =
                                MM1K_loss(K0_MM1K(i,j))*exp(dominant(i,j)*
                                (max_buffer_size(i,j)-K0_MM1K(i,j)));
        else
            MM1K_fluid_loss_at_max_buffer(i,j) =
                                MM1K_loss(max_buffer_size(i,j));
        end
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Find the beginning of the burst region in a M/D/1/K system       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j = 1:Nc,
    MD1K_loss = zeros(1,MAX_K);
    % handle first the case of K0=1 which requires a special set
    % of equations. then conitnue with values of K>1 systematically.
    K = 1;
    for k = 2:(NUM_OF_STATES-1)*i+1,
        P = zeros(1,K*r+1);
        P(1) = 1;
        for m = 1:r-1,
            P(m+1) = lamda(k)/(r*Cs(i,j))*P(m);
        end
        normalize_factor = sum(P);
        for m = 1:K*r+1,
            P(m) = P(m)/normalize_factor;
        end
        MD1K_loss(K) = MD1K_loss(K) + (lamda(k)-(1-P(1))*Cs(i,j))*
                       pi(k)/mean_rate;
    end
    for K = 2:MAX_K,
        for k = 2:(NUM_OF_STATES-1)*i+1,
            P = zeros(1,MAX_K*r+1);
            P(1) = 1;
            P(2) = lamda(k)/(r*Cs(i,j));
            for m = 1:r-2,
                P(m+2) = (lamda(k)/(r*Cs(i,j))+1)*P(m+1);
            end
            for m = r-1:K*r-r-1,
                P(m+2) = (lamda(k)/(r*Cs(i,j))+1)*
                         P(m+1)-lamda(k)/(r*Cs(i,j))*P(m+2-r);
            end
            for m = K*r-r:K*r-2,
                P(m+2) = P(m+1) - lamda(k)/(r*Cs(i,j))*P(m+2-r);
            end
            P(K*r+1) = lamda(k)/(r*Cs(i,j))*P(K*r-r);
            normalize_factor = sum(P);
            for m = 1:K*r+1,
                P(m) = P(m)/normalize_factor;
            end
            MD1K_loss(K) = MD1K_loss(K) + (lamda(k)-(1-P(1))*Cs(i,j))*
                           pi(k)/mean_rate;
        end
    end
    % find the beginning of the burst region (K0) for MD1K arrival
    lg_MD1K_loss = log(MD1K_loss);
    d_dK_lg_MD1K_loss = diff(lg_MD1K_loss);
    for k = 1:MAX_K,
        if d_dK_lg_MD1K_loss(k) > dominant(i,j),
            K0_MD1K(i,j) = k;
            break;
        end
    end
    if K0_MD1K(i,j) < max_buffer_size(i,j),
        MD1K_fluid_loss_at_max_buffer(i,j) = MD1K_loss(K0_MD1K(i,j))*
                                exp(dominant(i,j)*(max_buffer_size(i,j)-
                                K0_MD1K(i,j)));
    else
```

```
            MD1K_fluid_loss_at_max_buffer(i,j) =
                                MD1K_loss(max_buffer_size(i,j));
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of M/M/1/K and M/D/1/K systems simulation                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % find the aggregate state probabilities
    pi = conv(pi, pi_1);
end

% Plot results of theory and simulation
figure;
a = [120 125]; b1 = [0.5 0.5]; b2 = [0.3 0.3];
semilogy(Cs(1,:),DD1K_fluid_loss_at_max_buffer(1,:),'r',
         Cs_simulation(1,:),sim_result(1,:),'g',
         Cs(1,:),DD1K_fluid_loss_at_max_buffer(5,:),'r',
         Cs_simulation(1,:),sim_result(5,:),'g',
         Cs(1,:),DD1K_fluid_loss_at_max_buffer(10,:),'r',
         Cs_simulation(1,:),sim_result(10,:),'g',
         Cs(1,:),DD1K_fluid_loss_at_max_buffer(20,:),'r',
         Cs_simulation(1,:),sim_result(20,:),'g',
         a,b1,'r',a,b2,'g', [70 170], [CLPs CLPs], 'b-.');
axis([70 170 1E-04 1]);
xlabel('Link Capacity per Source (cells/sec)'); ylabel('Loss Probability');
text(127, 0.5, 'Analysis'); text(127, 0.3, 'Simulation');
whitebg;
```

B. STATIC ALLOCATION OF VIDEO SOURCES

```
%-----------------------------------------------------------------------
% Program:  Find the cell loss probability in the burst region for
%           Nv video sources.
%
% Video Model: 8-state Markov chain, considered as a 8-bin histogram
%
%             +--------------------------+ +-----------------------+
%             |                          | |                       |
%             |                          V |                       V
%          +-----+       +-----+       +-----+                   +-----+
%          |     |------>|     |------>|     |-->            -->|     |
%          |  1  |       |  2  |       |  3  |      . . .       |  8  |
%          |     |<------|     |<------|     |<--            <--|     |
%          +-----+       +--+--+       +--+--+                   +--+--+
%             |             |             |                         |
%             V             V             V                         V
%           lamda1        lamda2        lamda3                    lamda8 (cells/sec)
%
%    (only selected transitions are depicted)
%
% The lamda's are chosen, such that the video source generates on average
% 64kbps. Multiple state probabilities by the corresponding lamda, and get
% the mean arrival rate, which is 177.78 cells/sec for one source.
%-----------------------------------------------------------------------
sim_result=[.071 .059  .039   .027   .017   .011   .0069  .0046  .0024  0;
            .052 .034  .019   .0097  .0050  .0021  8.0e-4 2.5e-4 7.8e-5 3.8e-6;
            .042 .025  .012   .0048  .0014  4.2e-4 1.0e-4 1.4e-5 7.1e-6 0;
            .034 .018  .0062  .0017  5.5e-4 1.1e-4 2.1e-5 2.0e-6 0      0;
            .031 .016  .0053  .0019  2.1e-4 2.9e-5 2.9e-6 0      0      0;
            .029 .012  .0032  7.5e-4 1.3e-4 1.6e-5 2.7e-7 0      0      0;
            .025 .011  .0025  3.6e-4 3.8e-5 1.9e-6 0      0      0      0;
            .023 .0087 .0018  1.9e-4 1.3e-5 1.8e-7 0      0      0      0;
            .021 .0077 .0014  1.4e-4 4.8e-6 0      0      0      0      0;
            .019 .0058 8.7e-4 7.8e-5 2.2e-6 0      0      0      0      0;
            0    0     0      0      0      0      0      0      0      0;
            0    0     0      0      0      0      0      0      0      0;
            0    0     0      0      0      0      0      0      0      0;
            0    0     0      0      0      0      0      0      0      0;
            0    0     0      0      0      0      0      0      0      0;
            0    0     0      0      0      0      0      0      0      0;
            0    0     0      0      0      0      0      0      0      0;
            0    0     0      0      0      0      0      0      0      0;
            .013 .0018 5.0e-5 0      0      0      0      0      0      0];

M = [-4.737  4.105  0.000  0.316  0.000  0.316  0.000  0.000;
      0.643 -1.714  0.771  0.043  0.129  0.043  0.086  0.000;
      0.000  1.408 -2.254  0.845  0.000  0.000  0.000  0.000;
      0.000  0.082  0.740 -1.562  0.616  0.082  0.000  0.041;
      0.000  0.000  0.111  0.741 -1.333  0.407  0.074  0.000;
      0.000  0.000  0.000  0.095  1.518 -3.130  1.423  0.095;
      0.000  0.000  0.000  0.000  0.137  2.606 -2.743  0.000;
      0.000  0.000  0.000  0.000  0.189  0.000  0.189 -0.378];
lamda_1 = [140000 170000 200000 230000 260000 290000 320000 350000]/1320.75;
CTDv = 100E-03; % Cell Transfer Delay
```

```
CLPv = 5E-05;    % Cell Loss Probability
pi_1 = [0.025 0.19 0.145 0.21 0.24 0.09 0.06 0.04];
mean_rate_1 = sum(pi_1.*lamda_1);
Nv = 20;         % maximum number of video sources to be checked
Nc = 45;         % number of capacities to be checked
MAX_K = 5520;    % maximum allowed buffer size=1.45*10calls*
                 % 178cells/sec*100msec)
HISTOGRAM_LEVELS = 8;
r = 10;          % number of stages of Erlang service process
max_buffer_size = zeros(Nv,Nc);
Cv = zeros(Nv,Nc);
Cv_simulation = zeros(Nv,10);
dominant = zeros(Nv,Nc);
K0_MM1K = zeros(Nv,Nc);
K0_MD1K = zeros(Nv,Nc);
buffer = 1:MAX_K;
DD1K_loss = zeros(Nv,Nc);     % loss for D/D/1/K system
DD1K_fluid_loss = zeros(Nc,MAX_K);
DD1K_fluid_loss_at_max_buffer = zeros(Nv,Nc);
MM1K_fluid_loss_at_max_buffer = zeros(Nv,Nc);
MD1K_fluid_loss_at_max_buffer = zeros(Nv,Nc);
pi = pi_1;
for i = 1:Nv,
    % find the aggregate arrival rate
    for j = 1:(HISTOGRAM_LEVELS-1)*i+1,
        lamda(j) = (i*140000+(j-1)*30000)/1320.75;      %in cells/sec
    end
    % find the mean arrival rate
    mean_rate = i * mean_rate_1;
    % find the link capacities, in relation to the mean arrival rate
    % They are 1.01, 1.02, 1.03,...., and 1.45 times the mean_rate
    for j = 1:Nc,
        Cv(i,j) = (1 + (j-1)/100)*mean_rate;
        max_buffer_size(i,j) = floor(Cv(i,j)*CTDv)-1;
    end
    % find the link capacities, which have used in simulation
    Cv_simulation(i,:) =
                [1.01 1.05 1.1 1.15 1.2 1.25 1.3 1.35 1.4 1.45]*mean_rate;
    % find the CLP using D/D/1/K system (histogram approximation)
    for j = 1:Nc,
        for k = 1:(HISTOGRAM_LEVELS-1)*i+1,
            if lamda(k) > Cv(i,j),
                DD1K_loss(i,j) = DD1K_loss(i,j) +
                                 (1/mean_rate)*pi(k)*(lamda(k)-Cv(i,j));
            end
        end
    end
    % find the dominant eigenvalue of the fluid model
    for j = 1:Nc,
        D = zeros(HISTOGRAM_LEVELS,HISTOGRAM_LEVELS);
        for k = 1:HISTOGRAM_LEVELS,
            D(k,k) = lamda_1(k) - Cv(i,j)/i;
        end
        M_prime = M*inv(D);
        eigval = eig(M_prime);
        dominant(i,j) = -10000;
        for k = 1:HISTOGRAM_LEVELS,
            if eigval(k) < -1E-06,
                if eigval(k) > dominant(i,j),
                    dominant(i,j) = eigval(k);
```

```
                end
            end
        end
    end
    % find the CLP for D/D/1/K system
    for j = 1:Nc,
        DD1K_fluid_loss_at_max_buffer(i,j) = DD1K_loss(i,j)*exp(dominant(i,j)*
                                             max_buffer_size(i,j));
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulations of M/M/1/K and M/D/1/K systems                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if 0,
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Find the beginning of the burst region in a M/M/1/K system       %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for j = 1:Nc,
        rho = zeros(1,i*(HISTOGRAM_LEVELS-1)+1);
        for k = 1:i*(HISTOGRAM_LEVELS-1)+1,
            rho(k) = lamda(k)/Cv(i,j);
        end
        MM1K_loss = zeros(1,MAX_K);
        for K = 1:MAX_K,
            for k = 1:i*(HISTOGRAM_LEVELS-1)+1,
                MM1K_loss(K) = MM1K_loss(K) + (1-rho(k))*
                               rho(k)^K/(1-rho(k)^(K+1))*
                               pi(k)*lamda(k)/mean_rate;
            end
        end
        lg_MM1K_loss = log(MM1K_loss);
        d_dK_lg_MM1K_loss = diff(lg_MM1K_loss);
        for k = 1:MAX_K,
            if d_dK_lg_MM1K_loss(k) > dominant(i,j),
                K0_MM1K(i,j) = k;
                break;
            end
        end
        if K0_MM1K(i,j) < max_buffer_size(i,j),
            MM1K_fluid_loss_at_max_buffer(i,j) = MM1K_loss(K0_MM1K(i,j))*
                                                 exp(dominant(i,j)*
                                                 (max_buffer_size(i,j)-
                                                 K0_MM1K(i,j)));
        else
            MM1K_fluid_loss_at_max_buffer(i,j) =
                                    MM1K_loss(max_buffer_size(i,j));
        end
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Find the beginning of the burst region in a M/D/1/K system       %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for j = 1:Nc,
        MD1K_loss = zeros(1,MAX_K);
        % handle first the case of K0=1 which requires a special set
        % of equations. then conitnue with values of K>1 systematically.
        K = 1;
        for k = 2:(HISTOGRAM_LEVELS-1)*i+1,
            P = zeros(1,K*r+1);
            P(1) = 1;
            for m = 1:r-1,
                P(m+1) = lamda(k)/(r*Cv(i,j))*P(m);
```

```
            end
            normalize_factor = sum(P);
            for m = 1:K*r+1,
                P(m) = P(m)/normalize_factor;
            end
            MD1K_loss(K) = MD1K_loss(K) + (lamda(k)-(1-P(1))*Cv(i,j))*
                           pi(k)/mean_rate;
        end
        for K = 2:MAX_K,
            for k = 2:(HISTOGRAM_LEVELS-1)*i+1,
                P = zeros(1,MAX_K*r+1);
                P(1) = 1;
                P(2) = lamda(k)/(r*Cv(i,j));
                for m = 1:r-2,
                    P(m+2) = (lamda(k)/(r*Cv(i,j))+1)*P(m+1);
                end
                for m = r-1:K*r-r-1,
                    P(m+2) = (lamda(k)/(r*Cv(i,j))+1)*P(m+1)-
                             lamda(k)/(r*Cv(i,j))*P(m+2-r);
                end
                for m = K*r-r:K*r-2,
                    P(m+2) = P(m+1) - lamda(k)/(r*Cv(i,j))*P(m+2-r);
                end
                P(K*r+1) = lamda(k)/(r*Cv(i,j))*P(K*r-r);
                normalize_factor = sum(P);
                for m = 1:K*r+1,
                    P(m) = P(m)/normalize_factor;
                end
                MD1K_loss(K) = MD1K_loss(K) + (lamda(k)-(1-P(1))*Cv(i,j))*
                               pi(k)/mean_rate;
            end
        end
        % find the beginning of the burst region (K0) for MD1K arrival
        lg_MD1K_loss = log(MD1K_loss);
        d_dK_lg_MD1K_loss = diff(lg_MD1K_loss);
        for k = 1:MAX_K,
            if d_dK_lg_MD1K_loss(k) > dominant(i,j),
                K0_MD1K(i,j) = k;
                break;
            end
        end
        if K0_MD1K(i,j) < max_buffer_size(i,j),
            MD1K_fluid_loss_at_max_buffer(i,j) = MD1K_loss(K0_MD1K(i,j))*
                                                 exp(dominant(i,j)*
                                                 (max_buffer_size(i,j)-
                                                 K0_MD1K(i,j)));
        else
            MD1K_fluid_loss_at_max_buffer(i,j) =
                                    MD1K_loss(max_buffer_size(i,j));
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of M/M/1/K and M/D/1/K systems simulation                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % find the aggregate state probabilities
    pi = conv(pi, pi_1);
end

% Plot results of theory and simulation
```

```
figure;
a = [220 225]; b1 = [0.5 0.5]; b2 = [0.3 0.3];
semilogy(Cv(1,:),DD1K_fluid_loss_at_max_buffer(1,:),'r',
         Cv_simulation(1,:),sim_result(1,:),'g',
         Cv(1,:),DD1K_fluid_loss_at_max_buffer(5,:),'r',
         Cv_simulation(1,:),sim_result(5,:),'g',
         Cv(1,:),DD1K_fluid_loss_at_max_buffer(10,:),'r',
         Cv_simulation(1,:),sim_result(10,:),'g',
         Cv(1,:),DD1K_fluid_loss_at_max_buffer(20,:),'r',
         Cv_simulation(1,:),sim_result(20,:),'g',
         a,b1,'r',a,b2,'g', [180 260], [CLPv CLPv], 'b-.');
axis([180 260 1E-05 1]);
xlabel('Link Capacity per Source (cells/sec)'); ylabel('Loss Probability');
text(227, 0.5, 'Analysis'); text(227, 0.3, 'Simulation');
whitebg;
```

C. STATIC ALLOCATION OF DATA SOURCES

```
%------------------------------------------------------------------------
% Program:  Find the cell loss probability in the burst region for
%           Nd data sources.
%
% Model of One Data Source:
%                                    0.001
%                      +----------------------------+
%                      |                            |
%                   +--+--+  0.06 +--+--+        +--V--+
%                   |     |------>| low |        |high |
%                   | OFF |       |     |        |     |
%                   |     |<------| act.|        | act.|
%                   +--+--+   2   +--+--+        +--+--+
%                      ^             |              ||
%                      +-------------+--------------+|
%                              0.1    |              |
%                                     |              |
%                                     V              V
%                                 lamda1(cps)    lamda2(cps)
%
%
% Note: There are three cases in which a warning messgae is sent by Matlab:
%       "Warning: Log of zero"
%       There is nothing wrong with these messages, since on a  32-bit
%       machines, for some large values of rho (arrival rate in a bin
%       divided by the capacity), the value of rho^K (for some K) exceeds
%       the maximum allowed that can be represented by a variable of
%       size double.
%------------------------------------------------------------------------
% results of simulation
sim_result = [.92 .77  .56    .39    .25    .15    .085   .067   .040   .018
              8.7e-4;
              .83 .60  .30    .13    .046   .0037  2.9e-4 3.7e-4 8.8e-5 8.7e-6
              0;
              .76 .45  .14    .036   .0032  9.7e-4 2.2e-4 0      0      0
              0;
              .70 .35  .077   .0042  5.3e-4 3.5e-5 2.7e-5 0      0      0
              0;
              .65 .28  .034   .0027  5.5e-5 6.8e-7 0      0      0      0
              0;
              .59 .20  .014   .0011  9.5e-8 0      0      0      0      0
              0;
              .55 .16  .0072  3.5e-4 1.6e-5 0      0      0      0      0
              0;
              .51 .13  .0046  1.5e-5 0      0      0      0      0      0
              0;
              .49 .010 .0033  7.7e-5 0      0      0      0      0      0
              0;
              .46 .080 .0012  1.2e-6 0      0      0      0      0      0
              0
              0   0    0      0      0      0      0      0      0      0
              0
              0   0    0      0      0      0      0      0      0      0
              0
              0   0    0      0      0      0      0      0      0      0
```

```
                0
                0   0    0      0      0      0      0      0      0      0
                0
                0   0    0      0      0      0      0      0      0      0
                0
                0   0    0      0      0      0      0      0      0      0
                0
                0   0    0      0      0      0      0      0      0      0
                0
                0   0    0      0      0      0      0      0      0      0
                0
                0   0    0      0      0      0      0      0      0      0
                0
                .29 .013 2.5e-5 0      0      0      0      0      0      0
                0];
lamda_1 = [0 100 100];
M = [-0.061 0.06 0.001; 2 -2 0; 0.1 0 -0.1];
pi_1 = [0 0 1] * inv([-0.061 0.06 1; 2 -2 1; 0.1 0 1]);
mean_rate_1 = sum(pi_1.*lamda_1);
Nd = 20;              % maximum number of conversations to be checked
Nc = 96;              % number of capacities to be checked
MAX_K = 1000;         % maximum allowed buffer size
NUM_OF_STATES = 3;    % number of stases in Markov chain
Rd = 100;             % basic arrival rate (in low activity state)
CTDd = 1000E-03;      % Cell Transfer Delay
CLPd = 1E-06;         % Cell Loss Probability
r = 10;               % number of stages of Erlang service process
max_buffer_size = zeros(Nd,Nc);
Cd = zeros(Nd,Nc);
Cd_simulation = zeros(Nd,11);
dominant = zeros(Nd,Nc);
K0_MM1K = zeros(Nd,Nc);
K0_MD1K = zeros(Nd,Nc);
buffer = 1:MAX_K;
DD1K_loss = zeros(Nd,Nc);  % histogram CLP (D/D/1/K queue)
DD1K_fluid_loss = zeros(Nc,MAX_K);
DD1K_fluid_loss_at_max_buffer = zeros(Nd,Nc);
MM1K_fluid_loss_at_max_buffer = zeros(Nd,Nc);
MD1K_fluid_loss_at_max_buffer = zeros(Nd,Nc);
pi = [pi_1(1) pi_1(2)+pi_1(3)];
for i = 1:Nd,
    % Find the aggregate arrival rate and state probabilities
    % Due to the spcial constellation, the aggregate source will
    %   only i+1 distinct arrival rates (and hence probabilities)
    for j = 1:i+1,
        lamda(j) = (j-1)*Rd;      %in cells/sec
    end
    % find the mean arrival rate
    mean_rate = i * mean_rate_1;
    % find the link capacities, in relation to the mean arrival rate
    % They are 1.25, 1.5, 1.75,.... and 25 times the mean_rate
    for j = 1:Nc,
        Cd(i,j) = (1 + j/4) * mean_rate;
        max_buffer_size(i,j) = floor(Cd(i,j)*CTDd)-1;
    end
    % find the link capacities, which have used in simulation
    Cd_simulation(i,:) = [1.01 2.5 5 7.5 10 12.5 15 17.5 20 22.5 25]*mean_rate;
    % find the CLP using D/D/1/K system (histogram approximation)
    for j = 1:Nc,
        for k = 1:i+1,
```

```
            if lamda(k) > Cd(i,j),
                DD1K_loss(i,j) = DD1K_loss(i,j) + (1/mean_rate)*pi(k)*
                                 (lamda(k)-Cd(i,j));
            end
        end
    end
    % find the dominant eigenvalue of the fluid model
    for j = 1:Nc,
        D = zeros(NUM_OF_STATES,NUM_OF_STATES);
        for k = 1:NUM_OF_STATES,
            D(k,k) = lamda_1(k)-Cd(i,j)/i;
        end
        M_prime = M*inv(D);
        eigval = eig(M_prime);
        dominant(i,j) = -10000;
        for k = 1:NUM_OF_STATES,
            if eigval(k) < -1E-06,
                if eigval(k) > dominant(i,j),
                    dominant(i,j) = eigval(k);
                end
            end
        end
    end
    % find the CLP for D/D/1/K system
    for j = 1:Nc,
        DD1K_fluid_loss_at_max_buffer(i,j) = DD1K_loss(i,j)*exp(dominant(i,j)*
                                             max_buffer_size(i,j));
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulations of M/M/1/K and M/D/1/K systems                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if 0,
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Find the beginning of the burst region in a M/M/1/K system              %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for j = 1:Nc,
    for j = 1:Nc,
        rho = zeros(1,i+1);
            for k = 1:i+1,
                rho(k) = lamda(k)/Cd(i,j);
        end
        MM1K_loss = zeros(1,MAX_K);
        for K = 1:MAX_K,
            for k = 1:i+1,
                MM1K_loss(K) = MM1K_loss(K) + (1-rho(k))*rho(k)^K/
                               (1-rho(k)^(K+1))*pi(k)*lamda(k)/mean_rate;
            end
        end
        lg_MM1K_loss = log(MM1K_loss);
        d_dK_lg_MM1K_loss = diff(lg_MM1K_loss);
        for k = 1:MAX_K,
            if d_dK_lg_MM1K_loss(k) > dominant(i,j),
                K0_MM1K(i,j) = k;
                break;
            end
        end
        if K0_MM1K(i,j) < max_buffer_size(i,j),
            MM1K_fluid_loss_at_max_buffer(i,j) = MM1K_loss(K0_MM1K(i,j))*
                                                 exp(dominant(i,j)*
                                                 (max_buffer_size(i,j)-
```

```
                                                    K0_MM1K(i,j)));
    else
        MM1K_fluid_loss_at_max_buffer(i,j) =
                                MM1K_loss(max_buffer_size(i,j));
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Find the beginning of the burst region in a M/D/1/K system       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j = 1:Nc,
for j = 1:Nc,
    MD1K_loss = zeros(1,MAX_K);
    % handle first the case of K0=1 which requires a special set
    % of equations. then conitnue with values of K>1 systematically.
    K = 1;
    for k = 2:i+1,
        P = zeros(1,K*r+1);
        P(1) = 1;
        for m = 1:r-1,
            P(m+1) = lamda(k)/(r*Cd(i,j))*P(m);
        end
        normalize_factor = sum(P);
        for m = 1:K*r+1,
            P(m) = P(m)/normalize_factor;
        end
        MD1K_loss(K) = MD1K_loss(K) + (lamda(k)-(1-P(1))*Cd(i,j))*
                       pi(k)/mean_rate;
    end
    for K = 2:MAX_K,
        for k = 2:i+1,
            P = zeros(1,MAX_K*r+1);
            P(1) = 1;
            P(2) = lamda(k)/(r*Cd(i,j));
            for m = 1:r-2,
                P(m+2) = (lamda(k)/(r*Cd(i,j))+1)*P(m+1);
            end
            for m = r-1:K*r-r-1,
                P(m+2) = (lamda(k)/(r*Cd(i,j))+1)*P(m+1)-lamda(k)/
                         (r*Cd(i,j))*P(m+2-r);
            end
            for m = K*r-r:K*r-2,
                P(m+2) = P(m+1) - lamda(k)/(r*Cd(i,j))*P(m+2-r);
            end
            P(K*r+1) = lamda(k)/(r*Cd(i,j))*P(K*r-r);
            normalize_factor = sum(P);
            for m = 1:K*r+1,
                P(m) = P(m)/normalize_factor;
            end
            MD1K_loss(K) = MD1K_loss(K) + (lamda(k)-(1-P(1))*Cd(i,j))*
                           pi(k)/mean_rate;
        end
    end
    lg_MD1K_loss = log(MD1K_loss);
    d_dK_lg_MD1K_loss = diff(lg_MD1K_loss);
    for k = 1:MAX_K,
        if d_dK_lg_MD1K_loss(k) > dominant(i,j),
            K0_MD1K(i,j) = k;
            break;
        end
    end
```

```
        if K0_MD1K(i,j) < max_buffer_size(i,j),
            MD1K_fluid_loss_at_max_buffer(i,j) = MD1K_loss(K0_MD1K(i,j))*
                                                 exp(dominant(i,j)*
                                                (max_buffer_size(i,j)-
                                                 K0_MD1K(i,j)));
        else
            MD1K_fluid_loss_at_max_buffer(i,j) =
                                    MD1K_loss(max_buffer_size(i,j));
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of M/M/1/K and M/D/1/K systems simulation                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % find the aggregate state probabilities for the next case of Nd
    pi = zeros(1,i+2);
    for j = 1:i+2,
        for k = 0:j-1,
            pi(j) = pi(j) + n_over_k(j-1,k) * pi_1(2)^k * pi_1(3)^(j-1-k);
        end
        pi(j) = pi(j) * n_over_k(i+1,i+1-(j-1)) * pi_1(1)^(i+1-(j-1));
    end
end

% Plot results of theory and simulation
figure;
a = [72 77]; b1 = [0.6 0.6]; b2 = [0.3 0.3];
semilogy(Cd(1,:),DD1K_fluid_loss_at_max_buffer(1,:),'r',
         Cd_simulation(1,:),sim_result(1,:),'g',
         Cd(1,:),DD1K_fluid_loss_at_max_buffer(5,:),'r',
         Cd_simulation(1,:),sim_result(5,:),'g',
         Cd(1,:),DD1K_fluid_loss_at_max_buffer(20,:),'r',
         Cd_simulation(1,:),sim_result(20,:),'g',
         a,b1,'r',a,b2,'g', [0 100], [CLPd CLPd], 'b-.');
axis([0 100 1E-07 1]);
xlabel('Link Capacity per Source (cells/sec)'); ylabel('Loss Probability');
text(79, 0.6, 'Analysis'); text(79, 0.3, 'Simulation');
whitebg;
```

D. SCHEDULERS PERFORMANCE IN THE WIRELINE SYSTEM

```
% General constants
CLPs=1E-3;
CLPv=5E-5;
CLPd=1E-6;
Es=74.9;
Ev=177.8;
Ed=3.85;
Cw=1833.3;
% Simulation results of wireline case with 0 data sources (Nd=0)
N1_peak_Nd0=  [0 2 5 6];   N2_peak_Nd0=  [10 7  2  0];
N1_static_Nd0=[0 2 5 7 8]; N2_static_Nd0=[19 14 7  2 0];
N1_STE_Nd0=   [0 2 5 7 8]; N2_STE_Nd0=   [21 16 9  4 2];
N1_BCLPR_Nd0= [0 2 5 7 8]; N2_BCLPR_Nd0= [22 16 9  5 2];
N1_STEBR_Nd0= [0 2 5 7 8]; N2_STEBR_Nd0= [22 17 10 5 2];
% Interpolate the peak allocation scheme
N1i_peak_Nd0=0:0.1:6.3;
N1i_peak_Nd0(length(N1i_peak_Nd0)-2)=6; N1i_peak_Nd0(length(N1i_peak_Nd0)-1)=0;
N1i_peak_Nd0(length(N1i_peak_Nd0))=0; %close the polygon with 2 more points
N2i_peak_Nd0=  interp1(N1_peak_Nd0,   N2_peak_Nd0,   N1i_peak_Nd0, 'cubic');
N2i_peak_Nd0(length(N1i_peak_Nd0)-2)=0; N2i_peak_Nd0(length(N1i_peak_Nd0)-1)=0;
N2i_peak_Nd0(length(N1i_peak_Nd0))=  N2i_peak_Nd0(1);
% Interpolate all the other schemes
N1i_Nd0=0:0.1:8.3;
N1i_Nd0(length(N1i_Nd0)-2)=8; N1i_Nd0(length(N1i_Nd0)-1)=0;
N1i_Nd0(length(N1i_Nd0))=0; %close the polygon with 3 more points
N2i_static_Nd0=interp1(N1_static_Nd0, N2_static_Nd0, N1i_Nd0, 'cubic');
N2i_STE_Nd0=   interp1(N1_STE_Nd0,    N2_STE_Nd0,    N1i_Nd0, 'cubic');
N2i_BCLPR_Nd0= interp1(N1_BCLPR_Nd0,  N2_BCLPR_Nd0,  N1i_Nd0, 'cubic');
N2i_STEBR_Nd0= interp1(N1_STEBR_Nd0,  N2_STEBR_Nd0,  N1i_Nd0, 'cubic');
N2i_static_Nd0(length(N1i_Nd0)-2)=0; N2i_static_Nd0(length(N1i_Nd0)-1)=0;
N2i_static_Nd0(length(N1i_Nd0))=N2i_static_Nd0(1);
N2i_STE_Nd0(length(N1i_Nd0)-2)=0;    N2i_STE_Nd0(length(N1i_Nd0)-1)=0;
N2i_STE_Nd0(length(N1i_Nd0))=   N2i_STE_Nd0(1);
N2i_BCLPR_Nd0(length(N1i_Nd0)-2)=0;  N2i_BCLPR_Nd0(length(N1i_Nd0)-1)=0;
N2i_BCLPR_Nd0(length(N1i_Nd0))= N2i_BCLPR_Nd0(1);
N2i_STEBR_Nd0(length(N1i_Nd0)-2)=0;  N2i_STEBR_Nd0(length(N1i_Nd0)-1)=0;
N2i_STEBR_Nd0(length(N1i_Nd0))= N2i_STEBR_Nd0(1);
figure;
fill(N1i_Nd0, N2i_STEBR_Nd0, [.15 .15 .15]);
patch(N1i_Nd0, N2i_BCLPR_Nd0, [.35 .35 .35]);
patch(N1i_Nd0, N2i_STE_Nd0, [.55 .55 .55]);
patch(N1i_Nd0, N2i_static_Nd0, [.75 .75 .75]);
patch(N1i_peak_Nd0, N2i_peak_Nd0, [.95 .95 .95]);
xlabel('Number of Video Sources, Nv');
ylabel('Number of Speech Sources, Ns');
axis([0 8 0 24]);
grid on;
text(1.0,  5.0, 'PEAK');
text(2.0,  9.0, 'STATIC');
text(3.1, 11.8, 'STE');
text(0.2, 21.0, 'BCLPR');
text(7.0,  4.3, 'BCLPR');
text(3.8, 13.0, 'STEBR');
% Server Normalized Throughput
[Nv_Nd0,Ns_Nd0]=meshgrid(0:.05:10,0:.05:25);
```

```
rho_Nd0=(Nv_Nd0*Ev*(1-CLPv)+Ns_Nd0*Es*(1-CLPs))/Cw;
[M,N]=size(rho_Nd0);
for i=1:M,
   for j=1:N,
      if rho_Nd0(i,j)>1.01,
         rho_Nd0(i,j)=NaN;
      end;
   end;
end;
v_Nd0=[0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0];
% Mean allocation
N1_mean_Nd0 = [ 0  1  2  3  4  5  6  7  8  9 10];
N2_mean_Nd0 = [24 22 19 17 14 12 10  7  5  3  0];
N1i_mean_Nd0 = 0:0.1:10;
N2i_mean_Nd0 = interp1(N1_mean_Nd0, N2_mean_Nd0, N1i_mean_Nd0, 'cubic');
ROHi_mean_Nd0=(N1i_mean_Nd0*Ev*(1-CLPv)+N2i_mean_Nd0*Es*(1-CLPs))/Cw;
% For colored plot
figure;
pcolor(Nv_Nd0,Ns_Nd0,rho_Nd0);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9]);
xlabel('Number of Video Sources, Nv');
ylabel('Number of Speech Sources, Ns');
grid on;
hold on;
hndl_Nd0=plot(N1i_peak_Nd0, N2i_peak_Nd0, 'g-', N1i_Nd0, N2i_static_Nd0, 'g-',
         N1i_Nd0, N2i_STE_Nd0, 'g-', N1i_Nd0, N2i_BCLPR_Nd0, 'g-', N1i_Nd0,
         N2i_STEBR_Nd0, 'g-', N1i_mean_Nd0, N2i_mean_Nd0, 'y-');
set(hndl_Nd0,'LineWidth',2);
text(2.3,  4.3, 'PEAK');
text(2.6,  9.7, 'STATIC');
text(3.2, 11.5, 'STE');
text(0.2, 21.0, 'BCLPR');
text(6.5,  5.3, 'BCLPR');
text(4.2, 12.5, 'STEBR');
hold off;

% Simulation results of wireline case with 100 data sources (Nd=100)
N1_static_Nd100=[0 3];       N2_static_Nd100=[9  1];
N1_STE_Nd100=   [0 2 5 7 8]; N2_STE_Nd100=   [18 13 7 2 0];
N1_BCLPR_Nd100= [0 2 5 7 8]; N2_BCLPR_Nd100= [19 13 7 2 0];
N1_STEBR_Nd100= [0 2 5 7 8]; N2_STEBR_Nd100= [19 14 7 2 0];
% Interpolate the static allocation scheme
N1i_static_Nd100=0:0.1:3.3;
N1i_static_Nd100(length(N1i_static_Nd100)-2)=3;
N1i_static_Nd100(length(N1i_static_Nd100)-1)=0;
N1i_static_Nd100(length(N1i_static_Nd100))=0; %close the polygon w/ 2 more points
N2i_static_Nd100=interp1(N1_static_Nd100, N2_static_Nd100, N1i_static_Nd100, 'linear');
N2i_static_Nd100(length(N1i_static_Nd100)-2)=0;
N2i_static_Nd100(length(N1i_static_Nd100)-1)=0;
N2i_static_Nd100(length(N1i_static_Nd100))=N2i_static_Nd100(1);
% Interpolate all the other schemes
N1i_Nd100=0:0.1:8.3;
N1i_Nd100(length(N1i_Nd100)-2)=8; N1i_Nd100(length(N1i_Nd100)-1)=0;
N1i_Nd100(length(N1i_Nd100))=0; %close the polygon with 3 more points
N2i_STE_Nd100=  interp1(N1_STE_Nd100,   N2_STE_Nd100,   N1i_Nd100, 'cubic');
N2i_BCLPR_Nd100=interp1(N1_BCLPR_Nd100, N2_BCLPR_Nd100, N1i_Nd100, 'cubic');
N2i_STEBR_Nd100=interp1(N1_STEBR_Nd100, N2_STEBR_Nd100, N1i_Nd100, 'cubic');
```

```
N2i_STE_Nd100(length(N1i_Nd100)-2)=0;   N2i_STE_Nd100(length(N1i_Nd100)-1)=0;
N2i_STE_Nd100(length(N1i_Nd100))=   N2i_STE_Nd100(1);
N2i_BCLPR_Nd100(length(N1i_Nd100)-2)=0; N2i_BCLPR_Nd100(length(N1i_Nd100)-1)=0;
N2i_BCLPR_Nd100(length(N1i_Nd100))= N2i_BCLPR_Nd100(1);
N2i_STEBR_Nd100(length(N1i_Nd100)-2)=0; N2i_STEBR_Nd100(length(N1i_Nd100)-1)=0;
N2i_STEBR_Nd100(length(N1i_Nd100))= N2i_STEBR_Nd100(1);
figure;
fill(N1i_Nd100, N2i_STEBR_Nd100, [.15 .15 .15]);
patch(N1i_Nd100, N2i_BCLPR_Nd100, [.35 .35 .35]);
patch(N1i_Nd100, N2i_STE_Nd100, [.55 .55 .55]);
patch(N1i_static_Nd100, N2i_static_Nd100, [.75 .75 .75]);
xlabel('Number of Video Sources, Nv');
ylabel('Number of Speech Sources, Ns');
axis([0 8 0 20]);
grid on;
text(0.5,  3.0, 'STATIC');
text(1.8,  9.0, 'STE');
text(0.2, 18.0, 'BCLPR');
text(2.6, 13.0, 'STEBR');
% Server Normalized Throughput
[Nv_Nd100,Ns_Nd100]=meshgrid(0:.05:8,0:.05:20);
rho_Nd100=(Nv_Nd100*Ev*(1-CLPv)+Ns_Nd100*Es*(1-CLPs)+100*Ed*(1-CLPd))/Cw;
[M,N]=size(rho_Nd100);
for i=1:M,
   for j=1:N,
      if rho_Nd100(i,j)>1.0001,
         rho_Nd100(i,j)=NaN;
      end;
   end;
end;
v_Nd100=[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1];
% Mean allocation
N1_mean_Nd100 = [ 0  1  2  3  4  5  6  7  8];
N2_mean_Nd100 = [19 17 14 12  9  7  5  2  0];
N1i_mean_Nd100 = 0:0.1:8;
N2i_mean_Nd100 = interp1(N1_mean_Nd100, N2_mean_Nd100, N1i_mean_Nd100, 'linear');
ROHi_mean_Nd100=(N1i_mean_Nd100*Ev*(1-CLPv)+N2i_mean_Nd100*Es*
                (1-CLPs)+100*Ed*(1-CLPd))/Cw;
% For colored plot
figure;
pcolor(Nv_Nd100,Ns_Nd100,rho_Nd100);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0]);
xlabel('Number of Video Sources, Nv');
ylabel('Number of Speech Sources, Ns');
grid on;
hold on;
hndl_Nd100=plot(N1i_static_Nd100, N2i_static_Nd100, 'g-', N1i_Nd100,
           N2i_STE_Nd100, 'g-', N1i_Nd100, N2i_BCLPR_Nd100, 'g-', N1i_Nd100,
           N2i_STEBR_Nd100, 'g-');
set(hndl_Nd100,'LineWidth',2);
text(0.7,  4.0, 'STATIC');
text(3.2,  9.0, 'STE');
text(0.2, 18.0, 'BCLPR');
text(4.0, 10.0, 'STEBR');
hold off;

%----------------------------------
```

```
% Advantage of STEBR over STE
%---------------------------------
min_Ns=0;max_Ns=25;
min_Nv=0;max_Nv=10;
min_Nd=0;max_Nd=500;
Ns_density=100;
Nv_density=100;
Ns1=linspace(min_Ns,max_Ns,Ns_density);
Nv1=linspace(min_Nv,max_Nv,Nv_density);
[Nsi,Nvi]=meshgrid(Ns1,Nv1);

% Peak allocation
Ns_peak=[10 7 2 0   9   1   0];
Nv_peak=[ 0 2 5 6   0   3   0];
Nd_peak=[ 0 0 0 0 100 100 450];
Ndi_peak=griddata(Ns_peak, Nv_peak, Nd_peak, Nsi, Nvi, 'cubic');

%Static Allocation
Ns_static=[19 14 7 2 0  18  13   7   2   0   0];
Nv_static=[ 0  2 5 7 8   0   2   5   7   8   0];
Nd_static=[ 0  0 0 0 0 100 100 100 100 100 475];
Ndi_static=griddata(Ns_static, Nv_static, Nd_static, Nsi, Nvi, 'cubic');

% STE Allocation
Ns_STE=[21 16 9 4 2  18  13   7   2   0   0];
Nv_STE=[ 0  2 5 7 8   0   2   5   7   8   0];
Nd_STE=[ 0  0 0 0 0 100 100 100 100 100 475];
Ndi_STE=griddata(Ns_STE, Nv_STE, Nd_STE, Nsi, Nvi, 'cubic');

% BCLPR Allocation
Ns_BCLPR=[22 16 9 5 2  19  13   7   2   0   0];
Nv_BCLPR=[ 0  2 5 7 8   0   2   5   7   8   0];
Nd_BCLPR=[ 0  0 0 0 0 100 100 100 100 100 475];
Ndi_BCLPR=griddata(Ns_BCLPR, Nv_BCLPR, Nd_BCLPR, Nsi, Nvi, 'cubic');

% STEBR Allocation
Ns_STEBR=[22 17 10 5 2  19  14   7   2   0   0];
Nv_STEBR=[ 0  2  5 7 8   0   2   5   7   8   0];
Nd_STEBR=[ 0  0  0 0 0 100 100 100 100 100 475];
Ndi_STEBR=griddata(Ns_STEBR, Nv_STEBR, Nd_STEBR, Nsi, Nvi, 'cubic');

% Difference between STEBR and STE
Ndi_STE   =griddata(Ns_STE,   Nv_STE,   Nd_STE,   Nsi,Nvi, 'linear');
Ndi_STEBR =griddata(Ns_STEBR, Nv_STEBR, Nd_STEBR, Nsi,Nvi, 'linear');
figure;
pcolor(Nsi,Nvi,Ndi_STEBR-Ndi_STE);
shading interp;
view(0,90);
colorbar;
axis([min_Ns max_Ns min_Nv max_Nv min_Nd 60]);
xlabel('Speech Sources, Ns');
ylabel('Video Sources, Nv');
```

E. SCHEDULERS PERFORMANCE IN THE WIRELESS SYSTEM

```
% General constants
CLPs=1E-3;
CLPv=5E-5;
CLPd=1E-6;
Es=74.9;
Ev=177.8;
Ed=3.85;
Cw=1833.3;

% Simulation results of mobile scenario 1
% Partial Remote Status
N1_PEAK_sc1_partial=[ 0  1  2  3  4];
N2_PEAK_sc1_partial=[ 4  3  2  1  0];
N1_STATIC_sc1_partial=[ 2  3  4  5 10 15 20 30 40];
N2_STATIC_sc1_partial=[49 44 39 36 30 28 25 19 15];
N1_STE_sc1_partial=[  0  10  20  30  40];
N2_STE_sc1_partial=[165 155 150 144 137];
N1_BCLPR_sc1_partial=[ 2 10 20 30 40];
N2_BCLPR_sc1_partial=[45 40 29 23 18];
N1_STEBR_sc1_partial=[  0  10  20  30  40];
N2_STEBR_sc1_partial=[210 197 185 177 167];
% Interpolation of the allocation scheme
N1i_sc1_partial=0:0.05:50;
N2i_PEAK_sc1_partial=  interp1(N1_PEAK_sc1_partial,   N2_PEAK_sc1_partial,
                       N1i_sc1_partial, 'cubic');
N2i_STATIC_sc1_partial=interp1(N1_STATIC_sc1_partial, N2_STATIC_sc1_partial,
                       N1i_sc1_partial, 'cubic');
N2i_STE_sc1_partial=   interp1(N1_STE_sc1_partial,    N2_STE_sc1_partial,
                       N1i_sc1_partial, 'cubic');
N2i_BCLPR_sc1_partial= interp1(N1_BCLPR_sc1_partial,  N2_BCLPR_sc1_partial,
                       N1i_sc1_partial, 'cubic');
N2i_STEBR_sc1_partial= interp1(N1_STEBR_sc1_partial,  N2_STEBR_sc1_partial,
                       N1i_sc1_partial, 'cubic');
% Admissible Region and Throughput
[N1_sc1,N2_sc1]=meshgrid(0:0.5:40,0:1:220);
S_sc1=(2*Es*(1-CLPs)+3*Ev*(1-CLPv)+(N1_sc1+N2_sc1+3)*Ed*(1-CLPd))/Cw;
[M,N]=size(S_sc1);
for i=1:M,
   for j=1:N,
      if S_sc1(i,j)>0.901,
         S_sc1(i,j)=NaN;
      end;
   end;
end;
v_sc1=[0.3 0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc1,N2_sc1,S_sc1);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc1);
xlabel('N1 (Data Sources at the CP)');
ylabel('N2 (Data Sources at Remote 4)');
grid on;
hold on;
```

```
hndl_sc1=plot(N1i_sc1_partial, N2i_PEAK_sc1_partial, 'g-', N1i_sc1_partial,
        N2i_STATIC_sc1_partial, 'g-', N1i_sc1_partial, N2i_STE_sc1_partial,
        'g-', N1i_sc1_partial, N2i_BCLPR_sc1_partial, 'g-', N1i_sc1_partial,
        N2i_STEBR_sc1_partial, 'g-');
set(hndl_sc1,'LineWidth',2);
text( 1.5,   6.0, 'PEAK');
text( 5.2,  25.0, 'STATIC');
text(30.0, 148.0, 'STE');
text( 5.2,  48.0, 'BCLPR');
text(25.0, 185.0, 'STEBR');
hold off;
% Complete Remote Status
N1_STATIC_sc1_complete=[ 0 10 20 30 40];
N2_STATIC_sc1_complete=[64 60 58 48 38];
N1_STE_sc1_complete=[  0  10  20  30  40];
N2_STE_sc1_complete=[203 198 183 168 158];
N1_BCLPR_sc1_complete=[  0  10  20  30  40];
N2_BCLPR_sc1_complete=[120 108  98  90  78];
N1_STEBR_sc1_complete=[  0  10  20  30  40];
N2_STEBR_sc1_complete=[203 198 185 173 163];
% Interpolation of the allocation scheme
N1i_sc1_complete=0:0.05:50;
N2i_STATIC_sc1_complete=interp1(N1_STATIC_sc1_complete, N2_STATIC_sc1_complete,
                        N1i_sc1_complete, 'cubic');
N2i_STE_sc1_complete=   interp1(N1_STE_sc1_complete,    N2_STE_sc1_complete,
                        N1i_sc1_complete, 'cubic');
N2i_BCLPR_sc1_complete= interp1(N1_BCLPR_sc1_complete,  N2_BCLPR_sc1_complete,
                        N1i_sc1_complete, 'cubic');
N2i_STEBR_sc1_complete= interp1(N1_STEBR_sc1_complete,  N2_STEBR_sc1_complete,
                        N1i_sc1_complete, 'cubic');
% Admissible Region and Throughput
[N1_sc1,N2_sc1]=meshgrid(0:0.5:40,0:1:220);
S_sc1=(2*Es*(1-CLPs)+3*Ev*(1-CLPv)+(N1_sc1+N2_sc1+3)*Ed*(1-CLPd))/Cw;
[M,N]=size(S_sc1);
for i=1:M,
   for j=1:N,
      if S_sc1(i,j)>0.901,
         S_sc1(i,j)=NaN;
      end;
   end;
end;
v_sc1=[0.3 0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc1,N2_sc1,S_sc1);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc1);
xlabel('N1 (Data Sources at the CP)');
ylabel('N2 (Data Sources at Remote 4)');
grid on;
hold on;
hndl_sc1=plot(N1i_sc1_complete, N2i_STATIC_sc1_complete, 'y-',
         N1i_sc1_complete, N2i_STE_sc1_complete, 'y-', N1i_sc1_complete,
         N2i_BCLPR_sc1_complete, 'y-', N1i_sc1_complete,
         N2i_STEBR_sc1_complete, 'y-');
set(hndl_sc1,'LineWidth',2);
text( 5.2,  67.0, 'STATIC');
text(30.0, 160.5, 'STE');
text( 5.2, 117.0, 'BCLPR');
```

```
text(30.0, 176.5, 'STEBR');
hold off;
% Combine Plot - Partial & Complete
v_sc1=[0.3 0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc1,N2_sc1,S_sc1);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc1);
xlabel('N1 (Data Sources at the CP)');
ylabel('N2 (Data Sources at Remote 4)');
grid on;
hold on;
hndl_sc1=plot(N1i_sc1_partial, N2i_PEAK_sc1_partial, 'g-', N1i_sc1_partial,
        N2i_STATIC_sc1_partial, 'g-', N1i_sc1_partial, N2i_STE_sc1_partial,
        'g-', N1i_sc1_partial, N2i_BCLPR_sc1_partial, 'g-', N1i_sc1_partial,
        N2i_STEBR_sc1_partial, 'g-', N1i_sc1_complete,
        N2i_STATIC_sc1_complete, 'y-', N1i_sc1_complete, N2i_STE_sc1_complete,
        'y-', N1i_sc1_complete, N2i_BCLPR_sc1_complete, 'y-',
        N1i_sc1_complete, N2i_STEBR_sc1_complete, 'y-');
set(hndl_sc1,'LineWidth',2);
text( 1.5,   6.0, 'PEAK');
text( 5.2,  25.0, 'STATICp');
text(30.0, 148.0, 'STEp');
text( 5.2,  48.0, 'BCLPRp');
text(25.0, 185.0, 'STEBRp');
text( 5.2,  67.0, 'STATICc');
text(30.0, 160.5, 'STEc');
text( 5.2, 117.0, 'BCLPRc');
text(30.0, 176.5, 'STEBRc');
hold off;

% Simulation results of mobile scenario 2
% Partial Remote Status
N1_STATIC_sc2_partial=[ 0  1  2  3 3.02];
N2_STATIC_sc2_partial=[65 40 32 19    0];
N1_STE_sc2_partial=[  0   4  8 12 12.01];
N2_STE_sc2_partial=[165 135 77 20     0];
N1_BCLPR_sc2_partial=[  0  4  8 12 12.01];
N2_BCLPR_sc2_partial=[135 92 60 11     0];
N1_STEBR_sc2_partial=[  0   4  8 12 12.01];
N2_STEBR_sc2_partial=[187 148 88 23     0];
% Interpolation of the allocation scheme
N1i_sc2_partial=0:0.01:14;
N2i_STATIC_sc2_partial=interp1(N1_STATIC_sc2_partial, N2_STATIC_sc2_partial,
                       N1i_sc2_partial, 'cubic');
N2i_STE_sc2_partial=   interp1(N1_STE_sc2_partial,    N2_STE_sc2_partial,
                       N1i_sc2_partial, 'cubic');
N2i_BCLPR_sc2_partial= interp1(N1_BCLPR_sc2_partial,  N2_BCLPR_sc2_partial,
                       N1i_sc2_partial, 'cubic');
N2i_STEBR_sc2_partial= interp1(N1_STEBR_sc2_partial,  N2_STEBR_sc2_partial,
                       N1i_sc2_partial, 'cubic');
% Admissible Region and Throughput
[N1_sc2,N2_sc2]=meshgrid(0:.1:14,0:1:400);
S_sc2=(N1_sc2*Es*(1-CLPs)+(N2_sc2+20)*Ed*(1-CLPd))/Cw;
[M,N]=size(S_sc2);
for i=1:M,
   for j=1:N,
      if S_sc2(i,j)>0.601,
```

```
        S_sc2(i,j)=NaN;
      end;
   end;
end;
v_sc2=[0.0 0.1 0.2 0.3 0.4 0.5 0.6];
figure;
pcolor(N1_sc2,N2_sc2,S_sc2);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc2);
xlabel('N1 (Speech Sources at the CP)');
ylabel('N2 (Data Sources at the CP)');
grid on;
hold on;
hndl_sc2=plot(N1i_sc2_partial, N2i_STATIC_sc2_partial, 'g-', N1i_sc2_partial,
         N2i_STE_sc2_partial, 'g-', N1i_sc2_partial, N2i_BCLPR_sc2_partial, 'g-',
         N1i_sc2_partial, N2i_STEBR_sc2_partial, 'g-');
set(hndl_sc2,'LineWidth',2);
text(0.8,  49.0, 'STATIC');
text(0.8, 166.0, 'STE');
text(0.8, 132.0, 'BCLPR');
text(0.8, 188.0, 'STEBR');
hold off;
% Complete Remote Status
N1_STATIC_sc2_complete=[  1   4   8 12];
N2_STATIC_sc2_complete=[190 105  70  5];
N1_STE_sc2_complete=[  0   4   8  12  13  14];
N2_STE_sc2_complete=[360 290 210 130 100  70];
N1_BCLPR_sc2_complete=[  0   4   8  12  13  14];
N2_BCLPR_sc2_complete=[220 145  90  40  20   0];
N1_STEBR_sc2_complete=[  0   4   8  12  13  14];
N2_STEBR_sc2_complete=[360 292 213 135 107  80];
% Interpolation of the allocation scheme
N1i_sc2_complete=0:0.1:20;
N2i_STATIC_sc2_complete=interp1(N1_STATIC_sc2_complete, N2_STATIC_sc2_complete,
                        N1i_sc2_complete, 'cubic');
N2i_STE_sc2_complete=   interp1(N1_STE_sc2_complete,    N2_STE_sc2_complete,
                        N1i_sc2_complete, 'cubic');
N2i_BCLPR_sc2_complete= interp1(N1_BCLPR_sc2_complete,  N2_BCLPR_sc2_complete,
                        N1i_sc2_complete, 'cubic');
N2i_STEBR_sc2_complete= interp1(N1_STEBR_sc2_complete,  N2_STEBR_sc2_complete,
                        N1i_sc2_complete, 'cubic');
% Admissible Region and Throughput
[N1_sc2,N2_sc2]=meshgrid(0:.1:14,0:1:400);
S_sc2=(N1_sc2*Es*(1-CLPs)+(N2_sc2+20)*Ed*(1-CLPd))/Cw;
[M,N]=size(S_sc2);
for i=1:M,
   for j=1:N,
      if S_sc2(i,j)>0.901,
         S_sc2(i,j)=NaN;
      end;
   end;
end;
v_sc2=[0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc2,N2_sc2,S_sc2);
shading interp;
colormap(cool);
col=colorbar;
```

```
set(col,'ytick',v_sc2);
xlabel('N1 (Speech Sources at the CP)');
ylabel('N2 (Data Sources at the CP)');
grid on;
hold on;
hndl_sc2=plot(N1i_sc2_complete, N2i_STATIC_sc2_complete, 'y-',
         N1i_sc2_complete, N2i_STE_sc2_complete, 'y-', N1i_sc2_complete,
         N2i_BCLPR_sc2_complete, 'y-', N1i_sc2_complete,
         N2i_STEBR_sc2_complete, 'y-');
set(hndl_sc2,'LineWidth',2);
text( 2.0, 120.0, 'STATIC');
text(10.0, 140.0, 'STE');
text( 0.3, 220.0, 'BCLPR');
text(11.3, 150.0, 'STEBR');
hold off;
% Combine Plot - Partial & Complete
v_sc2=[0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc2,N2_sc2,S_sc2);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc2);
xlabel('N1 (Speech Sources at the CP)');
ylabel('N2 (Data Sources at the CP)');
grid on;
hold on;
hndl_sc2=plot(N1i_sc2_partial, N2i_STATIC_sc2_partial, 'g-', N1i_sc2_partial,
         N2i_STE_sc2_partial, 'g-', N1i_sc2_partial, N2i_BCLPR_sc2_partial, 'g-',
         N1i_sc2_partial, N2i_STEBR_sc2_partial, 'g-', N1i_sc2_complete,
         N2i_STATIC_sc2_complete, 'y-', N1i_sc2_complete, N2i_STE_sc2_complete,
         'y-', N1i_sc2_complete, N2i_BCLPR_sc2_complete, 'y-', N1i_sc2_complete,
         N2i_STEBR_sc2_complete, 'y-');
set(hndl_sc2,'LineWidth',2);
text( 0.8,  49.0, 'STATICp');
text( 4.7, 106.0, 'STEp');
text( 0.3, 138.0, 'BCLPRp');
text( 5.4, 136.0, 'STEBRp');
text( 2.3, 125.0, 'STATICc');
text(10.0, 140.0, 'STEc');
text( 0.3, 220.0, 'BCLPRc');
text(11.3, 150.0, 'STEBRc');
hold off;

% Simulation results of mobile scenario 3
% Partial Remote Status
N1_PEAK_sc3_partial=[ 0  1];
N2_PEAK_sc3_partial=[ 1  0];
N1_STATIC_sc3_partial=[ 1  4 10 20 30 40];
N2_STATIC_sc3_partial=[43 30 18 11  8  5];
N1_STE_sc3_partial=[  0  10  20  30  40];
N2_STE_sc3_partial=[175 160 150 145 135];
N1_BCLPR_sc3_partial=[ 4 10 20 30 40];
N2_BCLPR_sc3_partial=[33 31 25 24 21];
N1_STEBR_sc3_partial=[  0  10  20  30  40];
N2_STEBR_sc3_partial=[200 187 180 170 160];
% Interpolation of the allocation scheme
N1i_sc3_partial=0:0.1:50;
N2i_STATIC_sc3_partial=interp1(N1_STATIC_sc3_partial, N2_STATIC_sc3_partial,
                        N1i_sc3_partial, 'cubic');
```

```
N2i_STE_sc3_partial=   interp1(N1_STE_sc3_partial,    N2_STE_sc3_partial,
                       N1i_sc3_partial, 'cubic');
N2i_BCLPR_sc3_partial= interp1(N1_BCLPR_sc3_partial,  N2_BCLPR_sc3_partial,
                       N1i_sc3_partial, 'cubic');
N2i_STEBR_sc3_partial= interp1(N1_STEBR_sc3_partial,  N2_STEBR_sc3_partial,
                       N1i_sc3_partial, 'cubic');
% Admissible Region and Throughput
[N1_sc3,N2_sc3]=meshgrid(0:.1:40,0:1:300);
S_sc3=(2*Es*(1-CLPs)+1*Ev*(1-CLPv)+(N1_sc3+N2_sc3+11)*Ed*(1-CLPd))/Cw;
[M,N]=size(S_sc3);
for i=1:M,
   for j=1:N,
      if S_sc3(i,j)>0.701,
         S_sc3(i,j)=NaN;
      end;
   end;
end;
v_sc3=[0.2 0.3 0.4 0.5 0.6 0.7];
figure;
pcolor(N1_sc3,N2_sc3,S_sc3);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc3);
xlabel('N1 (Data Sources at the CP)');
ylabel('N2 (Data Sources at Remote 5)');
grid on;
hold on;
hndl_sc3=plot(N1_PEAK_sc3_partial, N2_PEAK_sc3_partial, 'g-', N1i_sc3_partial,
         N2i_STATIC_sc3_partial, 'g-', N1i_sc3_partial, N2i_STE_sc3_partial,
         'g-', N1i_sc3_partial, N2i_BCLPR_sc3_partial, 'g-', N1i_sc3_partial,
         N2i_STEBR_sc3_partial, 'g-');
set(hndl_sc3,'LineWidth',2);
text( 1.0,   5.0, 'PEAK');
text(15.1,  18.5, 'STATIC');
text(15.1, 158.5, 'STE');
text(15.1,  35.0, 'BCLPR');
text(15.1, 188.0, 'STEBR');
hold off;
% Complete Remote Status
N1_STATIC_sc3_complete=[ 0 10 20 30 40];
N2_STATIC_sc3_complete=[98 92 82 75 62];
N1_STE_sc3_complete=[  2  20  40];
N2_STE_sc3_complete=[290 270 247];
N1_BCLPR_sc3_complete=[  0  10  20  30  40];
N2_BCLPR_sc3_complete=[115 105 100  92  85];
N1_STEBR_sc3_complete=[  2  20  40];
N2_STEBR_sc3_complete=[290 272 250];
% Interpolation of the allocation scheme
N1i_sc3_complete=0:0.1:50;
N2i_STATIC_sc3_complete=interp1(N1_STATIC_sc3_complete, N2_STATIC_sc3_complete,
                        N1i_sc3_complete, 'cubic');
N2i_STE_sc3_complete=   interp1(N1_STE_sc3_complete,    N2_STE_sc3_complete,
                        N1i_sc3_complete, 'cubic');
N2i_BCLPR_sc3_complete= interp1(N1_BCLPR_sc3_complete,  N2_BCLPR_sc3_complete,
                        N1i_sc3_complete, 'cubic');
N2i_STEBR_sc3_complete= interp1(N1_STEBR_sc3_complete,  N2_STEBR_sc3_complete,
                        N1i_sc3_complete, 'cubic');
% Admissible Region and Throughput
[N1_sc3,N2_sc3]=meshgrid(0:.1:40,0:1:300);
```

```
S_sc3=(2*Es*(1-CLPs)+1*Ev*(1-CLPv)+(N1_sc3+N2_sc3+11)*Ed*(1-CLPd))/Cw;
[M,N]=size(S_sc3);
for i=1:M,
   for j=1:N,
      if S_sc3(i,j)>0.901,
         S_sc3(i,j)=NaN;
      end;
   end;
end;
v_sc3=[0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc3,N2_sc3,S_sc3);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc3);
xlabel('N1 (Data Sources at the CP)');
ylabel('N2 (Data Sources at Remote 5)');
grid on;
hold on;
hndl_sc3=plot(N1i_sc3_complete, N2i_STATIC_sc3_complete, 'y-',
         N1i_sc3_complete, N2i_STE_sc3_complete, 'y-', N1i_sc3_complete,
         N2i_BCLPR_sc3_complete, 'y-', N1i_sc3_complete,
         N2i_STEBR_sc3_complete, 'y-');
set(hndl_sc3,'LineWidth',2);
text( 1.0,  90.0, 'STATIC');
text(30.0, 250.0, 'STE');
text( 1.0, 120.0, 'BCLPR');
text(30.0, 268.0, 'STEBR');
hold off;
% Combine Plot - Partial & Complete
v_sc3=[0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc3,N2_sc3,S_sc3);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc3);
xlabel('N1 (Data Sources at the CP)');
ylabel('N2 (Data Sources at Remote 5)');
grid on;
hold on;
hndl_sc3=plot(N1_PEAK_sc3_partial, N2_PEAK_sc3_partial, 'g-', N1i_sc3_partial,
         N2i_STATIC_sc3_partial, 'g-', N1i_sc3_partial, N2i_STE_sc3_partial,
         'g-', N1i_sc3_partial, N2i_BCLPR_sc3_partial, 'g-', N1i_sc3_partial,
         N2i_STEBR_sc3_partial, 'g-', N1i_sc3_complete,
         N2i_STATIC_sc3_complete, 'y-', N1i_sc3_complete, N2i_STE_sc3_complete,
         'y-', N1i_sc3_complete, N2i_BCLPR_sc3_complete, 'y-',
         N1i_sc3_complete, N2i_STEBR_sc3_complete, 'y-');
set(hndl_sc3,'LineWidth',2);
text( 1.0,   5.0, 'PEAK');
text( 5.0,  14.5, 'STATICp');
text(30.0, 151.0, 'STEp');
text(15.1,  35.0, 'BCLPRp');
text(34.0, 171.8, 'STEBRp');
text( 5.0,  86.0, 'STATICc');
text(30.0, 250.0, 'STEc');
text(15.1, 108.0, 'BCLPRc');
text(34.0, 261.5, 'STEBRc');
hold off;
```

```
% Simulation results of mobile scenario 4
% Partial Remote Status
N1_PEAK_sc4_partial=[ 0  1];
N2_PEAK_sc4_partial=[ 3  0];
N1_STATIC_sc4_partial=[ 0  1  2 2.01];
N2_STATIC_sc4_partial=[ 5  2  1    0];
N1_STE_sc4_partial=[ 0  1  2  3 3.01];
N2_STE_sc4_partial=[ 9  7  4  1    0];
N1_BCLPR_sc4_partial=[ 0  1  2  3];
N2_BCLPR_sc4_partial=[ 9  5  3  0];
N1_STEBR_sc4_partial=[ 0  1  2  3  4];
N2_STEBR_sc4_partial=[11  8  6  3  0];
% Interpolation of the allocation scheme
N1i_sc4_partial=0:0.01:4;
N2i_STATIC_sc4_partial=interp1(N1_STATIC_sc4_partial, N2_STATIC_sc4_partial,
                        N1i_sc4_partial, 'cubic');
N2i_STE_sc4_partial=    interp1(N1_STE_sc4_partial,    N2_STE_sc4_partial,
                        N1i_sc4_partial, 'cubic');
N2i_BCLPR_sc4_partial= interp1(N1_BCLPR_sc4_partial,  N2_BCLPR_sc4_partial,
                        N1i_sc4_partial, 'cubic');
N2i_STEBR_sc4_partial= interp1(N1_STEBR_sc4_partial,  N2_STEBR_sc4_partial,
                        N1i_sc4_partial, 'cubic');
% Admissible Region and Throughput
[N1_sc4,N2_sc4]=meshgrid(0:0.01:4,0:0.01:12);
S_sc4=((N2_sc4+3)*Es*(1-CLPs)+(N1_sc4+3)*Ev*(1-CLPv))/Cw;
[M,N]=size(S_sc4);
for i=1:M,
   for j=1:N,
      if S_sc4(i,j)>0.900001,
         S_sc4(i,j)=NaN;
      end;
   end;
end;
v_sc4=[0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc4,N2_sc4,S_sc4);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc4);
xlabel('N1 (Video Sources at the CP)');
ylabel('N2 (Speech Sources at Remote 2)');
grid on;
hold on;
hndl_sc4=plot(N1_PEAK_sc4_partial, N2_PEAK_sc4_partial, 'g-', N1i_sc4_partial,
         N2i_STATIC_sc4_partial, 'g-', N1i_sc4_partial, N2i_STE_sc4_partial,
         'g-', N1i_sc4_partial, N2i_BCLPR_sc4_partial, 'g-', N1i_sc4_partial,
         N2i_STEBR_sc4_partial, 'g-');
text( 0.3,  2.3, 'PEAK');
text( 0.7,  3.0, 'STATIC');
text( 1.7,  5.2, 'STE');
text( 1.3,  4.5, 'BCLPR');
text( 2.1,  6.1, 'STEBR');
set(hndl_sc4,'LineWidth',2);
hold off;
% Complete Remote Status
N1_STATIC_sc4_complete=[ 0  1  2 2.01];
N2_STATIC_sc4_complete=[ 6  3  1    0];
N1_STE_sc4_complete=[ 0  1  2 2.01];
```

```
N2_STE_sc4_complete=[ 6  4  2    0];
N1_BCLPR_sc4_complete=[ 0  1  2 2.01];
N2_BCLPR_sc4_complete=[ 6  4  1    0];
N1_STEBR_sc4_complete=[ 0  1  2 2.01];
N2_STEBR_sc4_complete=[ 6  4  2    0];
% Interpolation of the allocation scheme
N1i_sc4_complete=0:0.01:4;
N2i_STATIC_sc4_complete=interp1(N1_STATIC_sc4_complete, N2_STATIC_sc4_complete,
                        N1i_sc4_complete, 'cubic');
N2i_STE_sc4_complete=   interp1(N1_STE_sc4_complete,    N2_STE_sc4_complete,
                        N1i_sc4_complete, 'cubic');
N2i_BCLPR_sc4_complete= interp1(N1_BCLPR_sc4_complete,  N2_BCLPR_sc4_complete,
                        N1i_sc4_complete, 'linear');
N2i_STEBR_sc4_complete= interp1(N1_STEBR_sc4_complete,  N2_STEBR_sc4_complete,
                        N1i_sc4_complete, 'cubic');
% Admissible Region and Throughput
[N1_sc4,N2_sc4]=meshgrid(0:0.01:4,0:0.01:12);
S_sc4=((N2_sc4+3)*Es*(1-CLPs)+(N1_sc4+3)*Ev*(1-CLPv))/Cw;
[M,N]=size(S_sc4);
for i=1:M,
   for j=1:N,
      if S_sc4(i,j)>0.70001,
         S_sc4(i,j)=NaN;
      end;
   end;
end;
v_sc4=[0.4 0.5 0.6 0.7];
figure;
pcolor(N1_sc4,N2_sc4,S_sc4);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc4);
xlabel('N1 (Video Sources at the CP)');
ylabel('N2 (Speech Sources at Remote 2)');
grid on;
hold on;
hndl_sc4=plot(N1i_sc4_complete, N2i_STATIC_sc4_complete, 'y-',
         N1i_sc4_complete, N2i_STE_sc4_complete, 'y-', N1i_sc4_complete,
         N2i_BCLPR_sc4_complete, 'y-', N1i_sc4_complete,
         N2i_STEBR_sc4_complete, 'y-');
text( 0.2,  4.0, 'STATIC');
text( 1.7,  2.8, 'STE/STEBR');
text( 1.5,  2.0, 'BCLPR');
set(hndl_sc4,'LineWidth',2);
hold off;
% Combine Plot - Partial & Complete
[N1_sc4,N2_sc4]=meshgrid(0:0.01:4,0:0.01:12);
S_sc4=((N2_sc4+3)*Es*(1-CLPs)+(N1_sc4+3)*Ev*(1-CLPv))/Cw;
[M,N]=size(S_sc4);
for i=1:M,
   for j=1:N,
      if S_sc4(i,j)>0.900001,
         S_sc4(i,j)=NaN;
      end;
   end;
end;
v_sc4=[0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc4,N2_sc4,S_sc4);
```

```
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc4);
xlabel('N1 (Video Sources at the CP)');
ylabel('N2 (Speech Sources at Remote 2)');
grid on;
hold on;
hndl_sc4=plot(N1_PEAK_sc4_partial, N2_PEAK_sc4_partial, 'g-', N1i_sc4_partial,
        N2i_STATIC_sc4_partial, 'g-', N1i_sc4_partial, N2i_STE_sc4_partial,
        'g-', N1i_sc4_partial, N2i_BCLPR_sc4_partial, 'g-', N1i_sc4_partial,
        N2i_STEBR_sc4_partial, 'g-', N1i_sc4_complete,
        N2i_STATIC_sc4_complete, 'y-', N1i_sc4_complete, N2i_STE_sc4_complete,
        'y-', N1i_sc4_complete, N2i_BCLPR_sc4_complete, 'y-',
        N1i_sc4_complete, N2i_STEBR_sc4_complete, 'y-');
set(hndl_sc4,'LineWidth',2);
text( 0.3,  1.6, 'PEAK');
text(0.15,  3.5, 'STATICp');
text( 2.3,  3.3, 'STEp');
text( 0.9,  4.5, 'BCLPRp');
text(3.15,  2.8, 'STEBRp');
text( 0.7,  3.0, 'STATICc');
text( 1.3,  3.3, 'STEc/');
text( 1.5,  2.0, 'BCLPRc');
text( 1.5,  2.8, 'STEBRc');
hold off;

% Simulation results of mobile scenario 5
% Partial Remote Status
N1_PEAK_sc5_partial=[ 0  1  2  3  4  5  6];
N2_PEAK_sc5_partial=[ 6  5  4  3  2  1  0];
N1_STATIC_sc5_partial=[ 0  1  2  3  4  6  7.01];
N2_STATIC_sc5_partial=[ 7  6  4  3  2  1  0];
N1_STE_sc5_partial=[ 1  2  5  7 7.01];
N2_STE_sc5_partial=[ 7  5  2  1    0];
N1_BCLPR_sc5_partial=[ 0  2  5  7];
N2_BCLPR_sc5_partial=[ 7  5  2  0];
N1_STEBR_sc5_partial=[ 0  1  3  5  7  7.01];
N2_STEBR_sc5_partial=[ 7  7  5  3  1  0];
% Interpolation of the allocation scheme
N1i_sc5_partial=0:0.01:8;
N2i_STATIC_sc5_partial=interp1(N1_STATIC_sc5_partial, N2_STATIC_sc5_partial,
                        N1i_sc5_partial, 'linear');
N2i_STE_sc5_partial=   interp1(N1_STE_sc5_partial,    N2_STE_sc5_partial,
                        N1i_sc5_partial, 'linear');
N2i_BCLPR_sc5_partial= interp1(N1_BCLPR_sc5_partial,  N2_BCLPR_sc5_partial,
                        N1i_sc5_partial, 'linear');
N2i_STEBR_sc5_partial= interp1(N1_STEBR_sc5_partial,  N2_STEBR_sc5_partial,
                        N1i_sc5_partial, 'linear');
% Admissible Region and Throughput
[N1_sc5,N2_sc5]=meshgrid(0:.05:8,0:.05:8);
S_sc5=(N1_sc5+N2_sc5+1)*Ev*(1-CLPv)/Cw;
[M,N]=size(S_sc5);
for i=1:M,
   for j=1:N,
      if S_sc5(i,j)>0.91,
         S_sc5(i,j)=NaN;
      end;
   end;
end;
```

```
v_sc5=[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc5,N2_sc5,S_sc5);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc5);
xlabel('N1 (Video Sources at Remote 2)');
ylabel('N2 (Video Sources at Remote 3)');
grid on;
hold on;
hndl_sc5=plot(N1_PEAK_sc5_partial, N2_PEAK_sc5_partial, 'g-', N1i_sc5_partial,
        N2i_STATIC_sc5_partial, 'g-', N1i_sc5_partial, N2i_STE_sc5_partial,
        'g-', N1i_sc5_partial, N2i_BCLPR_sc5_partial, 'g-', N1i_sc5_partial,
        N2i_STEBR_sc5_partial, 'g-');
set(hndl_sc5,'LineWidth',2);
text( 0.1, 5.7, 'PEAK');
text( 1.2, 4.8, 'STATIC');
text( 1.2, 6.0, 'STE');
text( 0.1, 6.5, 'BCLPR');
text( 2.0, 5.6, 'STEBR');
hold off;
% Complete Remote Status
N1_STATIC_sc5_complete=[ 0  2  3  5];
N2_STATIC_sc5_complete=[ 5  3  2  0];
N1_STE_sc5_complete=[ 0  1  3  4  6];
N2_STE_sc5_complete=[ 6  4  3  1  0];
N1_BCLPR_sc5_complete=[ 0  2  4  6];
N2_BCLPR_sc5_complete=[ 6  4  2  0];
N1_STEBR_sc5_complete=[ 0  2  4  6];
N2_STEBR_sc5_complete=[ 6  4  2  0];
% Interpolation of the allocation scheme
N1i_sc5_complete=0:0.1:7;
N2i_STATIC_sc5_complete=interp1(N1_STATIC_sc5_complete, N2_STATIC_sc5_complete,
                        N1i_sc5_complete, 'linear');
N2i_STE_sc5_complete=   interp1(N1_STE_sc5_complete,    N2_STE_sc5_complete,
                        N1i_sc5_complete, 'linear');
N2i_BCLPR_sc5_complete= interp1(N1_BCLPR_sc5_complete,  N2_BCLPR_sc5_complete,
                        N1i_sc5_complete, 'linear');
N2i_STEBR_sc5_complete= interp1(N1_STEBR_sc5_complete,  N2_STEBR_sc5_complete,
                        N1i_sc5_complete, 'linear');
% Admissible Region and Throughput
[N1_sc5,N2_sc5]=meshgrid(0:.05:8,0:.05:8);
S_sc5=(N1_sc5+N2_sc5+1)*Ev*(1-CLPv)/Cw;
[M,N]=size(S_sc5);
for i=1:M,
   for j=1:N,
      if S_sc5(i,j)>0.71,
         S_sc5(i,j)=NaN;
      end;
   end;
end;
v_sc5=[0.1 0.2 0.3 0.4 0.5 0.6 0.7];
figure;
pcolor(N1_sc5,N2_sc5,S_sc5);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc5);
xlabel('N1 (Video Sources at Remote 2)');
```

```
ylabel('N2 (Video Sources at Remote 3)');
grid on;
hold on;
hndl_sc5=plot(N1i_sc5_complete, N2i_STATIC_sc5_complete, 'y-',
         N1i_sc5_complete, N2i_STE_sc5_complete, 'y-', N1i_sc5_complete,
         N2i_BCLPR_sc5_complete, 'y-', N1i_sc5_complete,
         N2i_STEBR_sc5_complete, 'y-');
set(hndl_sc5,'LineWidth',2);
text(1.4, 2.8, 'STATIC');
text(1.3, 4.0, 'STE');
text(1.5, 4.7, 'BCLPR/STEBR');
hold off;
% Combine Plot - Partial & Complete
[N1_sc5,N2_sc5]=meshgrid(0:.05:8,0:.05:8);
S_sc5=(N1_sc5+N2_sc5+1)*Ev*(1-CLPv)/Cw;
[M,N]=size(S_sc5);
for i=1:M,
   for j=1:N,
      if S_sc5(i,j)>0.91,
         S_sc5(i,j)=NaN;
      end;
   end;
end;
v_sc5=[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9];
figure;
pcolor(N1_sc5,N2_sc5,S_sc5);
shading interp;
colormap(cool);
col=colorbar;
set(col,'ytick',v_sc5);
xlabel('N1 (Video Sources at Remote 2)');
ylabel('N2 (Video Sources at Remote 3)');
grid on;
hold on;
hndl_sc5=plot(N1_PEAK_sc5_partial, N2_PEAK_sc5_partial, 'g-', N1i_sc5_partial,
         N2i_STATIC_sc5_partial, 'g-', N1i_sc5_partial, N2i_STE_sc5_partial,
         'g-', N1i_sc5_partial, N2i_BCLPR_sc5_partial, 'g-', N1i_sc5_partial,
         N2i_STEBR_sc5_partial, 'g-', N1i_sc5_complete,
         N2i_STATIC_sc5_complete, 'y-', N1i_sc5_complete, N2i_STE_sc5_complete,
         'y-', N1i_sc5_complete, N2i_BCLPR_sc5_complete, 'y-',
         N1i_sc5_complete, N2i_STEBR_sc5_complete, 'y-');
set(hndl_sc5,'LineWidth',2);
text( 1.0, 5.1, 'STATICp');
text(1.15, 6.0, 'STEp');
text( 3.6, 2.9, 'BCLPRp');
text( 3.6, 4.0, 'STEBRp');
text( 2.0, 2.5, 'STATICc');
text( 0.2, 5.0, 'STEc');
text( 1.9, 3.8, 'BCLPRc/');
text( 2.2, 3.5, 'STEBRc');
hold off;
```

APPENDIX D. OPNET SIMULATION INPUTS AND OUTPUTS

A. SIMULATION INPUT PARAMETERS

1. Terminology

Simulation parameters have the following notation:

<Node Name>.<Process Name>.<Parameter Name>.

An asterisk (*) is used to mark all the nodes or processes that have the same parameters. For example, *CP.*.active_remotes* marks all the active_remotes parameters of processes within the CP, and **.remote_MAC.p_aloha* marks parameter *p_aloha* at process *remote_MAC* of all the remotes (that use the same process model).

2. Simulation Parameters

The simulation parameters are described in the following table.

Parameter	Meaning	Possible Values	Default Value
**.active_remotes*	Number of active remotes	Non-negative	0
CP..active_remotes*	Number of active remotes	Non-negative	0
**.active_sources*	Number of active speech, video and data sources at the remotes	Non-negative	0
CP..active_sources*	Number of active speech, video and data sources at the CP	Non-negative	0
**.remote_ATM_speech_sources.cell_rate_when_active*	Rate (cells/sec) of cell generation when one speaker is active	Non-negative	85.1
**.remote_MAC.speech_CTD*	Allowed speech delay at all remotes	Non-negative	0.04
**.remote_MAC.video_CTD*	Allowed video delay at all remotes	Non-negative	0.1
**.remote_MAC.data_CTD*	Allowed data delay at all remotes	Non-negative	30
CP.CP_MAC.speech_CTD	Allowed speech delay at the CP	Non-negative	0.04
CP.CP_MAC.video_CTD	Allowed video delay at the CP	Non-negative	0.1
CP.CP_MAC.data_CTD	Allowed data delay at the CP	Non-negative	30
**.*.remote_to_remote_connections*	Connections of type R2R (*) in all remotes	0 or 1	0
CP..remote_to_remote_connections*	Connections of type R2R (*) at the CP	0 or 1	0

**.remote_MAC.remote_op_ID*	Remote identifier	1, 2, ..., 15	1
**.remote_MAC.p_aloha*	Probability of transmission in the slotted-ALOHA protocol in remote's MAC	0-1	0.1
CP.CP_MAC.MAC_scheduling_scheme	Scheduling MAC used at the CP's MAC	Static (1), Proportional (2), STE (3), BCLPR(4), STEBR(5)	0
**.remote_MAC.MAC_scheduling_scheme*	Scheduling MAC used in the remote's MAC	Static (1), Proportional (2), STE (3), BCLPR(4), STEBR(5)	1
receiver.phd_receiver.MAC_scheduling_scheme	Scheduling MAC used in a monitoring station	Static (1), Proportional (2), STE (3), BCLPR(4), STEBR(5)	1
CP.CP_MAC.expiry_time_valid	Do arriving cells at the CP in remote-to-remote connections contain deadline?	FALSE (0), TRUE (1)	0
CP.CP_MAC.PAR_information_type	Type of information in PAR fields of ALLOCATE_REQUEST signaling message and information cell	WAITING_MESSAGES_PAR (1), MESSAGES_TO_EXPIRE_PAR (2), CLP_RATIO_PAR (3)	1
**.remote_MAC.PAR_information_type*	Type of information in PAR fields of ALLOCATE_REQUEST signaling message and information cell	WAITING_MESSAGES_PAR (1), MESSAGES_TO_EXPIRE_PAR (2), CLP_RATIO_PAR (3)	1
CP.CP_MAC.scenario_index	Index of scenario	1, 2, 3, 4, 5	1
**.remote_MAC.scenario_index*	Index of scenario	1, 2, 3, 4, 5	1
receiver.phd_receiver.scenario_index	Index of scenario	1, 2, 3, 4, 5	1

(*) By default, all the active connections are of type CP to remote (for CP sources) and remote to CP (for remote sources). Connections of type remote to remote are marked by a special bitwise parameter: the rightmost bit is associated with the first connection, the next bit is associated with the second connection and so on and so forth. For example, if at Remote 6, there are five active data connections in which the first and the forth are of type remote to remote, then parameter *node_6.remote_ATM_data_sources.remote_to_remote_connections* is assigned the value 01001 (binary) or 9 (decimal).

B. REPRESENTATIVE SIMULATION OUTPUT

1. Index

- Sources: The number of sources at the CP include remote-to-remote connections. In the example shown, 10 data sources are generated locally at the CP, and one source is originated by a remote and destined to another remote.

- Output: The first part includes the report by the CP, which includes the admission controller. Every line describes the MVCI of the active source, its source node (CP is Station 0), the source class (1-speech, 2-video, 3-data), and the connection type (1-CP to remote, 2-remote to CP, 3-remote to remote), number of arrived and discarded cells. The second part includes reports by the remotes involved in the scenario.

- Control Messages: The number of available uplink control slots throughout the simulation, the number of successfully used slots, the number of slots in which collision occurred, and the number of messages that collided.

2. Printout

```
Scheduling Scheme: BCLPR
Scenario: 1
p_aloha = 0.05
Simulation Duration: 1500 sec

Input:
------
CP:         S -  1, V -  1, D - 11 (including remote-to-remote calls via CP)
Station 1: S -  1, V -  0, D -  0
Station 2: S -  0, V -  1, D -  1
Station 3: S -  0, V -  1, D - 40
Station 4: S -  0, V -  0, D -  1

Output:
-------
Report by CP: MVCI  4, Station  0, Class 1, Type 1, ARRIVED 113252, DISCARDED 0
Report by CP: MVCI  5, Station  0, Class 2, Type 1, ARRIVED 267150, DISCARDED 0
Report by CP: MVCI  6, Station  0, Class 3, Type 1, ARRIVED   5263, DISCARDED 0
Report by CP: MVCI  7, Station  0, Class 3, Type 1, ARRIVED   5496, DISCARDED 0
Report by CP: MVCI  8, Station  0, Class 3, Type 1, ARRIVED   4503, DISCARDED 0
Report by CP: MVCI  9, Station  0, Class 3, Type 1, ARRIVED   4826, DISCARDED 0
Report by CP: MVCI 10, Station  0, Class 3, Type 1, ARRIVED   5778, DISCARDED 0
Report by CP: MVCI 11, Station  0, Class 3, Type 1, ARRIVED   5836, DISCARDED 0
Report by CP: MVCI 12, Station  0, Class 3, Type 1, ARRIVED   5420, DISCARDED 0
Report by CP: MVCI 13, Station  0, Class 3, Type 1, ARRIVED   5759, DISCARDED 0
Report by CP: MVCI 14, Station  0, Class 3, Type 1, ARRIVED   3734, DISCARDED 0
Report by CP: MVCI 15, Station  0, Class 3, Type 1, ARRIVED   5246, DISCARDED 0
Report by CP: MVCI 16, Station  1, Class 1, Type 2, DISCARDED 0
Report by CP: MVCI 17, Station  2, Class 2, Type 2, DISCARDED 0
Report by CP: MVCI 18, Station  2, Class 3, Type 2, DISCARDED 0
```

```
Report by CP: MVCI 19, Station  3, Class 2, Type 2, DISCARDED 0
Report by CP: MVCI 20, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 21, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 22, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 23, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 24, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 25, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 26, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 27, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 28, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 29, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 30, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 31, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 32, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 33, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 34, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 35, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 36, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 37, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 38, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 39, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 40, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 41, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 42, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 43, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 44, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 45, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 46, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 47, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 48, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 49, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 50, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 51, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 52, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 53, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 54, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 55, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 56, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 57, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 58, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 59, Station  3, Class 3, Type 2, DISCARDED 0
Report by CP: MVCI 60, Station  4, Class 3, Type 3, DISCARDED 0
Report by Remote  1: MVCI 16, ARRIVED 112895, DISCARDED 106
Report by Remote  2: MVCI 17, ARRIVED 272312, DISCARDED 0
Report by Remote  2: MVCI 18, ARRIVED 6491,   DISCARDED 0
Report by Remote  3: MVCI 19, ARRIVED 272743, DISCARDED 0
Report by Remote  3: MVCI 20, ARRIVED 4391,   DISCARDED 0
Report by Remote  3: MVCI 21, ARRIVED 3531,   DISCARDED 0
Report by Remote  3: MVCI 22, ARRIVED 4940,   DISCARDED 0
Report by Remote  3: MVCI 23, ARRIVED 4516,   DISCARDED 0
Report by Remote  3: MVCI 24, ARRIVED 5346,   DISCARDED 0
Report by Remote  3: MVCI 25, ARRIVED 5894,   DISCARDED 0
Report by Remote  3: MVCI 26, ARRIVED 3569,   DISCARDED 0
Report by Remote  3: MVCI 27, ARRIVED 5360,   DISCARDED 0
Report by Remote  3: MVCI 28, ARRIVED 8512,   DISCARDED 0
Report by Remote  3: MVCI 29, ARRIVED 4038,   DISCARDED 0
Report by Remote  3: MVCI 30, ARRIVED 10233,  DISCARDED 0
Report by Remote  3: MVCI 31, ARRIVED 5256,   DISCARDED 0
Report by Remote  3: MVCI 32, ARRIVED 4907,   DISCARDED 0
Report by Remote  3: MVCI 33, ARRIVED 4967,   DISCARDED 0
```

```
Report by Remote  3: MVCI 34, ARRIVED 9820,   DISCARDED 0
Report by Remote  3: MVCI 35, ARRIVED 4859,   DISCARDED 0
Report by Remote  3: MVCI 36, ARRIVED 4422,   DISCARDED 0
Report by Remote  3: MVCI 37, ARRIVED 3769,   DISCARDED 0
Report by Remote  3: MVCI 38, ARRIVED 4240,   DISCARDED 0
Report by Remote  3: MVCI 39, ARRIVED 4135,   DISCARDED 0
Report by Remote  3: MVCI 40, ARRIVED 3737,   DISCARDED 0
Report by Remote  3: MVCI 41, ARRIVED 7186,   DISCARDED 0
Report by Remote  3: MVCI 42, ARRIVED 6254,   DISCARDED 0
Report by Remote  3: MVCI 43, ARRIVED 5344,   DISCARDED 0
Report by Remote  3: MVCI 44, ARRIVED 5616,   DISCARDED 0
Report by Remote  3: MVCI 45, ARRIVED 5105,   DISCARDED 0
Report by Remote  3: MVCI 46, ARRIVED 4575,   DISCARDED 0
Report by Remote  3: MVCI 47, ARRIVED 5988,   DISCARDED 0
Report by Remote  3: MVCI 48, ARRIVED 13395,  DISCARDED 0
Report by Remote  3: MVCI 49, ARRIVED 7224,   DISCARDED 0
Report by Remote  3: MVCI 50, ARRIVED 3891,   DISCARDED 0
Report by Remote  3: MVCI 51, ARRIVED 7405,   DISCARDED 0
Report by Remote  3: MVCI 52, ARRIVED 8394,   DISCARDED 0
Report by Remote  3: MVCI 53, ARRIVED 3969,   DISCARDED 0
Report by Remote  3: MVCI 54, ARRIVED 3801,   DISCARDED 0
Report by Remote  3: MVCI 55, ARRIVED 7876,   DISCARDED 0
Report by Remote  3: MVCI 56, ARRIVED 5100,   DISCARDED 0
Report by Remote  3: MVCI 57, ARRIVED 7514,   DISCARDED 0
Report by Remote  3: MVCI 58, ARRIVED 4561,   DISCARDED 0
Report by Remote  3: MVCI 59, ARRIVED 12345,  DISCARDED 0
Report by Remote  4: MVCI 60, ARRIVED 4846,   DISCARDED 0

Control Cells:
--------------
Available 555859, Successful 60372, Collided messages 5022, Collided_slots 2493

Information Cells:
------------------
Total downlink cells in channel: 437107
Total uplink cells in channel: 905161

Channel Utilization: 0.408049
```

LIST OF REFERENCES

[1] N. Abramson, "The Throughput of Packet Broadcasting Channels," *IEEE Trans. on Comm.*, vol. 25, no. 1, pp. 117-128, Jan. 1977.

[2] A. Acampora, "Wireless ATM: A Perspective on Issues and Prospects," *IEEE Personal Comm.*, pp. 8-17, Aug. 1996.

[3] P. Agrawal, P. P. Mishra, and M. Srivastava, "Network, Architecture for Mobile and Wireless ATM," in *Proc. ICDCS*, 1996, pp. 299-310.

[4] J. B. Anderson, T. S. Rappaport, and S. Yoshida, "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Comm. Mag.*, pp. 42-49, Jan. 1995.

[5] D. Anick, D. Mitra, and M. Sondhi, "Stochastic Theory of a Data Handling System with Multiple Sources," *Bell Sys. Tech. J.*, vol. 61, no. 8, pp. 1871-1894, Oct. 1982.

[6] C. Apostolas, R. Tafazolli, and B. G. Evans, "Wireless ATM LAN," in *Proc. PIMRC*, 1995, pp. 773-777.

[7] U.S. Army, Program Executive Office Command, Control and Communications Systems (PEO C3S), Project Manager Tactical Radio Communications Systems (PM TCRS), Near Term Digital Radio http://www.monmouth.army.mil/peoc3s/trcs/trcs.htm (see also http://134.80.43.185/ntdrnf.html).

[8] The ATM Forum, *User-Network Interface Specification Ver. 3.1*, Upper Saddle River, NJ: Prentice-Hall, 1995.

[9] The ATM Forum Technical Committee, *Traffic Management Specification Ver. 4.0*, af-tm-0056.000, Apr. 1996.

[10] E. Ayanoglu, K. Y. Eng, and M. J. Karol, "Wireless ATM: Limits, Challenges, and Proposals," *IEEE Personal Comm.*, pp. 18-33, Aug. 1996.

[11] M. Barton and T. R. Hsing, "Architecture for Wireless ATM Networks," in *Proc. PIMRC*, 1995, pp. 778-782.

[12] D. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed., Upper Saddle River, NJ: Prentice-Hall, 1992.

[13] U. Black, *ATM: Foundation for Broadband Networks*, Englewood Cliffs, NJ: Prentice-Hall, 1995.

[14] U. Black, *ATM Volume II: Signaling in Broadband Networks*, Upper Saddle River, NJ: Prentice-Hall, 1998.

[15] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, 2nd ed., San Francisco, CA: Holden Day, 1976.

[16] P. T. Brady, "A Model for Generating On-Off Speech Patterns in Two-Way Conversation," *Bell Sys. Tech. J.*, pp. 2445-2472, Sep. 1969.

[17] K. –C. Chen and C. –H. Lee, "RAP - A Novel Medium Access Control Protocol for Wireless Data Networks," in *Proc. GLOBECOM*, 1993, pp. 1713-1717.

[18] R. Chipalkatti, J. F. Kurose, and D. Towsley, "Scheduling Policies for Real-Time and Non-Real-Time Traffic in a Statistical Multiplexer," *in Proc. INFOCOM*, 1989, pp. 774-783.

[19] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, Cambridge, MA: The MIT Press, 1990.

[20] J. N. Daigle and J. D. Langford, "Models for Analysis of Packet Voice Communications Systems," *IEEE J. of Select. Areas in Comm.*, vol. 4, no. 6, pp. 847-855, Sep. 1986.

[21] S. Dastangoo, "A Multimedia Medium Access Control Protocol for ATM Based Mobile Networks," in *Proc. PIMRC*, 1995, pp. 794-798.

[22] R. L. Davies, R. M. Watson, A. Munro, and M. H. Barton, "Ad-Hoc Wireless Networking: Contention Free Multiple Access," in *Proc. Of the 5th IEE Conf. on Telecomm.*, 1995, pp. 73-77.

[23] M. Decina, T. Toniatti, P. Vaccari, and L. Verri, "Bandwidth Assignment and Virtual Call Blocking in ATM Networks," in *Proc. INFOCOM*, 1990, pp. 881-888.

[24] M. Decina and T. Toniatti, "On Bandwidth Allocation to Bursty Virtual Connections in ATM Networks," in *Proc. ICC*, 1990, pp. 844-851.

[25] M. De Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, 3rd ed., Prentice-Hall International (UK), 1995.

[26] U. Dersch and W. R. Braun, "A Physical Mobile Radio Channel Model," in *Proc. Veh. Tech. Conf.*, 1991, pp. 289-294.

[27] A. I. Elwalid and D. Mitra, "Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, pp. 329-343, June 1993.

[28] Z. Fan and P. Mars "Accurate Approximation of Cell Loss Probability for Self-Similar Traffic in ATM Networks," *Electronic Letters*, vol. 32, no. 19, pp. 1749-1751, Sep. 1996.

[29] M. Fontaine and D. G. Smith, "Bandwidth Allocation and Connection Admission Control in ATM Networks," *Elec. & Comm. Eng. J.*, pp. 156-164, Aug. 1996.

[30] V. K. Garg and J. E. Wilkes, *Wireless and Personal Communications Systems*, Upper Saddle River, NJ: Prentice-Hall, 1996.

[31] S. J. Golestani, "A Framing Strategy for Congestion Management," *IEEE J. of Select. Areas in Comm.*, vol. 9, no. 7, pp. 1064-1077, Sep. 1991.

[32] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet Reservation Multiple Access," *IEEE Trans. on Comm.*, vol. 37, no. 8, pp. 885-890, Aug. 1989.

[33] D. J. Goodman and S. X. Wei, "Efficiency of Packet Reservation Multiple Access," *IEEE Trans. on Veh. Tech.*, vol. 40, no. 1, pp. 170-176, Feb. 1991.

[34] J. Gordon, "Pareto Process as a Model of Self-Similar Packet Traffic," in *Proc. GLOBECOM*, 1995, pp. 2232-2236.

[35] C. Graff, F. Halloran, and C. Lockhart, "Tactical Battlefield ATM," in *Proc. MILCOM*, 1994, pp. 473-478.

[36] H. Heffes and D. M. Lucantoni, "A Markov Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," *IEEE J. of Select. Areas in Comm.*, vol. 4, no. 6, pp. 856-868, Sep. 1986.

[37] S. –S Huang., "Modeling and Analysis for Packet Video," in *Proc. GLOBECOM*, 1989, pp. 881-885.

[38] J. M. Hyman, A. A. Lazar, and G. Pacifici, "Real-Time Scheduling with Quality of Service Constraints," *IEEE J. of Select. Areas in Comm.*, vol. 9, no. 7, pp. 1052-1063, Sep. 1991.

[39] International Organization for Standardization, *Basic Reference Model for Open Systems Interconnection*, ISO 7498, 1984.

[40] ITU-T (formerly CCITT) Recommendation I.121, *Broadband Aspects of ISDN*, 1992.

[41] ITU Study Group 11, Q.2931, Clause 1, 2, 3, ed. Shiraishi S., Yao H., June 1994.

[42] J. R. Jackson, "Scheduling a Production Line to Minimize Maximum Tardiness," Research Report 43, Management Science Report, University of California, Los Angeles, 1955.

[43] K. Joseph, D. Raychaudhuri, and J. Zdepski, "Shared Access Packet Transmission Systems for Compressed Digital Video," *IEEE J. of Select. Areas in Comm.*, vol. 7, no. 5, pp. 815-825, June 1989.

[45] R. E. Kahn, "Advances in Packet Radio Technology," *Proc. IEEE*, vol. 66, no. 11, pp. 1468-1496, Nov. 1978.

[46] M. J. Karol, Z. Liu, and K. Y. Eng, "Distributed-Queueing Request Update Multiple Access (DQRUMA) for Wireless Packet (ATM) Networks," in *Proc. ICC*, 1995, pp. 1224-1231.

[47] G. Kesidis, *ATM Network Performance*, Kluwer Academic Publishers, 1996.

[48] P. C. Kiessler, C. J. Wypasek, R. E. Fennell, and J. M. Westall, "Markov Renewal Models for Traffic Exhibiting Self-Similar Behavior," in *Proc. IEEE SOUTHEASTCON*, 1996, pp. 76-79.

[49] L. Kleinrock, *Queueing Systems, Volume I: Theory*, Wiley-Interscience, 1975.

[50] L. Kleinrock, "Principles and Lessons in Packet Communications," in *Proc. of the IEEE*, vol. 66, no. 11, pp. 1320-1329, Nov. 1978.

[51] T. Kurner, D. J. Cichon, and W. Wiesbeck, "Evaluation and Verification of the VHF/UHF Propagation Channel Based on a 3-D-Wave Propagation Model," *IEEE Trans. on Antennas and Propagation*, vol. 44, no. 3, pp. 393-404, Mar. 1996.

[52] S. S. Lam, "Packet Broadcast Networks - A Performance Analysis of the R-ALOHA Protocol," *IEEE Trans. on Comp.*, vol. 29, no. 7, pp. 596-603, July 1980.

[53] R. O. LaMaire, A. Krishna, and H. Ahmadi, "Analysis of a Wireless MAC Protocol with Client-Server Traffic and Capture," *IEEE J. of Select. Areas in Comm.*, vol. 12, no. 8, pp. 1299-1313, Oct. 1994.

[54] A. A. Lazar and G. Pacifici, "Control of Resources in Broadband Networks with Quality of Service Guarantees," *IEEE Comm. Mag.*, vol. 29, no. 10, pp. 66-73, Oct. 1991.

[55] B. M. Leiner, D. L. Nielson, and F. A. Tobagi, "Issues in Packet Radio Network Design," in *Proc. of the IEEE*, vol. 75, no. 1, pp. 6-20, Jan. 1987.

[56] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Trans. on Networking*, vol. 2, no. 1, pp. 1-15, Feb. 1994.

[57] H. Levy and M. Sidi, "Polling Systems: Applications, Modeling, and Optimization," *IEEE Trans. on Comm.*, vol. 38, no. 10, pp. 1750-1760, Oct. 1990.

[58] N. Likhanov, B. Tsybakov, and N. D. Georganas, "Analysis of an ATM Buffer with Self-Similar ("Fractal") Input Traffic," in *Proc. INFOCOM*, 1995, pp. 985-992.

[59] T. –L. Ling and N. Shroff, "Scheduling Real-Time Traffic in ATM Networks," in *Proc. INFOCOM*, 1996, pp. 198-205.

[60] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance Models of Statistical Multiplexing in Packet Video Communications," *IEEE Trans. on Comm.*, vol. 36, no. 7, pp. 834-844, July 1988.

[61] N. Movahhedinia, G. Stamatelos, and H. M. Hafez, "A Slot Assignment Protocol for Indoor Wireless ATM Networks using the Channel Characteristics and the Traffic Parameters," in *Proc. GLOBECOM*, 1995, pp. 327-331.

[62] N. Movahhedinia, G. Stamatelos, and H. M. Hafez, "Polling Based Multiple Access for Indoor Broadband Wireless Systems," in *Proc. PIMRC*, 1995, pp. 1052-1056.

[63] S. Nanda, D. J. Goodman, and U. Timor, "Performance of PRMA: A Packet Protocol for Cellular Systems," *IEEE Trans. on Veh. Tech.*, vol. 40, no. 3, pp. 584-598, Aug. 1991.

[64] R. O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues*, 2nd ed., Norwood, MA: Artech-House, 1995.

[65] A. Oppenhaim and A. Willsky, *Signals and Systems*, Prentice-Hall, 1983.

[66] S. S. Panwar, D. Towsley, and J. K. Wolf, "Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service," *J. of the ACM*, vol. 35, no. 4, pp. 832-844, Oct. 1988.

[67] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, pp. 344-357, June 1993.

[68] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Trans. on Networking*, vol. 3, no. 3, pp. 226-244, June 1995.

[69] J. M. Peha and F. A. Tobagi, "Evaluating Scheduling Algorithms for Traffic with Heterogeneous Performance Objectives," in *Proc. GLOBECOM*, 1990, pp. 21-27.

[70] J. M. Peha and F. A. Tobagi, "A Cost-Based Scheduling Algorithm to Support Integrated Services," in *Proc. INFOCOM*, 1991, pp. 741-753.

[71] J. M. Peha and F. A. Tobagi, "Cost-Based Scheduling and Dropping Algorithms to Support Integrated Services," *IEEE Trans. on Comm.*, vol. 44, no. 2, pp. 192-202, Feb. 1996.

[72] R. Pichna and Q. Wang, "A MAC Protocol for the Integrated Wireless Access Networks," in *Proc. PIMRC*, 1995, pp. 248-252.

[73] T. T. Piper and D. L. Hagen, "ATM-Providing Broadband Services to the Warrior," in *Proc. of the Tactical Comm. Conf.*, 1994, pp. 217-224.

[74] G. Ramamurthy and B. Sengupta, "Modeling and Analysis of a Variable Bit Rate Video Multiplexer," in *Proc. INFOCOM*, 1992, pp. 817-827.

[75] D. Raychaudhuri and N. Wilson, "Multimedia Transport in Next-Generation Personal Communication Networks," in *Proc. ICC*, 1993, pp. 858-862.

[76] D. Raychaudhuri and N. D. Wilson, "ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks," *IEEE J. of Select. Areas in Comm.*, vol. 12, no. 8, pp. 1401-1414, Oct. 1994.

[77] D. Raychaudhuri, "Wireless ATM Networks: Architecture, System Design and Prototyping," *IEEE Personal Comm.*, pp. 42-49, Aug. 1996.

[78] D. Raychaudhuri *et al.*, "WATMnet: A Prototype Wireless ATM System for Multimedia Personal Communication," *IEEE J. of Select. Areas in Comm.*, vol. 15, no. 1, pp. 83-95, Jan. 1997.

[79] M. Schwartz, "Network Management and Control Issues in Multimedia Wireless Networks," *IEEE Personal Comm.*, vol. 2, no. 3, pp. 8-16, June 1995.

[80] M. Schwartz, *Broadband Integrated Networks*, Upper Saddle River, NJ: Prentice-Hall, 1996.

[81] M. Schwartz, Columbia University, *personal communication*, 1997.

[82] P. Sen, B. Maglaris, N. –E. Rikli, and D. Anastassiou, "Models for Packet Switching of Variable-Bit-Rate Video Sources," *IEEE J. of Select. Areas in Comm.*, vol. 7, no. 5, pp. 865-869, June 1989.

[83] N. B. Shroff, "Traffic Modeling and Analysis in High Speed ATM Networks," Ph.D. Thesis, Graduate School of Arts and Sciences, Columbia University, 1995.

[84] P. Skelly, M. Schwartz, and M. Dixit, "A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 446-459, Aug. 1993.

[85] K. Sriram, "Dynamic Bandwidth Allocation and Congestion Control Schemes for Voice and Data Multiplexing in Wideband Packet Technology," in *Proc. ICC*, 1990, pp. 1003-1009.

[86] W. Stallings, *Data and Computer Communications*, 4th ed., Upper Saddle River, NJ: Prentice-Hall, 1994.

[87] Z. S. Su and K. C. Sevick, "A Combinatorical Approach to Scheduling Problems," Operational Research, vol. 26, no. 5, pp. 836-844, Sep.-Oct. 1978.

[88] S. N. Subramanian and T. Le-Ngoc, "Traffic Modeling in a Multimedia Environment," in *Proc. of the Canadian Conf. on Elect. and Comp. Eng.*, 1995, pp. 838-841.

[89] C. –K. Toh, *Wireless ATM and Ad-Hoc Networks*, Kluwer Academic Publishers, 1997.

[90] A. Uziel and M. Tummala, "Modeling of Low Data Rate Services for Mobile ATM," in *Proc. PIMRC*, 1997, pp. 194-198.

[91] A. Uziel and M. Tummala, "Protocol Architecture for Tactical Integrated Services Mobile Networks," in *Proc. MILCOM*, 1997, pp. 2532-2536.

[92] R. Van Engelshoven, "ATM for Military Communications," in *Proc. MILCOM*, 1995, pp. 217-223.

[93] M. Veeraraghavan, T. F. La Porta, and R. Ramjee, "A Distributed Control Strategy for Wireless ATM Networks," *ACM Wireless Networks J. (WINET)*, vol. 1, no. 3, pp. 323-339, Sep. 1995.

[94] D. Veitch, "Novel Models of Broadband Traffic," in *Proc. GLOBECOM*, 1993, pp. 1057-1061.

[95] W. Verbiest and L. Pinnoo, "A Variable Bit Rate Video Codec for Asynchronous Transfer Mode Networks," *IEEE J. of Select. Areas in Comm.*, vol. 7, no. 5, pp. 761-770, June 1989.

[96] J. H. Wen and J. W. Wang, "A New Protocol for Voice Communications - Non-Collision Packet Reservation Multiple Access," in *Proc. PIMRC*, 1994, pp. 638-642.

[97] D. B. West, *Introduction to Graph Theory*, Upper Saddle River, NJ: Prentice-Hall, 1996.

[98] T. J. Wheeler *et al.* "Designing a Tactical ATM Network Integrating Performance Engineering and Design," in *Proc. MILCOM*, 1994, pp. 215-219.

[99] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[100] H. Xie, R. Yuan, and D. Raychaudhuri, "Data Link Control Protocols for Wireless ATM Access Channels," in *Proc. ICUPC*, 1995, pp. 753-757.

[101] R. Yuan, K. Biswas, and D. Raychaudhuri, "A Signaling and Control Architecture for Mobility Support in Wireless ATM Networks," in *Proc. ICC*, 1996, pp. 478-484.

[102] W. Zhuang, "Medium Access Control Protocol for Multimedia Wireless Networks," in *Proc. GLOBECOM*, 1995, pp. 1094-1098.

INITIAL DISTRIBUTION LIST

1. Defense technical Information Center 2
   8725 John J. Kingman Rd., Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Dean of Research, Code 09 1
   Naval Postgraduate School
   Monterey, CA 93943-5138

4. Chairman, Department of Electrical and Computer Engineering, Code EC 1
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Prof. Murali Tummala, Code EC/Tu 4
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Prof. Herschel H. Loomis, Jr., Code EC/Lm 1
   Naval Postgraduate School
   Monterey, CA 93943-5121

7. Prof. Gus K. Lott, Jr., Code EC/Lt 1
   Naval Postgraduate School
   Monterey, CA 93943-5121

8. Prof. Gilbert M. Lundy, Code CS/Ln 1
   Naval Postgraduate School
   Monterey, CA 93943-5121

9. Prof. Craig Rasmussen, Code MA/Ra 1
   Naval Postgraduate School
   Monterey, CA 93943-5121

10. Prof. John E. McEachen, Code EC/Mj 1
Naval Postgraduate School
Monterey, CA 93943-5121

11. Dr. Don Gingras 1
SPAWAR Systems Center San-Diego, Code D8805
Communication and Information Systems Department
San-Diego, CA 92152-5001

12. Commander of Signal, Electronics, and Computers Corps 1
Signal, Electronics, and Computers Corps Headquarters
Military P. O. Box 02150
Israeli Defense Forces
Israel

13. Head of Signal Systems Department 1
Signal, Electronics, and Computers Corps Headquarters
Military P. O. Box 02150
Israeli Defense Forces
Israel

14. Head of Signal Department 1
Ground Forces Headquarters
Military P. O. Box 02243
Israeli Defense Forces
Israel

15. LCDR Robert E. Parker, U.S. Navy 1
SGC 1165
Naval Postgraduate School
Monterey, CA 93943-1165

16. Amir Uziel 3
90 Herzel St.
Bat-Yam 59471
Israel