A Burst-Oriented Traffic Control Framework and Associated Call Admission Control Schemes for ATM Networks

By

Jose Roberto Santiano Fernandez

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Computer Science Department

1995

Abstract

A Burst-Oriented Traffic Control Framework and Associated Call Admission Control Schemes for ATM Networks

By

Jose Roberto Santiano Fernandez

Asynchronous transfer mode (ATM) networks are intended to support integrated digital communications traffic. This means that they are expected to handle different traffic generation characteristics ranging from smooth continuous bit rate transmission to highly irregular bursty data generation. This dissertation proposes a traffic control approach that is tailored for the efficient handling of highly bursty traffic in which related bursts of cells may be treated as units. The approach is based on a connection-oriented transmission mode that allows "on-the-fly" allocation of bandwidth in the switches.

The proposal includes a framework of ATM switch mechanisms that allow a bursty connection to acquire resources at a switch when a burst arrives and to release them when a burst leaves. The framework supports bursty connections that are silent between their bursty periods as well as connections that may transmit at a non-zero rate between bursts. Furthermore, it allows the bundling of these bursty connections into virtual paths.

The task of controlling congestion in ATM networks primarily rests on a call admission control (CAC) scheme that limits the number of established connections. In this dissertation, three different CAC schemes that are geared towards maintaining the burst blocking probabilities (BBPs) of connections are presented. The first scheme is a highly computationally efficient approach based on the Erlang loss system model. This is compared with a more general model (already proposed by other researchers)

that requires more computation. The second scheme is an extension of this general model to handle burst-level (rather than cell-level) priorities. The third scheme is another extension of the general model whose purpose is to take into account quantifiable correlations between related connections. This approach is useful for supporting distributed parallel computation over large ATM networks. The CAC schemes have been evaluated using analytical and simulation techniques. The results show that the proposals contribute to utilizing bandwidth more efficiently while maintaining the BBPs of admitted connections.

To my Creator.

Acknowledgments

First, I thank my research adviser, Prof. Matt W. Mutka, for his guidance and direction in this work.

I also thank the following for supporting me in one way or another during my years at MSU: Pepito, Cielo, Bolet, Billy, Joji, Katrina, my Guardian Angel, and the Department of Computer Science.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The future of telecommunications is in high-speed integrated networks. The most promising candidate for the technology that will facilitate the integration of data, telephony, and cable television networks is the *asynchronous transfer mode* (ATM) network [11]. In this network, all data is transported in streams of small protocol units called *cells* that are relayed at high speed by special hardware called *ATM switches* [66].

ATM networks are designed to carry all types of communication traffic including *continuous bit rate* (CBR), *variable bit rate* (VBR), and "best-effort"[1] traffic. The main advantage of the ATM approach over circuit-switched networks or traditional packet-switched networks lies in its ability to support bursty connections with a guaranteed quality of service. The statistical multiplexing of bursty connections allows a higher bandwidth utilization than that which the circuit-switching approach could achieve with comparable bursty traffic. In addition, with proper network design and management, a quality of service in terms of low loss rate and/or bounded delay can be guaranteed to ATM network connections.

It is expected that in the future, a large portion of integrated network traffic will

[1] Alternatively, "*available bit rate*" (ABR) is a new term for non-guaranteed service.

be generated by bursty sources. Examples of bursty applications that are already beginning to be more widely used are resource discovery tools [53] such as the *World-Wide Web* and *Gopher*. In addition, multimedia applications [4, 67] including video teleconferencing and multimedia database retrieval will be more commonplace. Finally, basic electronic communications services including electronic mail and network news (i.e., USENET) will be practically universal.

In order for ATM networks to successfully support a wide range of bursty applications, a new traffic control strategy must be developed. The new approach has to be tailored for highly bursty connections while still efficiently supporting the less bursty or well-behaved connections. In particular, it has to tolerate difficult connections with extra long bursts or prolonged quiescent periods. Furthermore, it has to achieve an acceptable bandwidth utilization while still guaranteeing the required level of service for the bursty as well as the non-bursty connections. Finally it is desirable that the scheme not require potentially expensive hardware such as large buffers in all the switching nodes.

To this end, a comprehensive framework for burst-level handling of cell traffic has been developed. The distinguishing characteristic of this proposal is that bursts of cells that are logically related are handled as a unit rather than individually as independent entities. While this general approach has already been suggested by Hui [35], the proposed framework builds on the idea, fills in some of the important details, addresses some important related issues, and provides features such as burst-level priorities and "correlation awareness."

The highlights of the current proposal include the following. It supports "burst-oriented" connections with "on-the-fly" bandwidth allocation and eager transmission of bursts, where "eager" means that the bursts are transmitted before bandwidth has been committed and acknowledged for them. Furthermore, it supports the bundling of burst-oriented connections into virtual paths (VPs). This feature is not covered in

any of the other published proposals of a similar nature. It supports two varieties of bursty connections, including one with a zero rate between bursts and another that allows the connections to have a non-zero nominal rate between bursts.

The proposal also includes three different but related call admission control schemes that guarantee limits on the loss rate of bursts. The first admission scheme proposed is a computationally efficient approach based on the *Erlang loss system* model. Its main attraction is that the aggregate load of a set of sources can be obtained additively from the individual loads of the sources. The bandwidth requirement of the set can then be obtained from the aggregate load using a simple recursion or a table lookup. This approach is compared with a more general model that does not assume any burst arrival process, but just assumes that the sources are independent. The second proposed scheme extends the aforementioned general model (that has been investigated by other researchers) by adding burst-level priorities to it. To be more specific, it allows two sets of sources to experience different levels of burst loss rates while sharing the same bandwidth resources. The third scheme also uses the same general model but allows subsets of the sources to be correlated. In this scheme, the correlated sources are grouped into independent *applications* that are to be modeled separately. The primary purpose of this scheme is to properly consider the strong correlation that may exist among traffic sources participating in distributed parallel computation over an ATM network.

The rest of the dissertation is organized as follows. An overview of high-speed network standards and technologies is given in Chapter 2. Chapter 3 gives a survey of related research results in the area of high-speed network communication, with emphasis on traffic control schemes. Chapter 4 presents the proposed framework of protocols and mechanisms designed to facilitate the burst-level handling of traffic. The call admission control scheme based on Erlang's loss formula is described and evaluated in Chapter 5. Chapter 6 presents and evaluates the proposed burst-level

priority scheme. The "correlation-aware" admission control scheme is given and evaluated in Chapter 7. Finally, Chapter 8 concludes this dissertation with a summary of the work accomplished and possible directions for further research in the burst-oriented approach to ATM traffic control.

Chapter 2

Background: High-Speed Network Technologies

The information superhighway of tomorrow will be the result of the interconnection of the telephone, cable television, and data networks. In the past, telephone networks primarily carried voice conversations, cable TV networks distributed one-way TV signals in guided media, and data networks carried computer data. However, nowadays it is not uncommon for telephone lines intended for voice to be used for the transmission of digitized information using fax machines or modems. Similarly, computer networks such as the *Internet* and nationwide or international on-line services (e.g., *CompuServe*, *Prodigy*, and *America Online*) are being utilized for the collection and dissemination of all kinds of information including text, still or moving-image pictures, and audio recordings. It is therefore apparent that there is a demand for a network infrastructure that is capable of carrying all sorts of information and providing all kinds of services. Note that some of the desirable services from such a network require continuous transmission (e.g., live audio or video) and that some of the services require a large available bandwidth (full-motion video). For this reason, new network technologies are being developed that satisfy both requirements.

In summary, here are some of the desirable properties of the network infrastructure for the information superhighway of the future:

- high-speed (e.g., sufficient for full-motion video)
- support for different traffic requirements (e.g., connection-oriented, connectionless, smooth, bursty), and
- scalability over greater data rates and distances.

Section 2.1 is an overview of competing network technologies intended for limited areas of coverage. Section 2.2 discusses some of the high-speed network services that are available or are being planned. Section 2.3 describes the ATM network, which is a viable all-purpose high-speed network with an unlimited area of coverage.

2.1 High-Speed Network Standards for Limited Areas

In this section, some of the more popular standards for high-speed networks are described. These networks offer high-speed networking solutions only for limited areas (i.e., local and metropolitan areas), but are worthwhile to mention in this work because they offer competing solutions for their intended areas of coverage. In addition, they share some characteristics with ATM networks:

- HIPPI uses interconnection fabrics such as *crossbar switches* to interconnect multiple links.
- Both FDDI-II and DQDB support connection-oriented, connectionless, and *isochronous* (i.e., periodic or circuit-switched) traffic.

- DQDB cells are similar to ATM cells. Both DQDB and ATM networks can support SMDS (described in Section 2.2.3 below).

2.1.1 HIPPI

The High-Performance Parallel Interface (HIPPI) [80] is a point-to-point interface for transferring data at high-speeds over short distances. The standard was developed by the ANSI Task Group X3T9.3 based on a proposal from the Los Alamos National Laboratory. The primary application of HIPPI is the interconnection of supercomputers and associated equipment such as visualization terminals and high-performance storage systems.

The basic circuit of HIPPI is a simplex point-to-point link, a collection of which can be organized into a more sophisticated interconnection system such as a crossbar switch. The physical layer of HIPPI is a set of copper twisted-pair cables capable of parallel 32-bit or 64-bit data transmission at a total rate of 800 or 1600 Mbps and at a distance of up to 25 meters (about 82 feet). Not part of the standard, but rather an "implementor's agreement," is a proposal for a serial version of HIPPI based on fiber for distances of up to 10 km (about 6 miles).

2.1.2 FDDI

Fiber Distributed Data Interface (FDDI) [1, 42, 73, 74, 79, 84] is a set of standards for a high-speed local or metropolitan area network. It was developed by the ANSI X3T9.5 Task Group. Its main features include the following.

- the use of fiber links at the rate of 100 Mbps
- a dual-ring topology for fault tolerance

- a medium access control (MAC) protocol based on the IEEE 802.5 token-ring standard [42, 73, 74, 79, 84] but taking advantage of the token rotation time
- dynamic bandwidth allocation with synchronous and asynchronous transmission
- support of up to 1000 stations over distances of up to 100 km[1] (about 60 miles)
- frame sizes of up to 4500 octets

FDDI was originally intended for data traffic and its synchronous transmission service suffers from having variable delay (even though it is bounded). FDDI-II is an upward-compatible extension to FDDI that supports isochronous service in addition to the synchronous and asynchronous packet-switched service.

The FDDI standard has evolved to the point that its original name has become practically a misnomer. The latest standard allows for any transmission medium (including optical fiber and twisted-pair wire), distributed or centralized control, any traffic (including data, voice, and video), and may refer to an interface, LAN, or MAN [37].

2.1.3 DQDB

Distributed Queue Dual Bus (DQDB) [1, 42, 74, 84] is the standard proposed by the IEEE 802.6 committee for metropolitan area networks. It is based on the Queued Packet and Synchronous Circuit Exchange (QPSX) protocol developed at the University of Western Australia. Its main features include the following.

- a dual unidirectional bus topology
- use of a variety of media including optical fiber and coaxial cable, at data rates ranging from 34 to 155 Mbps and higher

[1] More precisely, the standard specifies a maximum *fiber length* of 200 km.

- support of connection-oriented and connectionless packet-switched data services and isochronous service
- support of over 500 nodes on a 160 km (about 100 miles) dual bus network
- use of 53-octet slots (48 octets of which are the data payload) within the large frames of the underlying transmission system

In addition, the standard includes the option of using a looped dual bus topology for fault tolerance.

In order to support both packet-switched and circuit-switched services, DQDB uses two bus access control methods. The queue arbitrated (QA) access method is based on a distributed queue concept where each node keeps track of its position in the global distributed queue of each bus. This method is intended for data services that are not delay-critical. The pre-arbitrated (PA) access method is intended for isochronous applications such as voice transmission. It is based on reserving a portion of PA slots that are generated periodically by the head nodes of each bus.

Research on performance aspects, fairness, and enhancements of DQDB are continuously worked on. Sadiku and Arvind have compiled an annotated bibliography in [69].

2.2 Services Offered by High-Speed Networks

Traditional X.25 [72, 73, 75, 79] packet-switching service and the plain old telephone service (POTS) comprise the bulk of communication services today. These are low-speed services with X.25 limited to 64 Kbps and POTS limited to 4 KHz analog signals (64 Kbps digital). ISDN, frame-relay, SMDS, and BISDN are high-speed services being offered or being planned for high-speed networks. ISDN and BISDN, in addition, provide integrated circuit-switched and packet-switched services.

2.2.1 ISDN

The Integrated Services Digital Network (ISDN) [42, 72, 73, 75, 79, 84] is a small step toward replacing the POTS with a multipurpose communication facility capable of providing a variety of services such as voice, data transfer, fax, slow-scan video, and interactive terminal services.

ISDN services are provided to customers at two levels. *Basic rate interface* (BRI) access is intended or residential customers and consists of two full-duplex 64 Kbps B-channels (B for *bearer*) and one full-duplex 16 Kbps D-channel (D for *data*) on a 192 Kbps link. Telephone, fax, slow-scan TV, and packet-switched data services are provided on the B-channels. The D-channel is primarily intended for signaling (e.g., call setup) but can also be used for telemetry and low-rate packet-switched data. *Primary rate interface* (PRI) access, which is intended for larger customers such as businesses, provides 23[2] B-channels and a 64 Kbps D-channel totaling 1.544 Mbps, which is the data rate of the so-called T1 carrier service. The PRI may also be reconfigured to carry higher bit rate channels called H-channels up to the capacity of the underlying carrier.

ISDN, like most modern telephone systems, uses *common channel signaling* (CCS) [72, 73, 75, 79] in which all signaling information is transmitted over a separate network, typically an X.25 packet-switching network. However, in addition to having a circuit-switched network for supporting synchronous services like telephone service, it also has a packet-switched network for packet-switched data service separate from the CCS network.

[2]30 for countries using the E1 rather than the T1 carrier link

2.2.2 Frame-Relay

Frame-relay [11, 42, 73, 75] is a connection-oriented fast packet-switching service intended for intermediate speeds from about 64 Kbps to 1.544 Mbps. Traditional packet switching based on X.25 has a large *per-packet* and *per-node* processing overhead that makes it unsuitable for rates higher than about 64 Kbps. Frame-relay reduces this overhead by allowing variable-sized frames to be forwarded as soon as possible, even before the whole frame is received at a node. This is practical in modern transmission facilities that have very low bit error rates and high data rates, since only a small fraction of the bandwidth is wasted by forwarding frames that may be corrupt.

2.2.3 SMDS

Switched Multimegabit Data Service (SMDS) [11, 42] is a connectionless packet-switching service specification appropriate for data rates higher than those provided by frame-relay. It is intended to provide features usually found in local area networks, such as high throughput and low delay, but for large metropolitan areas.

Access to SMDS is provided through a three-layer SMDS Interface Protocol (SIP) in which data packets of up to 9188 octets are broken up to be transmitted in 53-octet cells appropriate for DQDB, the underlying transport subnetwork. SMDS does not provide isochronous service.

2.2.4 BISDN

Broadband ISDN (BISDN) [42, 72, 73, 75, 84] is the higher bandwidth version of ISDN that supports channels with rates higher than the ISDN primary rate. These rates ranging from under 50 Mbps to over 150 Mbps are capable of supporting full-motion video services of varying quality. These would make available broadband services

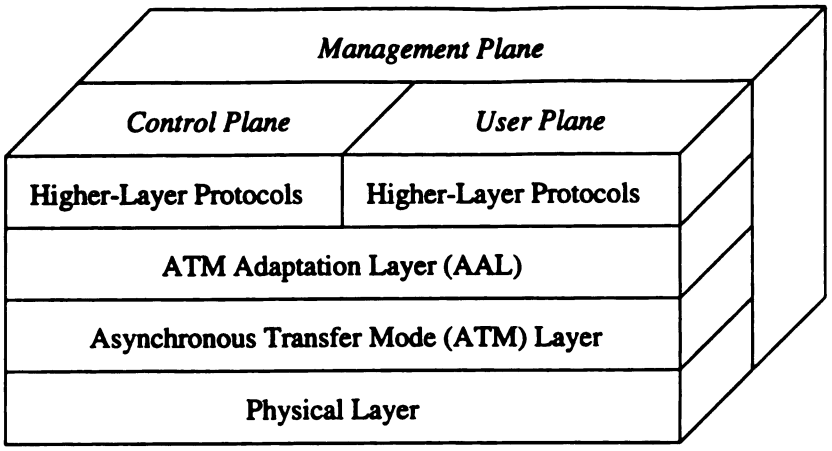

Figure 2.1: BISDN protocol reference model.

such as digital TV and full multimedia communication on a single network.

The underlying network technology proposed for supporting BISDN is the Asynchronous Transfer Mode (ATM) network based on the *cell-relay* concept. The underlying high-speed transmission system is proposed to be based on the Synchronous Optical Network (SONET) standard [16, 42, 72]. The ATM network is covered in Section 2.3 below.

2.3 Asynchronous Transfer Mode Networks

CCITT[3] has proposed a protocol reference model for BISDN [7, 16, 60, 63, 73, 75], which is shown in Figure 2.1. The physical layer is based on SONET [16, 42, 72], which is a hierarchy of optical signaling rates starting at 51.84 Mbps and ascending to the multi-Gbps range.

The ATM layer is in charge of the multiplexing and switching of 53-octet data units called *cells*. The AAL is an endpoint-to-endpoint layer that provides a variety of services on top of the ATM layer.

[3]International Telegraph and Telephone Consultative Committee, now known as the International Telecommunications Union-Telecommunications Standardization Sector (ITU-TSS or ITU-T for short)

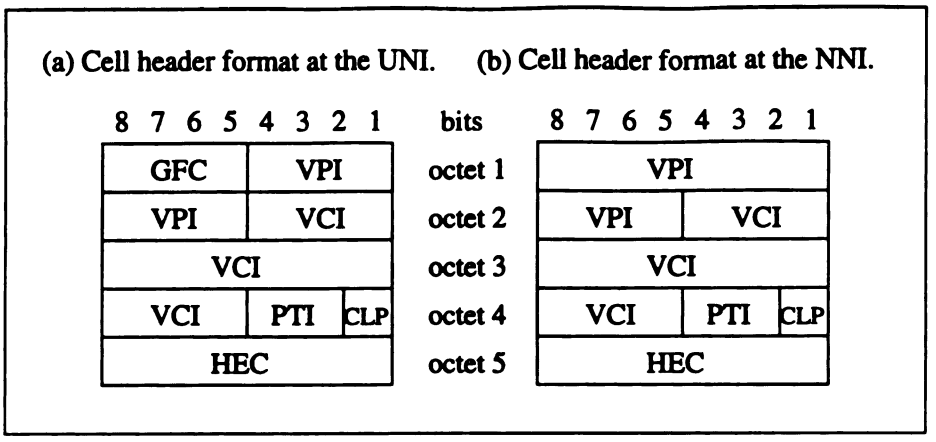

Figure 2.2: Cell header formats.

The higher-layer protocols are divided into a user plane and a control plane to provide out-of-band signaling over the same ATM subnetwork. Finally, the management plane manages all the planes and layers in the protocol stack.

2.3.1 The ATM Layer

Asynchronous Transfer Mode (ATM) [7, 11, 16, 24, 42, 60, 72, 75] is a technique of transporting, multiplexing, and switching fixed-sized data units called *cells* using special hardware called *ATM switches* [66]. An ATM cell consists of a 5-octet header field and a 48-octet payload. The ATM layer processes a cell based solely on the information in the header field, the format of which is shown in Figure 2.2.

Figure 2.2(a) shows the format of the header at the User-Network Interface (UNI), which includes a *generic flow control* (GFC) field that is used to control flow or indicate priority levels at the edge of the network. The GFC field is not used inside the network and is not included in the header format at the Node-Network Interface (NNI) shown in Figure 2.2(b).

The *virtual path identifier* (VPI) and *virtual channel identifier* (VCI) indicate the virtual connection to which the cell belongs. ATM switches contain routing tables and procedures for properly switching (i.e., forwarding) cells based on their VPI and

AAL type	AAL-1	AAL-2	AAL-3	AAL-4	AAL-5
Synchronization	required		not required		
Bit rate	constant	variable			
Connection mode	connection-oriented (CO)			connectionless	CO

Figure 2.3: AAL protocols and service types.

VCI labels. If the cell belongs to a virtual path connection, then it is switched based on the VPI alone—the destination port and VPI are obtained from the routing table and the VCI is left unaltered. Otherwise, the cell is switched according to the VPI and VCI combined, and new VPI and VCI labels are obtained from the routing table.

The *payload type identifier* (PTI) indicates whether a cell is a user cell or some other special-purpose cell. The *cell loss priority* (CLP) bit indicates whether the cell is recommended for dropping in case of congestion. This is used to distinguish between cells that are costly to lose from those that are less so. Finally, the *header error control* (HEC) field contains a Cyclic Redundancy Check (CRC) code used to detect bit errors in the cell header.

2.3.2 The ATM Adaptation Layer

The ATM adaptation layer (AAL) [7, 11, 16, 24, 42, 60, 75] provides a variety of services for applications with different traffic requirements. A list of different proposed AAL protocols (other than the *null* AAL) and the types of traffic characteristics they support are given in Figure 2.3.

AAL-1 is essentially *circuit emulation* and is appropriate for *constant bit rate* (CBR) synchronized services such as CBR audio and video. AAL-2 is appropriate of *variable bit rate* (VBR) synchronized services such as the variable-rate (i.e., compressed) versions of audio and video.

AAL-3, AAL-4, and AAL-5 are intended for bursty data services. AAL-3 and AAL-4 used to be separate standards but these have been merged into one protocol called AAL-3/4 because of their similarities. AAL-3/4 provides two modes of data service: *message mode* service for framed data (compatible with SMDS) and *streaming mode* service for data streams. AAL-5 [77] is a simple and efficient AAL protocol that is optimized for connection-oriented data service. It has less processing and transmission overhead that the more general-purpose AAL-3/4.

The AAL protocols are divided into two sublayers. The *segmentation and reassembly* (SAR) sublayer produces cells from the higher-layer data units (segmentation) at the sending end, and reconstructs the higher-layer data units (reassembly) from cells at the receiving end. The *convergence sublayer* (CS) performs adaptation functions other than segmentation and reassembly, such as synchronization for AAL-1 and AAL-2 and cell loss and error detection for AAL-3/4 and AAL-5.

2.3.3 QOS Guarantees in ATM Networks

ATM virtual connections are provided with *quality of service* (QOS) guarantees [11, 46, 83]. This means that during the life of the connection, its traffic will experience cell losses and delays that are within specified ranges.

Physical transmission systems are always subject to bit errors and hence will occasionally cause cells to be lost when the cell header is corrupted. ATM networks do not perform link-by-link error correction; therefore, these losses are not recovered within the network. Another cause of cell losses is buffer overflow within the ATM switches. Because of the stochastic and asynchronous nature of the traffic in ATM networks, buffer overflow is inevitable.

Cell delay and delay jitter (variation in the delay) happen as a result of the competition that exists between active connections and the buffering of cells within

ATM switches. Again, the stochastic and asynchronous nature of the ATM approach contributes to the occurrence of cell delay and delay jitter.

In order to control the cell losses and delays among the connections in an ATM network, before a new connection is admitted, the question has to be asked [46]

"Can the requested call be accepted by the network at its requested QOS, without violating existing QOS guarantees made to on-going calls?"

To answer this question, the network will require, among other things, descriptions of the source traffic characteristics and the QOS requirements of the new connection.

The traffic descriptors may include parameters such as the mean rate, peak rate, mean burst length, maximum burst length, and proportion of time in the active state. The QOS requirements may be specified in terms of throughput, cell loss rate, burst blocking probability, delay bounds, and delay jitter bounds.

Guaranteeing the QOS of connections is a matter of employing a variety of traffic control techniques. Some of the available techniques are the following:

- rate control and/or flow control
- traffic enforcement and shaping
- bandwidth and buffer management

These are covered in Chapter 3.

Finally, it is instructive to explain what is meant by "best-effort" traffic or "best-effort" service. In its common usage in the data communications field, "best-effort" is not to be interpreted as "highest quality" (i.e., no better service exists). Rather, it is to be considered as an abbreviation for "best but non-guaranteed delivery using residual resources." A new term, *available bit rate* (ABR), has been coined to distinguish it from (and to rhyme with) CBR and VBR.

Chapter 3

Related Research: High-Speed Network Traffic Control

This chapter covers research topics in high-speed network communication that are related to the author's subject of interest, which is burst-level traffic control in ATM networks. Network traffic and source traffic characterization and modeling are relevant in the design and performance evaluation of traffic control schemes. Traffic characterization and modeling is covered in Section 3.1. In high-speed networks, fast connection establishment techniques are needed for minimizing the bandwidth wasted during connection setup, and for efficiently supporting connectionless services. Connection establishment techniques are discussed in Section 3.2. High-speed wide-area networks require preventive rather than reactive congestion control. This is explained in Section 3.3. Finally, a variety of traffic control schemes for high-speed networks are described in Section 3.4. (N.B. In this chapter, with a few exceptions, the related work reviewed in each (sub)section is presented in chronological order of publication.)

3.1 Traffic Characterization and Modeling

In this section, various studies of network and source traffic characteristics, and models for such, are discussed.

Heffes and Lucantoni [32] present a performance study of a statistical multiplexer of packetized voice and data traffic. In the study, the superposition of packetized voice sources is modeled using a Markov-modulated Poisson process (MMPP). Each voice source is assumed to be a renewal process that alternates between talk spurts that are approximately exponentially distributed (actually geometric; i.e., discrete) and silence periods that are exponentially distributed (with a different mean). The superposition of these processes is a complex non-renewal process. A two-state MMPP is a doubly-stochastic Poisson process whose arrival rate is determined by a two-state continuous-time Markov chain. It is also a non-renewal process but is simpler than the superposition process. It has four parameters: the mean sojourn times of the two states of the Markov chain, and the arrival rates of the Poisson process at each state. Using sophisticated techniques, the four MMPP parameters are obtained to match four statistical characteristics (involving first, second, and third moments) of the superposition process. The MMPP approximation is compared with results from simulation and is shown to be more accurate than a comparable Poisson process model.

In [38], Jain and Routhier propose a model for network traffic based on "packet trains." In the packet train model, packets are assumed to travel in groups called trains. Trains are identified in terms of a *maximum allowed inter-car gap* (MAIG) parameter. Interarrival times less than the MAIG indicate "cars" in the same "train" and those greater than the MAIG indicate gaps between distinct trains. The packet train model is therefore defined by specifying an inter-train arrival process and an inter-car arrival process. The authors analyze the packet arrivals in an actual 10 Mbps

token-ring network and show that the packet train model fits the observed arrival process much better than the Poisson or compound-Poisson process models. The authors summarize the appropriate parameters (e.g., mean and coefficient of variation of the inter-car time) for that network but do not propose any well-fitting probability distributions.

In [15], Cáceres and others discuss the results of their study of the characteristics of wide-area TCP/IP conversations. In the study, conversations between application endpoints communicating via TCP/IP are analyzed in terms of conversation length (in bytes), conversation duration (in ms), number of packets transferred, packet sizes, and interarrival times. The results of the study include the following (among others):

- Most classic bulk transfer applications transfer less than 10 Kbytes per connection.
- A large portion of bulk transfer applications show strongly *bidirectional* traffic flow.
- While interarrival times for bulk transfers exhibit the packet-train phenomenon, the interarrival times for interactive applications closely resemble a *constant plus exponential* distribution.
- Most interactive applications generate 10 times more data in one direction than the other.

In the article, the authors also suggest a way to use the findings in the study to randomly generate source traffic for an internetwork traffic simulation model.

Wirth and Meier-Hellstern [85] propose a canonical model for ISDN packet data traffic. The model is based on a three-state Markov chain packet generator. One state represents machine-generated packet arrivals, another one represents active typing, and the last one represents pauses such as host response time and user think time.

Each state generates a run of packet arrivals. The interarrival time distribution, run length distribution, initial probability, and transition probabilities of each state are estimated from actual measurements obtained from monitoring real ISDN user activity. The resulting model parameters and approximating distributions, which are quite complicated, are given in detail in the article.

In [10], Bottomley and Nilsson present their study of the traffic characteristics of the CoNCert network, which is a wide-area network covering a great portion of the state of North Carolina. In their study, they observe the following.

- The interarrival distribution does not appear to be exponential. In particular, the coefficients of variation of the interarrivals are observed to be between 1.3 and 1.7 (compared to 1.0 for the exponential distribution).
- The packet length distribution is clearly bimodal with peaks at the minimum and maximum packet sizes.
- Two of the three sampling locations appear to have correlated interarrival times. The other location, which happens to be the busiest, does not show significant correlation. (Does this suggest that the combination of a large number of processes does eventually result in an uncorrelated process? This is an open question.)

The article also summarizes the results of their modeling efforts using a semi-Markov process and two-state MMPP as in [32]. Both models seem to fit the interarrival distribution well. However, both models do not seem to capture the correlation process very well. In any case, the MMPP model is recommended in lieu of a renewal process model (e.g., a Poisson process).

Leland and others evaluate Ethernet LAN (also known as a CSMA/CD or IEEE Standard 802.3 LAN [79, 73]) traffic in [47]. The main proposition of the article is

that Ethernet LAN traffic is a *self-similar* or *fractal-like* process. In particular, the "burstiness" of Ethernet packet arrivals exhibits itself across a wide range of time scales. This conclusion was drawn from rigorous statistical analysis of a very large set of high-precision Ethernet measurements. In the article, the authors contend that their findings have dire consequences on congestion control in future broadband networks. However, it is not clear how the traffic properties of a LAN using a largely undisciplined connectionless protocol would apply to a broadband network, such as an ATM network, that is based on connection-oriented controlled access, i.e., call admission control and usage parameter control (see Section 3.4). That is, it may well be the case that most of the self-similarity would be ironed out by these access control mechanisms. However, their observations would very likely be relevant in the context of *ATM LAN emulation* [50] or connectionless "best-effort" protocols in general (see Section 3.4.4).

3.2 Connection Establishment Techniques

In this section, various proposed connection establishment techniques for high-speed networks, including ATM-like networks, are discussed.

An adaptive dynamic technique of caching virtual circuits in connection-oriented packet-switching networks is proposed by Jomer in [39]. The proposal is about using an X.25 (connection-oriented packet switching) network[1] to connect LANs. The LANs communicate in a connectionless fashion using the X.25 network as a virtual data link service. In the system, virtual circuits are cached in the sense that a virtual circuit that is set up to transfer a packet from one LAN to another is kept for some time after that packet is delivered so that it may be reused by a subsequent packet with

[1]Granted, an X.25 network is not a high-speed network; however, the ideas in the article seem to be applicable to faster networks that are also based on virtual circuit communication.

the same source and destination LAN. An unused virtual circuit is released only after a timeout expires. The author proposes a traffic adaptive timeout mechanism that varies the timeout value depending on the current traffic load in order to optimize the performance of the system.

Cidon and others [17] give a fast algorithm for setting up connections in fast packet-switching networks. A key objective in their design is to make bandwidth reservation very fast in order to reduce the call blocking that is due to call establishment request processing overhead. Routing is done using Automatic Network Routing (ANR), also known as *source routing*, in which the route of a packet is predetermined at the source and prefixed to the data in the packet. The algorithm takes advantage of a *selective copy* feature that allows an establishment request packet to be forwarded to the next node while it is being delivered to the current node's network control unit. In addition to the detailed protocol algorithms, proofs of correctness are provided in the article.

Doeringer and others [21] propose a hierarchical approach to connection establishment in large-scale networks. In the proposal, the network is organized in a hierarchy so that in the global view of the network, the internals of the nodes are hidden. A node in the global view may be non-trivial in the sense that it may consist of several interconnected sub-nodes or a set of multiple internal paths in a switching fabric. The internal (intra-node) details of the connection establishment procedure (i.e., path computation and bandwidth reservation) are done in parallel among all the nodes after the path is computed on the network (inter-node) level. The authors argue that the parallel approach is more efficient than the traditional approach and that its execution time is independent of the network complexity but is essentially a function of the round-trip delay.

Olsen and Landweber [55] present a method for fast virtual channel establishment in ATM-like networks. The method, called NoW (No Waiting), uses in-band signaling

to allow the data cells to be sent immediately after the VC connection establishment message. Logically, the technique is equivalent to using preestablished virtual paths in the network so that setting up a connection becomes a matter of choosing available VCIs toward the destination. Scalability is achieved via partitioning of the network. The method does not concern itself with bandwidth allocation, only VC label management. A software implementation described in the article requires the buffering of a few data cells while the connection is being set up. A hardware design that is supposed to be fast enough to require no buffering is also briefly described in the article.

Finally, a short overview of routing in ATM networks is presented by Onvural and Nikolaidis in [56]. The paper addresses some of the issues in routing with virtual circuits and virtual paths including bandwidth reallocation and VC/VP rerouting. The paper also discusses different routing methodologies from the ATM network perspective. The same authors have also compiled a bibliography on performance issues in ATM networks, including routing, in [51].

3.3 Preventive vs. Reactive Congestion Control

Traditional low-speed networks control traffic congestion via *reactive* means such as the use of *source quench* messages [18, 73] or *choke packets* [72, 73, 79, 84]. Similarly, link-by-link or end-to-end flow control is based on *feedback-based* techniques such as *stop-and-wait* and *sliding window* protocols [72, 73, 84]. These techniques are practical for low-speed networks because in those networks, the time to send a message is dominated by the *transmission time* ("the time to put the bits on the medium").

The high-speed links in future networks will shrink the transmission time considerably, making it insignificant compared to the *propagation delay* ("the time for the bits to travel through the medium") in wide-area networks. The large propagation

delay to transmission time ratio will make *reactive* and *feedback-based* traffic control techniques inappropriate for high-speed networks for reasons of efficiency and cost. On the one hand, if a conservative approach is used, such using *stop-and-wait* or reasonably-sized *sliding windows*, then the network will be under-utilized (see [73]). This is so because the links will tend to be idle for relatively long periods while the endpoints are waiting for acknowledgments for messages that are transmitted almost instantaneously. On the other hand, if an aggressive approach is used, then unless very large receiving-end buffers are available throughout the network, large amounts of data may be lost during periods of congestion. This is a consequence of the large propagation delay experienced by the congestion control messages themselves—by the time these messages reach the sources, megabits or gigabits of data may already be on the way (i.e., traveling on the medium). (See [43] for a more detailed discussion of the above issues).

The preferred approach for high-speed networks in the future is therefore *congestion avoidance* or *preventive congestion control*. Preventive congestion control in ATM networks, in which data transmission is connection-oriented, can be broadly categorized into *call admission control* and *usage parameter control*. Both of these types of controls are essential in guaranteeing quality of service and work hand-in-hand.

Call admission control (CAC) schemes control the traffic entering the network by limiting the number of VC and VP connections in the network. A CAC scheme accepts or rejects connection requests based on the traffic characteristics indicated in the request and the traffic load offered by the other preestablished connections. CAC schemes are concerned with high-level resource allocation including bandwidth and buffer allocation. These are discussed in Section 3.4.2 below.

Usage parameter control (UPC) schemes are applied to the admitted connections to ensure that they comply with the accepted traffic description. Since they control the rate at which an application or site sends information into the network, they

are also known as *rate control* schemes. Similarly, they are sometimes called *source policing* schemes and their job *bandwidth enforcement*. Some UPC schemes change the behavior of the input traffic streams and are therefore called *traffic shaping* schemes. UPC schemes are covered in Section 3.4.1 below.

3.4 Traffic Control Schemes

This section covers related research proposals that have been published in the area of traffic control in high-speed networks and especially ATM networks. To begin, here are a number of published surveys or overviews on traffic control issues and proposals for high-speed networks.

- The survey of Bae and Suda [7] covers traffic source modeling (especially voice and video), admission control, bandwidth enforcement (especially leaky bucket), priority control, and ATM networks and protocols.

- The overview of Vakil and Saito [83] covers traffic description, admission control (several approaches), usage parameter control (especially leaky bucket), cell loss priority (CLP), and other congestion control strategies.

- In [30], Habib and Saadawi discuss congestion avoidance in high-speed networks at three levels including the network level (management of links and virtual paths to minimize virtual call blocking), the call level (admission control), and the cell level (leaky bucket and other schemes).

- The overview of Gün and Guérin [29] discusses an integrated set of controls including, among others, bandwidth allocation based on equivalent capacity [28] and rate control based on a buffered leaky bucket and spacer scheme [8].

- Doshi and Johri [24] survey communication protocols for high-speed networks. Some relevant topics include ATM protocols, congestion control, admission control, and dynamic bandwidth controls (e.g., in-call negotiation).

- Nikolaidis and Onvural [51] provide an extensive bibliography on performance issues in ATM networks, including traffic source characterization, call admission, congestion control, source policing, and fairness.

The rest of this section is organized as follows. Section 3.4.1 goes over various proposals for usage parameter control and Section 3.4.2 deals with proposals for call admission control. Burst-level traffic control proposals are covered in Section 3.4.3. Section 3.4.4 covers special schemes for best-effort traffic. Finally, a few special-purpose schemes for delay-sensitive traffic (e.g., audio and video) are covered in Section 3.4.5.

3.4.1 Usage Parameter Control Schemes

Most likely, the most widely-known input rate control schemes are those based on the "leaky bucket" scheme which is described by Turner in [81]:

> "... A counter associated with each user transmitting on a connection is incremented whenever the user sends a packet and is decremented periodically. If the counter exceeds a threshold upon being incremented, the network discards the packet. The user specifies the rate at which the counter is decremented (this determines the average bandwidth) and the value of the threshold (a measure of burstiness)...."

There are many variations and extensions of this scheme. One generalization proposed in [8] is a combination of a *leaky bucket*, *marker*, and *spacer* shown in Figure 3.1. This particular version is for variable-sized packets that consume a variable

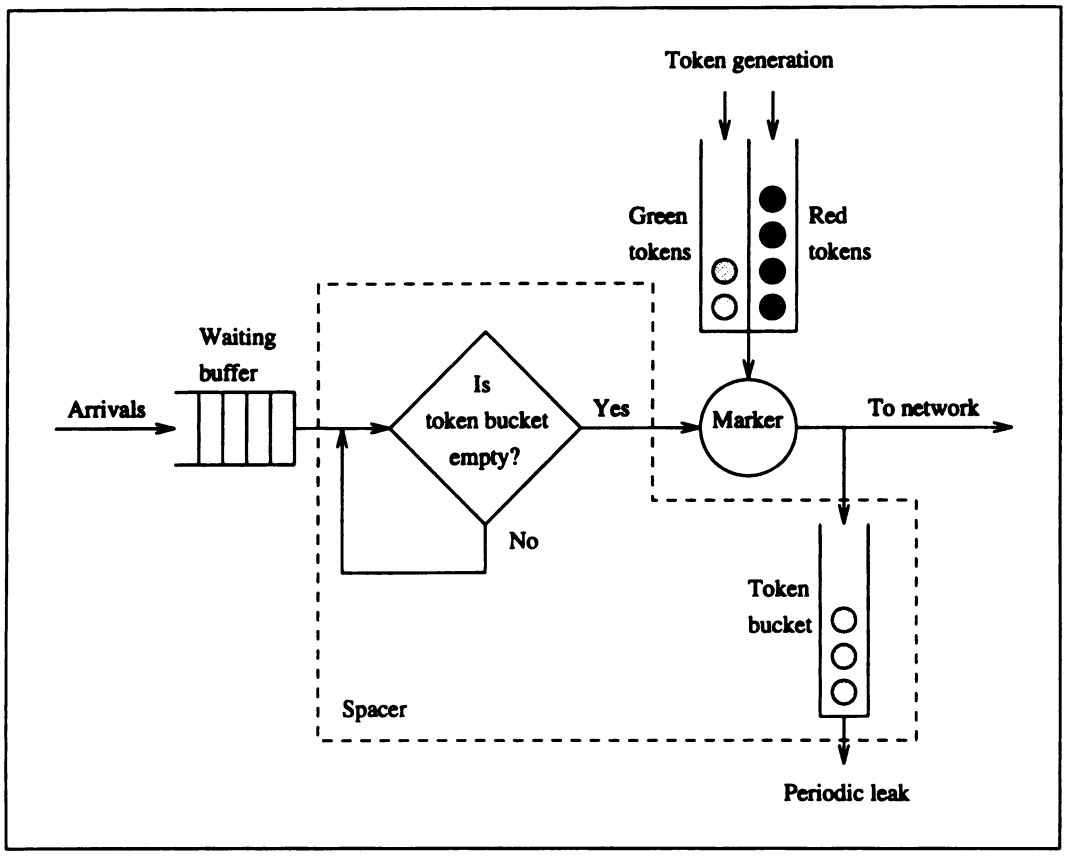

Figure 3.1: Generalized leaky bucket with marker and spacer.

number of tokens proportional to their lengths (for fixed-sized cells, the bottom token bucket will only have to hold at most one token). Colored tokens are obtained from the top buckets and dropped into the bottom bucket which leaks periodically at a fixed rate determined by the maximum rate allowed for the source. High-priority green tokens are generated according to the reserved average rate of the traffic source. Low-priority red tokens are generated to accommodate traffic exceeding the reserved rate as "best-effort" traffic. These are used whenever a packet cannot conveniently wait for the necessary green tokens because the buffer occupancy has reached a certain threshold. A packet waits in the buffer until the bottom bucket is empty and is then marked "green" or "red" as appropriate. It then drops its tokens in the token bucket before entering the network. Inside the network, green packets are given preferential treatment using a buffer-threshold policy. This is done by limiting the number of red packets in the node buffers, even when the buffers are not yet completely full, in order to leave room for additional green packets.

Another generalization of the leaky bucket scheme is presented by Bovopoulos in [12]. The scheme, called *traffic control mechanism* (TCM), is essentially a buffered leaky bucket scheme with a more sophisticated token generation pattern. Instead of generating the tokens in a constant periodic rate, they are generated according to a multi-state, cyclic, and deterministic Markov chain. For each state of the Markov chain, a fixed number of tokens is generated. The time between state transitions of the Markov chain is fixed. It is claimed in the article that the more general token generation pattern will support a wider variety of traffic behaviors.

The effectiveness of the (unbuffered) leaky bucket scheme is analyzed by Butto' and others in [14]. Their results show that the leaky bucket can be effectively used to control the peak rate of a source. However, for controlling the mean rate without introducing unacceptable cell loss rates, a long observation time (high counter threshold) is required. Finally, they conclude that the leaky bucket is insensitive

to the burst and silence length distributions (of bursty *on-off* sources) and hence is inappropriate for controlling burst durations.

Similarly, the performance of the buffered leaky bucket scheme is analyzed by Holtsinger and Perros in [33]. Using a two-state Markov Modulated Bernoulli Process (MMBP) model for the cell arrival process, they show that the buffered leaky bucket can be configured to control the mean rate or the departure burstiness (expressed in terms of the squared coefficient of variation of the inter-departure time) but not both. However, the authors suggest that both configurations may be used together. The performance of the unbuffered leaky bucket is compared with that of the buffered version. The results suggest that the buffered scheme is more effective at controlling the burstiness of the cell arrivals, but suffers from introducing a non-zero waiting time on the cells. Finally, the authors also show that there is a tradeoff between the arrival cell loss rate (at the buffer) and the departure burstiness.

In [62], Rathgeb compares the performance of the (unbuffered) leaky bucket (LB) mechanism with several other policing mechanisms, including the *jumping window* (JW) mechanism, the *triggered jumping window* (TJW) mechanism, the *exponentially weighted moving average* (EWMA) mechanism and the *moving window* (MW) mechanism. Using a combination of analysis and simulation methods, the policing schemes are evaluated in terms of violation probability for well-behaved sources, sensitivity to short-term and long-term overload, and worst-case traffic allowed. The LB and EWMA mechanisms are found to be the most promising approaches by virtue of their flexibility to handle short-term statistical traffic fluctuations.

Shimokoshi [71] proposes the use of "pseudo cell buffering" to enhance the JW and MW mechanisms. In general, window-based mechanisms work by imposing a certain limit on the number of cells in a time window. The pseudo buffering mechanism allows this limit to be exceeded (up to the pseudo buffer limit) with the excess to be charged

in the next window. Using simulation techniques, the performance characteristics of the *pseudo jumping window* (PJW) and *pseudo moving window* (PMW) mechanisms are compared with the (unbuffered) LB, JW, and MW methods. The results show that the LB and PJW mechanisms are the most effective at regulating the mean source rate.

3.4.2 Call Admission Control Schemes

In this section, a variety of call admission control (CAC) schemes are discussed. CAC schemes typically manage bandwidth and buffers at the connection level, in order to guarantee a quality of service (QOS) to the admitted calls based on the expected cell loss rate (CLR).

Hui's monograph [35] discusses multi-level resource allocation for ATM/BISDN networks. Being a relatively early article, it has a few discrepancies with subsequent developments, including the persistent use of the word *packet* rather than *cell*, the absence of VP/VC routing (connections can have multiple routes set up), and the possibility of cells arriving out of order (which is not in line with the virtual circuit view of an ATM connection). In the proposal, traffic is to be managed at three levels: the packet or cell level, the burst level, and the call level. Blocking probabilities are to be maintained at each level. At the call level or burst level, a *pilot packet* is used to indicate the amount of bandwidth or change of bandwidth required by the source. This packet may also contain other information such as the estimated holding time for the bandwidth increase, but the author does not propose how such information is to be used except maybe in the computation of the call blocking probability which uses "holding time statistics." At each level, the algorithm keeps routing and load information and makes decisions based on computations of blocking probabilities. The computations are to be done according to the methods presented for the three levels.

In conclusion, the article presents simulation results with the following findings:

- VBR traffic benefits more from statistical averaging than bursty (*on-off*) traffic.
- Bursty traffic utilization is more sensitive to the tolerable blocking probability than VBR traffic utilization.

In [27], Gallassi and others present a *class related rule* (CRR) for bandwidth assignment in ATM networks. Classes consist of homogeneous *on-off* sources with the same peak rate, burstiness (peak rate to mean rate ratio), and mean burst length. With a predetermined buffer size (which determines the cell delay limit) and a target cell loss probability, the bandwidth requirements of each class (for different numbers of sources in the class) are determined via *simulation* (and stored). The CRR is then used to determine the required bandwidth of a set of classes of varying number of sources.

The *class related rule* (CRR) from [27] is evaluated by Decina and others [20] using simulation. The performance gains obtained using the CRR is found to be significant for burstiness values greater than five. The CRR appears to be a conservative approach for the assumed traffic for a wide range of cases. The call-level blocking probability was also evaluated for CRR and was found to offer significant gains.

Suzuki and others [76] propose a bandwidth allocation strategy based on the concept of "virtual cell loss probability," which is a quality estimation method based on a "logically buffer-less" model instead of a buffered queueing system. The method sacrifices link utilization for simplicity and robustness. It uses only the maximum and average rates of the calls (with *on-off* sources) to estimate their virtual cell loss probability, i.e., no burst lengths or buffer size parameters are used. The connections are grouped into classes where each class has a maximum rate, average rate, and number of calls. The performance of the approach is analyzed and the authors suggest that traffic with high maximum rates (with respect to the link capacity) be allocated

the maximum rate, and that available capacity be separately reserved for such traffic and CBR traffic.

Guérin and others [28] propose a method for computing the "equivalent capacity" of connections. This method uses a fluid-flow two-state Markov source model approximation. The traffic parameters are the peak rate, the *source utilization* (fraction of time the source is active), and the mean burst length for exponentially distributed burst and idle periods. The effective bandwidth is computed for a single source, or for a combination of sources in which case the statistical gain is higher—i.e., the effective bandwidth of a combination is less than the sum of the effective bandwidths of each individual in the combination. The scheme uses approximations to make computations achievable in real-time. Two complementary formulas for equivalent capacity are used. A "fluid-flow approximation" is used for low numbers of sources and a "stationary approximation" is used for large numbers of sources. The authors suggest the use of standard grades of service for which the effective bandwidths have been precomputed. Extensions to handle non-exponential distributions are discussed. One major disadvantage of the approach mentioned in the article is the fact that it is not effective in those cases where the connections have long bursts or low source utilization.

In [70], Saito proposes a call admission control scheme for ATM networks using output buffers. The scheme guarantees the cell loss probability and the author claims (perhaps misleadingly) that it does not have any assumptions on the cell arrival process. A buffer size is assumed that guarantees that the cell delays will remain within bounds. The traffic parameters required are the average and maximum number of cell arrivals in an interval equal to the time required to half-fill the buffer (note that the only case where this is not a restriction on the traffic process is when the interval is infinite–the smaller the interval, the more restrictive it is). The performance of the scheme was investigated and it was concluded that the use of only the maximum

and average number specification sacrifices some utilization loss compared to more detailed traffic specifications.

3.4.3 Burst-Level Control Schemes

This section covers some proposals on handling bursty data traffic at the burst level. Already mentioned in Section 3.4.2 is Hui's multi-layer control scheme [35] which uses *pilot packets* to reserve bandwidth for bursts. A burst-level approach are also briefly described by Ohnishi and others in [54] in the form of a *short hold mode* service.

In the early 1980s, researchers from GTE developed the "burst-switching" approach [3] for supporting integrated voice and data traffic. The burst-switching concept is described in [31] as follows:

> "Information contained within the header of each voice or data burst establishes a path through the network. A burst consists of a header (containing routing, control information and error control bytes) , followed by the completely variable length information burst and ending with an end-of-message byte.... The header effectively latches up contiguous switch network channels for the duration of the burst."

Transmission on traditional digital carriers such as the T1 carrier is typically organized into *frames* divided into separate *channels*. In burst switching, these channels are statistically multiplexed among the connections on the burst level. That is, rather than dedicating the channels to bursty connections for the duration of their lifetimes, the channels are assigned to currently active connections on demand. The scheme depends on speech silence detection to take advantage of the fact that voice connections are active only 35% to 40% of the time. Both voice and data bursts are buffered in case of congestion, except that a buffered voice burst is limited to a few milliseconds worth of voice samples, after which speech clipping or *freeze-out* [49] is done. To

minimize freeze-out, voice bursts are to be given higher priority over data bursts by the resource allocation scheme.

In [13], Boyer and Tranchier propose fast reservation protocols for bursty traffic. The article is primarily about FRP/DT (fast reservation protocol with delayed transmission) but also introduces its sibling FRP/IT (fast reservation protocol with immediate transmission). FRP/DT works best for highly bursty traffic where the transmission time for a burst is large compared to the round trip delay. The round trip delay (RTD) is required for reserving bandwidth. If the bursts are short, the efficiency (utilization) becomes compromised. FRP/IT is tailored for applications with smaller bursts or which cannot tolerate the setup time (including the RTD). Their analysis shows that FRP/DT allowed for a statistical gain equal to or greater than that of ideal rate-based multiplexing which required more stringent traffic policing. The scheme uses the *burst blocking probability* (BBP) as a parameter analogous to the *burst congestion probability* (cell loss rate) in traditional call admission control schemes. FRP/DT is centralized in the sense that the user communicates with an *FRP Control Unit* which handles all the signaling within the network. It allows a user to change its bandwidth allocation in a stepwise fashion through negotiation such that unsuccessful negotiations do not prevent it from using its previously allocated bandwidth. FRP/IT, which is based on ideas from [35], is only briefly described in the article. Adaptive in-call renegotiation schemes for burst-level resource allocation protocols such as FRP/DT are proposed in [23] (for long file transfers) and [22] (for short intermittent file transfers).

Turner proposes a fast buffer reservation scheme in [82]. The scheme allocates buffers to bursty virtual circuits on a burst-by-burst basis. The bursty traffic includes *start* and *end* cells to indicate the beginning and end of bursts. The arrivals of these cells and the *middle* cells, together with a timeout mechanism, affect a state machine associated with each bursty virtual circuit. The state machine has two

states representing the *idle* and *active* states of a connection. Because of the timeout mechanism (which is intended to protect against cell losses but at the same time remain tolerant to inter-cell delay jitter), the author proposes that the scheme be used only for virtual circuits with peak rates exceeding 1% to 2% of the link rate. In addition, there are computational reasons why high peak rate to link rate ratios are desirable in the scheme. The multiplexing efficiency of the scheme was analyzed and presented, albeit for relatively high blocking probabilities (0.01 and 0.001). In the analysis, the author did not address the issue of excessive overhead (i.e., high control cell to data cell ratio) for high burstiness (peak rate to mean rate ratio) or low peak rate cases. Finally, the scheme does not seem to be based on the equivalent bandwidth concept and requires the re-computation of all *contention probabilities* (burst loss probabilities) for all circuits whenever a new one is added.

Crosby investigates the approach of *in-call renegotiation* in [19]. The most important result from the study is that renegotiation offers the most benefit in those cases where the ratio of the peak rate to link rate is high. When the statistical multiplexing level is high (e.g., high speed backbone links), it offers little gain. The analysis uses two types of source models: the familiar *on-off* source and a more sophisticated source with a miniprocess to characterize an *on* state. The performance of the following approaches are compared:

- static allocation (no renegotiation but with a bound on the burst blocking probability)
- simple renegotiation (renegotiation is performed before every state change), and
- optimal renegotiation (the algorithm is unspecified; the performance is determined using upper/lower bounds analysis).

With renegotiation, the traffic load is controlled with a parameter P_b, the probability of renegotiation failure. The results show that the simple renegotiation scheme offers

increased utilization when the ratio of the peak rate to the link rate is high. However, the results also show that the simple scheme is far from optimal. It should be pointed out that in the study, the author only considered the case of one ATM multiplexer and did not take into account a possibly large round-trip delay (RTD); more realistically, one would need to take the RTD from source to destination into account, especially in evaluating the utilization.

3.4.4 Special Schemes for Best-Effort Traffic

In this section, special schemes intended for best-effort or *available bit rate* (ABR) service are described. These schemes do not aim to provide a guaranteed QOS in terms of throughput or delay.[2] They are primarily intended for connectionless communication where the characteristics of the traffic generation cannot be adequately specified or where throughput and delay guarantees are just not necessary. These schemes, however, aim to distribute the available unreserved bandwidth and buffers fairly among the ABR connections (for some appropriate definition of "fairness"). Moreover, they aim to totally prevent cell losses due to buffer overflow. These schemes do not depend on call blocking to control congestion.[3] However, they have the property that the more ABR (and bandwidth-guaranteed) connections there are, the poorer the service each ABR connection will receive. Furthermore, they would generally require large buffers to avoid cell losses while keeping the bandwidth utilization high in the case of high-speed long-distance connections. There are two main camps in the ABR debate: the *rate-based* and the *credit-based*.

The *rate-based* flow control approach for ABR service is surveyed in [9]. Conceptually, it is a feedback-based source rate throttling scheme where the destination

[2]However, some proposals such as EPRCA described in this section include support for a minimum cell rate, in which case that portion of the service is guaranteed.

[3]Naturally, a scheme supporting non-zero minimum cell rates will have to resort to call blocking if the minimum rate cannot be guaranteed.

and intermediate nodes participate in congestion detection. The culmination of the evolution of this concept is called Enhanced Proportional Rate Control Algorithm (EPRCA). EPRCA has three main components.

1. The first component is a source rate control mechanism that can raise or lower its sending rate based on feedback from backward Resource Management (RM) cells. A source periodically sends RM cells in the forward direction and periodically decreases its sending rate (by default). It increases its sending rate whenever it receives a backward RM cell with a negative Congestion Indication (CI) flag. Backward RM cells also contain an Explicit Rate (ER) field. A source receiving an ER value less than its current sending rate lowers it rate to the ER value.

2. The second component consists of congestion detection and congestion indication mechanisms at the intermediate switches. Basic congestion detection may be done by monitoring the instantaneous queue lengths in the switches. Congestion indication is done by the switches either by setting the Explicit Forward Congestion Indication (EFCI) flag in the forward *data* cells or by setting the CI flags in the forward RM cells. Switches may also optionally send backward RM cells to the source.

3. The third component is a destination mechanism that returns backward RM cells to the source to inform it of any congestion detected in the forward direction. The destination bounces back forward RM cells to the source and may optionally set the ER or CI fields based on congestion at the destination or on EFCI information from the forward data cells.

The proposal also includes provisions for breaking up long end-to-end feedback loops into shorter segments in order to improve performance. In this case, the source and

destination roles are also performed by Virtual Source and Destination nodes at the endpoints of each segment. Overall, the main attraction of the rate-based approach is its relative simplicity of implementation. At the minimum, participating switches only have to be able to raise the EFCI flags in data cells when appropriate. Furthermore, it does not require hop-by-hop per-VC queueing within the switches.

The *credit-based* flow control approach for ABR service is described in [45]. It is based on the Flow-Controlled Virtual Channels proposal presented in [44]. The idea is to perform credit-based flow control on a link-by-link, per-VC basis. In general, credit-based flow control works by requiring a sending entity to get "credits" from the receiving entity before it can start forwarding data cells. The receiving entity gives credits to the sending entity based on buffer availability on the receiving end. This assures that the sent data cells will not be lost because of buffer unavailability on receipt. The receiving entity sends credit cells upstream at a rate proportional to the rate it can empty the buffers used by the virtual channel. Per-VC queueing together with round-robin scheduling is used to achieve fairness among the ABR connections. Fairness, high bandwidth utilization, and robustness under extreme conditions are the main strengths of the credit-based approach. Its main disadvantage is high switch complexity. This includes the per-VC queueing and credit management requirements of the scheme. Furthermore, the approach would generally require large buffers to keep the links highly utilized in the case of long-distance high-speed links. Finally, it also requires considerable bandwidth overhead for the credit cells.

In addition to the rate-based and credit-based approaches, there are hybrid schemes that have been proposed. These schemes aim to combine the strengths of the two main approaches in an integrated system. In [61], the authors contend that the rate-based approach would be appropriate for a (public) WAN, while the credit-based approach would be appropriate for LANs. A WAN involves large propagation delays and very large numbers of connections (VCs). These conditions amplify

the disadvantages of the credit-based approach. However, the simplicity of the rate-based approach based on EFCI marking becomes specially attractive in this case. Furthermore, the rate-based approach appears to be more conducive to proper billing of ABR services. In contrast, a LAN would greatly benefit from the advantages of the credit-based approach with less penalties. The ABR service essentially emulates the behavior of traditional LAN technologies, and the credit-based scheme offers high utilization, no congestion-based loss, and robustness in the presence of highly unpredictable traffic. Three levels of integration are described by the authors. The basic scheme is to strictly use the credit-based approach in the LAN and the rate-based approach in the WAN. A more general scheme uses the rate-based approach as the default, but provides the credit-based approach as an option (when supported by all switches in the path). The proposed scheme is to fully integrate the two approaches so that rate-based and credit-based switches may operate with each other. It involves using the RM cells to convey both rate information and credit information. Essentially, in each segment of adjacent rate-based switches, the (virtual) source is limited in *rate* by the rate-based switches and *volume* by the credit-based (virtual) destination. More details can be found in [61].

Another hybrid scheme is described in [36]. It combines the rate-based approach with link-by-link back-pressure control. The purpose of the link-by-link back-pressure control is to avoid cell losses due to sudden increases in queue length that may happen in spite of the efforts of the (not-too-conservative) rate-based control implementation. In the scheme, the rate-based congestion indication is performed whenever the queue length reaches a certain threshold. Back-pressure is applied whenever the queue length reaches an even higher threshold to give time for the rate adjustments to take effect.

3.4.5 Special Schemes for Delay-Sensitive Traffic

There are also specialized traffic control schemes intended for *delay-sensitive* (typically audio and video) traffic. Since this dissertation is primarily concerned with bursty data traffic, only a few of these specialized schemes will be mentioned in passing.

Congestion control based on the *selective discarding* of packets or cells is presented and analyzed in [52, 86]. These techniques take advantage of the fact that the perceived performance of audio and video transmission is more sensitive to the integrity of some portions of the encoded data than others. This allows the less important portions to be sent in low-priority cells for preferential discarding during congestion. The priority is to be indicated in the CLP bit of the cell header.

A scheme for smoothing VBR video traffic is presented and analyzed by Ott and others in [57]. The scheme dynamically adjusts the rate of sending cells from an input buffer in order to keep the rate as close to constant as possible but without violating the delay constraints. The technique is based on using the recent behavior of the traffic stream to forecast the future traffic behavior.

A variable bandwidth allocation scheme for VBR video is presented and analyzed by Pancha and Zarki in [59]. The scheme uses a simple prediction algorithm to estimate the bandwidth requirements of a video encoder. To further improve the performance, the scheme also takes advantage of *layered coding* where the encoded output is separated into different streams according to their relative importance. These streams are given different priorities with different loss characteristics.

Chapter 4

Framework for Burst-Level Traffic Control

Recall that in ATM networks, data are transferred in a connection-oriented mode with a guaranteed quality of service (QOS). The connections are statistically multiplexed in order to increase the number of concurrent connections and to keep the bandwidth utilization high. This approach is ideal for connections that carry traffic at a near constant rate. However, to accommodate highly bursty traffic, either large buffers have to be provided within the switches, or the bandwidth utilization has to be kept low. Otherwise traffic congestion due to bursts will cause frequent buffer overflows. Avoiding congestion at high utilization in the presence of bursty traffic is a major concern in the design of high-speed networks.

The most conservative approach to congestion avoidance is to allocate bandwidth deterministically (rather than statistically) to each connection. This means that each connection will hold bandwidth equal to its peak rate (to be policed at the source) throughout its lifetime. This approach does not offer any statistical multiplexing gain and will result in low bandwidth utilizations in the presence of highly bursty traffic.

Another approach to congestion avoidance is to have a call admission control

(CAC) procedure that will limit the number of concurrent connections so that the expected congestion is kept within acceptable limits. Based on parameters included in the connection establishment request, the CAC procedure determines if it has enough buffer space and bandwidth available to provide a guaranteed QOS (e.g., maximum cell loss rate) to the new connection. Depending on the approach used, the CAC procedure may also have to verify that the service guarantees to all previously established connections will continue to be met.

Several proposals for allocating bandwidth and/or buffers to bursty connections have been proposed. Many of these approaches are based on guaranteeing a maximum cell loss rate (CLR) to the connections (see Section 3.4.2 above). These approaches typically treat cells indiscriminately without regard to their belonging to a burst. For many applications (e.g., file transfer), this is undesirable because losing a single cell in a burst may make the rest of the burst useless. A more appropriate approach is to manage the bursty traffic at the burst level and try to maintain a maximum expected blocking or loss rate of bursts. This has been proposed by Hui [35] and subsequently by Boyer and Tranchier [13] and Turner [82] (see Section 3.4.3 above).

In this chapter, a framework for supporting highly bursty connections in ATM networks is presented. The proposal includes mechanisms designed to facilitate traffic control at the burst level. The framework is designed to support a wide variety of bursty connections, including connections that may have extremely long bursts and/or extremely long periods of inactivity. It also supports connections that are continuously active in a variable bursty fashion. Finally, it is capable of handling all of the above connection types on a burst-by-burst basis even when they are bundled into virtual paths. This last feature is an important capability that is taken for granted or overlooked by the other burst-level approaches that have been proposed.

Section 4.1 presents the types of burst-oriented connections that the framework is designed to support. The connection establishment phase is briefly covered in

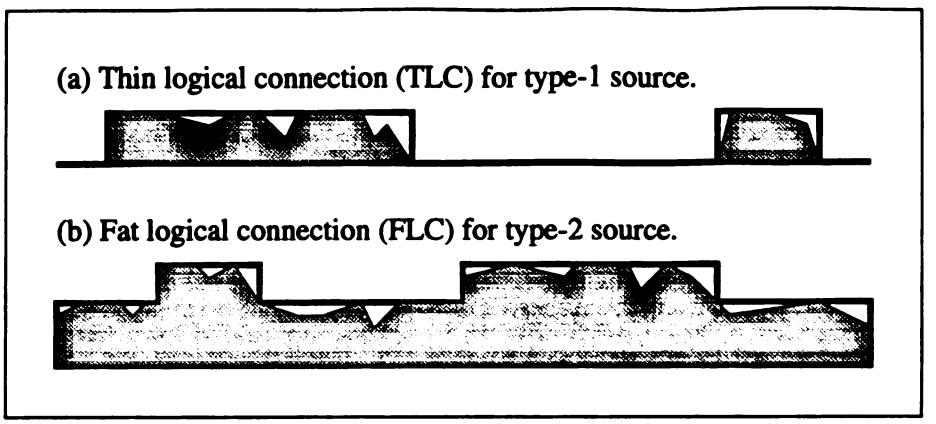

Figure 4.1: Thin and fat logical connections. The shaded areas indicate the instantaneous bit rates of the sources and the heavy lines indicate the bandwidth dynamically allocated to the connections.

Section 4.2. The basic mechanisms for proper burst-level handling of the traffic are described in Section 4.3 and Section 4.4. Advanced mechanisms including the methods of bundling these bursty connections into virtual paths are given in Section 4.5 and Section 4.6. Section 4.7 gives a summary of the chapter.

4.1 Burst-Oriented Services

We are interested in dealing with two types of bursty traffic sources, which we will call *type-1* and *type-2*. A *type-1* source, also known as an *on-off* source, alternates between periods of burst and silence. During bursts, it generates cells at a rate not exceeding its peak rate. Between bursts, it does not generate cells. A *type-2* source is a variable rate source that alternates between periods of high activity and low activity. It generates cells at a nominal non-zero rate between its high activity (burst) periods, which is unlike a *type-1* source that is totally silent. The nominal rate is allowed to change but only in a time scale larger than the burst arrival time scale.

To support these bursty sources, two kinds of burst-oriented connections are proposed; these are illustrated in Figure 4.1. These connections, called *thin* and *fat logical connections*, are essentially virtual channel connections with dynamically allocated

bandwidth and buffers. A *thin logical connection* (TLC) is a burst-oriented virtual channel connection that supports wait-free transmission of bursts with a guaranteed burst blocking probability (BBP). TLCs are intended for *type-1* sources. A *fat logical connection* (FLC) is a burst-oriented virtual channel connection for *type-2* sources (cf. [13]). It guarantees CBR-quality[1] service on the nominal rate portion of its traffic, and it allows the wait-free transmission of bursts above the nominal rate with a guaranteed BBP. Furthermore, through renegotiation, it allows the nominal rate to be increased or decreased.

In order for the network to handle the cell traffic from these connections on a burst-by-burst basis, a way has to be provided of recognizing the beginning and end of bursts. Each burst submitted by the source will be prepended with a reservation request cell (cf. [13, 35, 82]), which we will call the *head cell* and postpended with a reservation release cell, which we will call the *tail cell*. After an appropriate lead time, the data cells may be transmitted behind the *head cell* without waiting for acknowledgment. We will call this mode of transmitting the burst *eager transmission*, as opposed to *patient transmission* in which transmission does not begin until the reservations along the whole path are acknowledged.

All the data is sent through a TLC on a burst-by-burst basis. At the network interface (i.e., the ATM adaptation layer), the head and tail cells are added to each burst. Inside the network, the bursts are handled one by one. When a burst arrives at each switch, the connection is *activated* if bandwidth can be allocated to the burst at that time. Conversely, when a burst leaves, the connection is *deactivated* and the bandwidth is released.

In comparison, data is sent through an FLC in two ways. All bursts that can be sent without exceeding the established nominal rate are sent continuously. No

[1] delay and loss rate equal to that of a continuous bit rate (CBR) connection

burst-level handling is required for these bursts in the network, so head and tail cells are not required for them. However, bursts that will cause the aggregate rate to exceed the nominal rate have to be sent on a burst-by-burst basis, complete with head and tail cells. The additional bandwidth required is allocated when a head cell arrives and is deallocated when the tail cell departs.

In summary, these logical connections work as follows. First, they have to be established just like any virtual channel (or virtual path) connection before any data can be sent. This involves routing and virtual channel assignment. A TLC is not allocated any bandwidth or buffers initially. In contrast, an FLC is allocated resources sufficient for its nominal rate portion. The buffers and bandwidth reserved are equivalent to those required by a CBR connection of the same rate. After connection establishment, the connections are ready to carry bursty traffic as described above. Furthermore, an FLC is allowed to increase or decrease its nominal rate portion via renegotiation with the network.

Finally, note that it is not necessarily assumed that the bit rate of each source has a perfect square wave-form pattern. Rather, it is assumed that the rate of each connection is limited to the declared peak rate[2] at the source. Within the network, the fluctuations of the instantaneous bit rate due to jitter are absorbed by buffers appropriate for regular CBR connections subject to similar jitter.

4.2 Connection Establishment

Data transmission in ATM networks is connection-oriented. This means that a virtual connection has to be established from the source to destination before any data can be transferred. Connection establishment involves the following important steps, among others:

[2]nominal peak and burst peak rates in the case of a *type-2* source

1. finding a route from the source to destination and assigning a virtual path identifier (VPI) and a virtual channel identifier (VCI) to the connection at each switch in the connection's path

2. allocating bandwidth and buffers to the connection to satisfy its quality of service requirements, and

3. setting up the routing table of the dedicated cell switching hardware at each node.

Several fast approaches for step 1 are discussed in Section 3.2 above. Various proposals for step 2 are given in Sections 3.4.2 and 3.4.3. Finally, step 3 is hardware dependent but will be addressed in Section 4.5 below. This section will briefly describe a simple proposal for step 2 that will be covered in detail in Chapter 5 below. Chapters 6 and 7 cover additional proposals for step 2.

A simple burst-oriented bandwidth reservation scheme for bursty connections has been developed. The proposed scheme groups TLCs into classes according to their peak rates (but the mean rates may vary among the connections in a class). For each class at each link, the scheme allocates a number of *virtual servers* so that the burst blocking probability (BBP) for each connection in the class is within the guaranteed value. A *virtual server* is simply a portion of the link bandwidth equal to the peak rate of the connections in the class. Assuming that the burst arrivals are Poisson, the set of virtual servers can be modeled as an $M/G/k/k$ queueing system. Given the appropriate traffic parameters from each connection in the class, an upper bound for the BBP for the class can be obtained using *Erlang's loss formula* (given in Section 5.1 below).

Therefore, at connection establishment, each switch in the path will attempt to incorporate the new connection into an appropriate class of preexisting connections. The expected BBP after adding the new connection is determined to see if

an additional virtual server is required for the class to maintain the BBP at its acceptable level. If an additional virtual server is required, then additional bandwidth is allocated to the class (not the connection—bandwidth is reserved for the connections on a burst-by-burst basis) if it is available; otherwise the connection request is rejected. More details on the bookkeeping involved can be found in Chapter 5 below.

After accepting a new connection, a routing table entry must be provided for it at each switch. To use the burst-oriented approach, it is required that the switching hardware provide a method for turning a connection on and off as bursts come and go. This is covered in Section 4.3 below.

4.3 Eager Transmission Detection and Handling

Inside the network, the switches need a way of detecting eagerly transmitted bursts in order to allocate bandwidth for the bursts as they arrive at the switch. One approach is to identify the head and tail cells with a special payload type identifier (PTI). The switches intercept and process the cells with the special PTI. This way, the head and tail cells can be sent in-band with the same cell headers as the data cells. Another approach is to send the head and tail cells on a separate signaling band, i.e., with a different virtual channel address from the data cells. A critical problem with this approach is the timing—it has to be ensured that the head cell always precedes the data cells by a sufficient margin to give the switches enough time to activate the connection before the data cells arrive. In addition, the tail cell must not be allowed to overtake any of the data cells.

A useful feature for reducing the effects of the head cell processing delays is to use a limited form of multicasting similar to the *selective copy* technique described in [17]. This will allow the head cell to be forwarded to the next node downstream at the same time it is automatically delivered to the switch control processor (SCP) at a

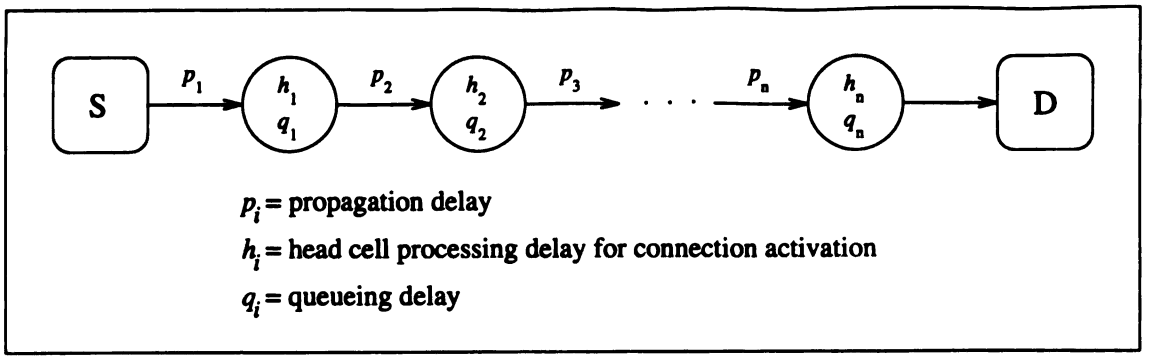

Figure 4.2: Required head cell lead times. Without multicasting, the time it will take for the head cell to reserve the bandwidth at the last link, after it is transmitted at source S, is given by $\sum_{i=1}^{n}(p_i + h_i)$ plus the queueing delays. With multicasting, this reduces to $\sum_{i=1}^{n} p_i$ plus the queueing delays. In contrast, the first data cell may take only $\sum_{i=1}^{n} p_i$ to reach the last switch. Note that if each h_i is less than the cell switching time, then the queueing delays are not a factor in the multicasting case.

node. General multicasting may also be used to simultaneously forward the head cells downstream and to the SCP. However, the head cells will have to be transmitted on a separate virtual channel/path from the data cells. Without any of the multicasting features above, the head cell processing delays at the nodes will accumulate and the lead time for the head cells has to be adjusted accordingly. This is illustrated in Figure 4.2.

A required feature to allow the safe introduction of eager transmission into the network is a method of dropping the cells of unadmitted bursts. These unadmitted bursts may be either bursts for which no bandwidth is available or bursts whose head cells are corrupted or lost. These bursts must be prevented from consuming bandwidth and buffers that have not been reserved for them. One straightforward approach is to use an *active/inactive* bit flag in the switch routing table.[3] For an *activated* TLC (i.e., with bandwidth currently reserved), the bit is set to '1', otherwise it is set to '0'. The switch checks this flag whenever it consults the routing table to forward a cell. However, this approach is not sufficient for logical connections routed

[3]The Fore Systems ASX-100 switch [25] has a "valid route" bit in its routing table that could serve this purpose.

by virtual path. A solution for virtual path connections is presented in Section 4.5.

4.4 In-Band Burst Control Cells

In this section, we discuss in more detail the function and handling of the in-band burst control cells used by this burst-oriented framework. There are two basic types of such cells, the *head cell* and the *tail cell*.

The *head cell* is prepended at the beginning of each burst to activate the connection, i.e., to turn on the routing table entry of the connection if bandwidth is available for it at that time. The head cell contains the following:

1. special PTI to identify the cell as a burst control cell
2. logical connection identifier (i.e., incoming port, VPI, and VCI)
3. control cell type (head cell)
4. connection type (e.g., thin logical connection)
5. burst rate (connection class), and
6. estimated burst duration.

The first two items above are located in the cell header, while the rest are stored in the cell payload. The third item is stored in the cell payload so as not to use up precious bits in the cell header. The fourth and fifth items are not necessary but may speed up table searches. The last item, which is optional, is useful for setting appropriate timeout durations.

When a switch identifies a head cell, it checks to see if the associated connection is activated. If not, then it checks if bandwidth is available for the burst.[4] If it is,

[4]In the case of the reservation scheme presented in Chapter 5 below, this is a matter of determining if the class to which the connection belongs has a currently unused *virtual server*.

then the switch activates the connection and reserves the bandwidth for it. Once the connection is activated, the data cells following the head cell will be switched properly. If the head cell is lost or corrupted, or if bandwidth is not available for an arriving burst, then the connection will not be activated and the data cells will be dropped and will not use up resources reserved for other connections.

The *tail cell* is appended to the end of each burst to deactivate the connection. It contains the same information as the head cell except for a different control cell type. When a switch identifies a tail cell, it checks to see if the connection is activated. It it is, then the switch deactivates it and deallocates the bandwidth temporarily allocated to it. If the tail cell is lost or corrupted, then the connection will not be deactivated and will continue to hold resources it is not using, possibly causing bursts from other connections to block unnecessarily. To avoid this, a timeout mechanism has to be used to deactivate connections automatically.

The timer for a connection is set when the connection is activated and is reset when the connection is deactivated. The duration is set according to information in the head cell payload or according to connection or switch parameters. In any case, an additional burst control cell may be used to extend the timer in the middle of a burst. We will call that cell a *refresh cell.*

A *refresh cell* is similar to a head cell except that it does not activate a deactivated connection. It only adjusts the timer associated with the connection. Therefore, if the head cell is not successful in acquiring bandwidth for a burst, then the whole burst up to the tail cell will be dropped even though the burst contains refresh cells. A refresh cell contains the same information as the head cell except for a different control cell type.

Finally, if the timer for a connection fires, then the switch assumes that a tail cell has been lost. It then deactivates the connection and deallocates its temporary bandwidth reservation.

4.5 Thin Logical Connections in a Virtual Path

The handling of a TLC at switches where it is routed by virtual channel, i.e., by looking at its VPI and VCI header fields, is straightforward since each connection will have a separate routing table entry. The mechanisms described in Section 4.3 and Section 4.4 above are sufficient.

This section deals with how thin logical connections are to be handled when they are bundled in a nonterminating virtual path in a switch. A nonterminating virtual path is a virtual path (VP) in which the cells are routed by VPI alone—the VCIs are ignored in the switching process. The main problem that has to be solved is that of properly dropping cells from an eager burst that is rejected or whose head cell is lost. Since the cells in the VP are routed via the VPI alone (i.e., there is only one routing table entry for all of them) and since each virtual channel (VC) in the VP has to be allowed to transmit bursts independently of the other VCs in the VP, the switch needs a way to distinguish between cells with reserved bandwidth and cells without reserved bandwidth in the VP.

It is proposed that, in addition to the routing table, another table be provided to contain bitmaps of the activated VCs in the VPs. Virtual paths with thin logical connections will have an entry in this table. The entry is a bit vector indicating which VCs in the VP are currently activated. For convenience, the first entry in this table may be a special bit vector containing all '1's that will be used for all virtual paths not carrying any burst-oriented connections. The other bit vectors will be allocated to the VPs containing thin logical connections, with one bit for each VCI.[5]

When the switch receives a cell belonging to a nonterminal VP, it will have to check the bit vector for the VP to see if the VC is activated. An activated VC will

[5]To support all possible VCI values, a bit vector length of 2^{16} is necessary. However, it is straight forward to allow only a subrange of possible burst-oriented VCIs to decrease the length of the bit vectors.

have a '1' in its position and a deactivated VC will have a '0'. This is illustrated in Figure 4.3. The size of the VC bitmap table will be the product of the maximum number of VCs per burst-oriented VP and the maximum number of such VPs to be concurrently supported.

In summary, it is proposed that the routing table that a switch uses to determine the output VPI label of cells be supplemented with virtual channel bitmap table. This table of bit vectors is used to distinguish between activated and deactivated bursty virtual channels in a virtual path. Cells for activated virtual channels are to be forwarded and cells for deactivated virtual channels are to be dropped.

4.6 Fat Logical Connections

In this section, a way of supporting fat logical connections using the mechanisms already described above is presented. First, we assume that we have at our disposal a type of continuous bit rate (CBR) connection that supports renegotiation. This type of connection has bandwidth deterministically allocated to it that it can fully utilize during its lifetime. Furthermore, during its lifetime, the connection is allowed to negotiate with the network long-term stepwise changes in its bandwidth allocation. The renegotiation process is *patient* in the sense that the connection is required to receive an acknowledgment from the network before it can start using any additional bandwidth requested. The *FRP/DT* proposal in [13] is an example of a scheme that supports this type of connection.

We reduce the problem of handling fat logical connections with eager bursts to the problem of handling thin logical connections with eager bursts by using *tandem logical connections*. We use two virtual channel connections on the same virtual path to carry the traffic for a single fat logical connection. One of the virtual channels is a CBR connection with renegotiation as described above, and the other is a TLC.

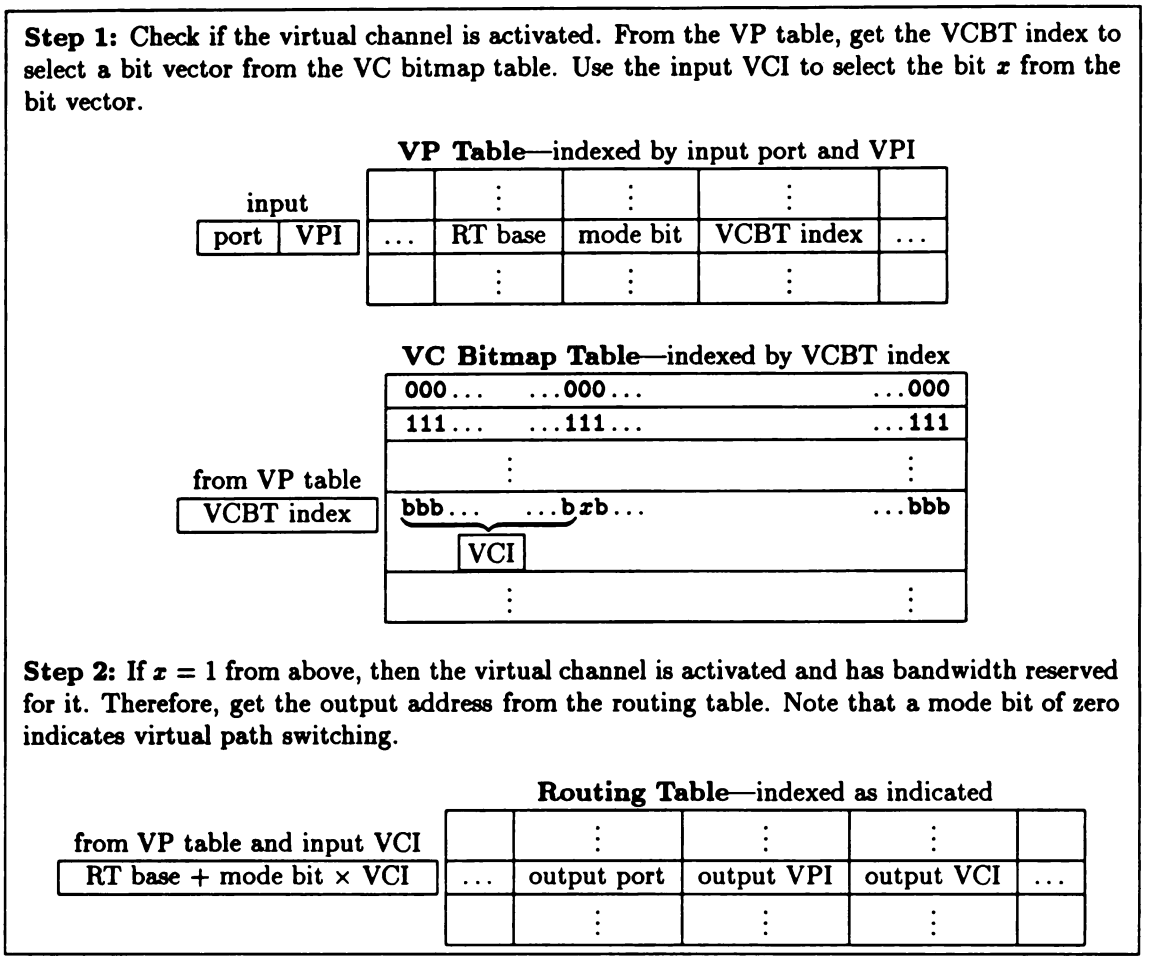

Figure 4.3: A cell routing algorithm that is safe for eager transmission in VPs. The VP table and routing table, except for the VCBT index, are modeled after the Fore Systems ASX-100 switch architecture. However, any ATM switch will need a table to contain the output VPI for each established VP. That same table may be modified to contain the VCBT index. In fact, the VCBT index could be stored in the output VCI field, which is not used for nonterminating VPs.

Putting the two connections in the same virtual path means that at each switch along the way, the two connections will have the same incoming port and VPI and same outgoing port and VPI. However, the two components will have different VCIs.

We will assume that cells in a virtual path, even if they belong to different virtual channels, are not allowed to overtake each other.[6] This way, there will be no unwanted overtaking between the cells sent with the TLC VCI and the cells sent with the CBR VCI. If it is necessary for the burst to persist (i.e., the connection requires long-term bandwidth increase rather than a short temporary increase for a short burst), then the CBR renegotiation request and the eager burst can be sent at the same time. This way, when the renegotiation is approved, all the subsequent traffic can be sent on the upgraded CBR component. This subsequent traffic could include the still-to-be-sent portion of the eager burst. After transferring the latter portion of an eager burst to the CBR component, then the TLC component can be used for a new burst.

In general, when a fat logical connection initiates a burst with a head cell, then it has at its disposal an amount of bandwidth equal to the sum of the CBR bandwidth and the TLC peak rate. Also, it has two VCIs to work with. It can send data cells using the two VCIs with the following restrictions:

1. The total cell traffic sent using the two VCIs must not exceed the total bandwidth of the two components.

2. The cell traffic sent using the CBR VCI must not exceed the CBR bandwidth because the head cell for the TLC portion might be rejected.

3. The cell traffic sent using the TLC VCI can be more than the TLC peak rate but less than the total bandwidth of the two components. The TLC VCI is used with the understanding that if the head cell is rejected, then all the cells

[6] Alternatively, we assume that the ATM network supports a special type of virtual path with the explicit feature that all cells in that type of virtual path are kept in order.

sent with that VCI will be dropped even though a portion of the burst may be using bandwidth from the CBR portion.

Various approaches for splitting the FLC data cells among the two components are possible. Some of these are illustrated in Figure 4.4.

Finally, if thin logical connections with eager bursts can be bundled in virtual paths, then fat logical connections with eager bursts, implemented as described above, can be bundled in virtual paths as well.

4.7 Summary

A traffic control framework for ATM networks has been presented. The framework is designed to handle bursty data traffic in a burst-by-burst fashion. That is, a burst of logically related data cells is treated as a unit by the protocol framework. Bandwidth is allocated and deallocated to bursts as they come and go by dynamic activation and deactivation of virtual channels at the individual switches in the connection path. Mechanisms are provided to allow sources to transmit bursts without requiring that bandwidth be reserved throughout the whole connection path. Two kinds of bursty connections are supported: one with zero bandwidth between bursts and another with non-zero bandwidth between bursts. Mechanisms are provided to allow such connections to be bundled into virtual paths.

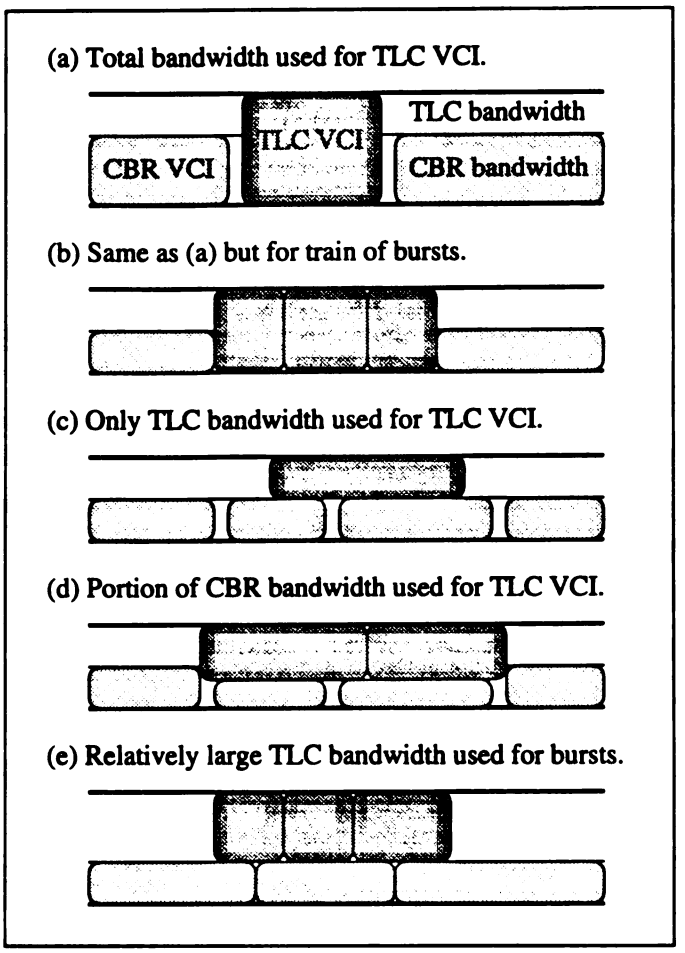

Figure 4.4: Bursts in tandem logical connections. The total bandwidth consists of TLC and CBR components. Shaded bursts are sent eagerly using the TLC VCI; these may be dropped by the network. Unshaded bursts are sent using the CBR VCI, which has bandwidth preallocated to it. Note that the eager bursts in (b), (d), and (e) are correlated. This phenomenon has to be taken into account in the traffic description.

Chapter 5

A Simple CAC Scheme Based on the Erlang Loss System Model

Recall that in ATM networks, connections that carry bursty traffic are statistically multiplexed in order to increase the number of concurrent connections and the bandwidth utilization. In order to prevent the deterioration of the network performance, a call admission control (CAC) strategy limits the number of concurrent connections so that a specific quality of service is guaranteed. For each new connection request, the CAC strategy determines if it can accept the connection and continue to meet the service guarantees. Research has been conducted on the problem of determining the bandwidth and/or buffer requirements of bursty connections for this purpose. Many of the proposed approaches are based on using the cell loss rate (CLR) as the measure of the quality of service (see Section 3.4.2 above).

A new approach for allocating bandwidth to highly bursty connections in ATM networks is presented in this chapter. Following the lead of Hui [35] (see Section 3.4.3 above), it is proposed that bandwidth be requested and allocated for bursts on a burst-by-burst basis. Each burst is to be preceded by a bandwidth reservation request and then transmitted. If the requested bandwidth is available, then the burst is forwarded;

otherwise the burst is dropped. The new scheme is based on modeling the burst-by-burst traffic control system as an $M/G/k/k$ queueing system. On this model, *Erlang's loss formula*[1] [68, 78] is applied to guarantee a maximum burst blocking probability (BBP). This approach exhibits several desirable characteristics.

First, the determination of the availability of bandwidth for a new connection can be performed within a few microseconds per link. This is faster than all other published proposals known to the author. In addition, the determination of whether a particular burst can be accommodated is straightforward, such that its evaluation can be implemented easily in hardware. Furthermore, the model does not impose a constraint on the burst duration distribution other than its average length must be specified. The scheme is burst-oriented in the sense that bursts are treated as units. Bursts will be delivered or dropped completely—the only isolated cell loss that will occur will be the same as that experienced by a generic continuous bit rate (CBR) connection. Moreover, bursts are not buffered as such—the only cell buffers required are those that are allocated to any CBR connection (to handle jitter). Finally, the burst-oriented approach allows the application of *traffic policing* (UPC), priorities, and preemption at the burst level.

Performance studies to compare the new approach with the other proposals cited above have been performed. From the results, it appears that the new approach achieves comparable maximum utilization levels of within a few percent over a wide range of input traffic parameters. The results are presented in Section 5.5.

Section 5.1 describes the proposed service model for the connection types described in Section 4.1 above. Section 5.2 derives the basic BBP computation method for a set of homogeneous sources from *Erlang's loss formula.* The method is then extended

[1]The formula, given in Section 5.1 below, is widely known to apply to the $M/M/k/k$ system. However, it has also been shown to apply to the more general $M/G/k/k$ system. See [78] for more information.

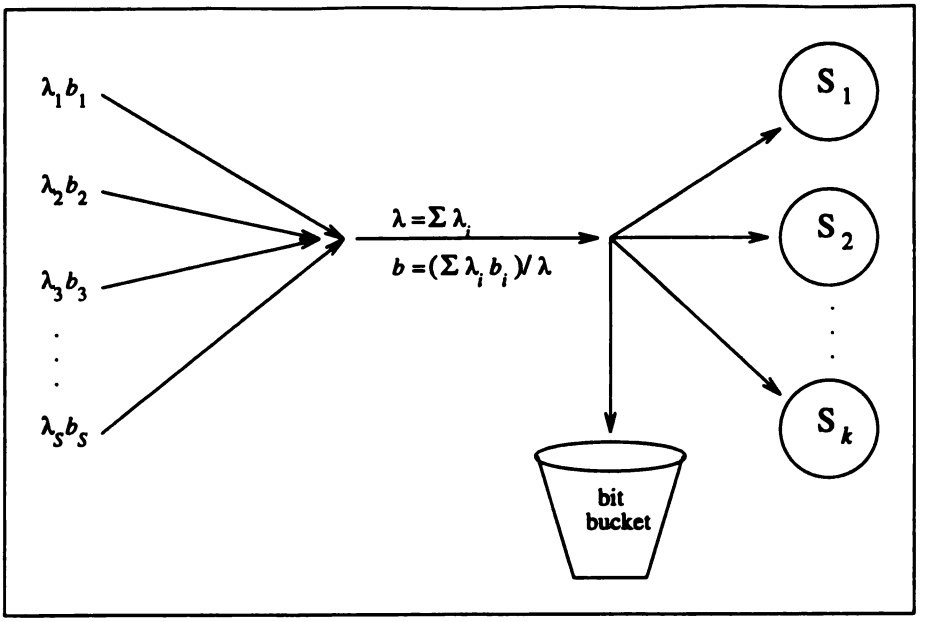

Figure 5.1: $M/G/k/k$ modeling of a class of connections. All S sources at the left have the same peak rate, which is the bandwidth of each of the k servers at the right, where $k \leq S$. The burst arrivals from the sources are Poisson processes with rates λ_i and the burst durations are given by the means b_i. Arriving bursts for which no server is available are dropped in the bit bucket.

to the computation of traffic bandwidth requirements of a class of connections in Section 5.3. The multi-class generalization of the Erlang loss system is discussed briefly in Section 5.4. The performance of the single-class system, including discussion of the impact of non-Poisson burst arrivals, is covered in Section 5.5. Finally, a summary for the chapter is given in Section 5.6.

5.1 The M/G/k/k Service Model

As mentioned briefly in Section 4.2 above, the model that is being considered for the burst-by-burst service scheme is the $M/G/k/k$ queueing model that is illustrated in Figure 5.1. Recall that in the notation $M/G/k/k$, M indicates Markovian interarrival intervals (or a Poisson arrival process) and G indicates a general service time distribution. The constant k is the number of identical servers which is also the maximum

capacity of the bufferless system. The advantages of the model are:

- It has no restrictive assumption on the service time distribution. In particular, the burst durations do not have to be exponentially distributed as is the case in [13, 20, 27, 28, 40].

- It has been shown that for this system, for any service time distribution, the probability that i servers are being used is given by *Erlang's loss formula* [78]

$$P_i = \frac{(\lambda E[X])^i / i!}{\sum_{j=0}^{k} (\lambda E[X])^j / j!}$$

  where λ is the arrival rate and $E[X]$ is the mean service (or holding) time.

The limitations of using this model are:

- It requires the grouping of connections into classes to achieve utilization gains. However, the peak rate is the only common characteristic required for a class. In this scheme, connections in the same class are allowed to have different mean rates. This is in contrast to the other proposals that require connections to have the same peak rate and mean rate to be combined into a class [13, 20, 27, 40, 76].

- The burst arrivals are assumed to be Poisson. Real traffic may have undesirably correlated arrivals. However, the effects of these non-Poisson behavior may be minimized by appropriate *traffic shaping* at the burst level. Note that the Poisson assumption is also made in many of the previous proposals [13, 20, 27, 28, 35, 40]. Also, though the proposals of Saito [70] and Suzuki *et al.* [76] do not use this assumption, they are based on guaranteeing the CLR. Furthermore, Saito's proposal has a restriction on the maximum and average rates over a given time interval. Finally, the proposal of Turner [82] does not require the Poisson assumption; however, its computational requirements are significantly greater

and (for other reasons) is only proposed for traffic whose peak rates are at least an order of 1% to 2% of the link rate.

5.2 Basic BBP Computation Method

Recall that a burst is a sequence of cells sent consecutively at a certain peak rate. The *head cell* at the beginning of a burst is treated by each node as a reservation request for that burst. If the node has the requested bandwidth, then the whole burst is switched to the next node via appropriate manipulation of the switch routing table. Otherwise, the burst is effectively dropped. The probability that the latter happens is called the burst blocking probability (BBP).

Using the $M/G/k/k$ model, we first show how the BBP is determined for S_0 identical sources. Let the arrival rate from each source be λ_0, the mean burst duration be b, and the peak rate be R. Then, the aggregate Poisson arrival rate is given by $\lambda = S_0\lambda_0$ (by property of Poisson processes) and the aggregate mean service time is simply b (ignoring overhead). Now, conceptually divide the available capacity C into discrete *virtual servers* of capacity equal to the peak rate R, giving a number of discrete servers equal to $k = \lfloor C/R \rfloor$. The probability that the system is full, given by Erlang's loss formula, is

$$P_k = \frac{(\lambda b)^k/k!}{\sum_{i=0}^{k}(\lambda b)^i/i!} \tag{5.1}$$

This is an upper bound of the BBP. A slightly tighter bound for the BBP can be obtained by taking into account the fact that a burst from a source does not arrive while that source still has a burst in the system.[2] That is, a burst competes only with the bursts from other sources. Therefore, for any particular source, its BBP is equal to the probability that the system is filled with bursts from the other sources. This

[2]Recall that the bursts are not queued.

can be obtained by replacing λ with $\bar{\lambda} = (S_0 - 1)\lambda_0$ in Equation 5.1 above. Note that for large S_0 (or small R), this will make little difference.

This technique of combining the Poisson arrival processes can be extended to a set of non-identical sources as long as the sources have the same peak rate. Therefore, it is proposed that the bursty sources be classified according to their peak rates. One advantage of doing this is that the peak rate is relatively easy to enforce at the entrance to the network (see Section 3.4.1 above). The peak rates supported can be optimized for widely used applications. For example, 64 Kbps can support voice, low-speed data transfers, and fax; 2 Mbps can support color image and high-speed data transfers; and 10 Mbps can support video applications (cf. [20, 27]).

5.3 Bandwidth Requirement of a Class of Sources

The total capacity available for bursty connections is divided among the different classes dynamically as connections are set up and torn down. This section shows how the BBP computation method from the previous section is extended to compute the bandwidth required by a class of bursty connections.

Let S_q indicate the number of bursty sources in class q of peak rate R_q. Let the burst arrival rates for the individual sources in the class be $\lambda_1, \lambda_2, \ldots, \lambda_{S_q}$ and their corresponding mean burst durations be $b_1, b_2, \ldots, b_{S_q}$.[3] For $i = 1, 2, \ldots, S_q$, let $\rho_i = \lambda_i(b_i + h)$, where h is the burstwise overhead (e.g., for bandwidth reservation and release). The aggregate Poisson burst arrival rate is therefore given by

$$\lambda = \sum_{i=1}^{S_q} \lambda_i$$

[3]The following relationship holds for bursty sources $B_i = 1/\rho_i = 1/(\lambda_i b_i)$ where ρ_i is the normalized load (fraction of time at peak rate) and B_i is the burstiness (peak rate to mean rate ratio) of source i. Therefore for a source i, instead of specifying both λ_i and b_i, either one of them in combination with one of ρ_i and B_i could specify the traffic.

and the aggregate mean burst serving time by

$$b = \frac{1}{\lambda} \sum_{i=1}^{S_q} \rho_i$$

Let

$$\rho = \lambda b = \sum_{i=1}^{S_q} \rho_i$$

Then, to determine the required number of servers for the class, find the minimum k such that $P_k \leq \text{BBP}$ where

$$P_k = \frac{\rho^k / k!}{\sum_{i=0}^{k} \rho^i / i!} \tag{5.2}$$

The required bandwidth is therefore kR_q. Note that Equation 5.2 above overestimates the blocking probability for a single source as already explained in Section 5.1. To be more precise, we have to compute the maximum probability that bursts from only $(S_q - 1)$ of the sources occupy all k servers. Note that this will be experienced by a source j with $\rho_j = \rho_{\min} = \min\{\rho_1, \rho_2, \ldots, \rho_{S_q}\}$. Therefore, we can simply replace ρ with $\bar{\rho} = (\rho - \rho_{\min})$ in the formula above. In practice, this might be done only for small S_q, because for large S_q, the cost of finding $\rho_{\min}$ might be too great relative to the additional utilization gained.

An efficient way of finding the required number of virtual servers is needed for fast connection establishment. Note that this is not absolutely necessary for supporting eager bursts as long as we allow the preestablishment of the logical connections. However, if the same processor will be handling the connection admission and burst admission requests, then it would be imperative to have an efficient CAC that will not impede the burst admission processing.

Note that a class will require at most one additional virtual server when adding a new connection. Therefore, if the current number of virtual servers allocated to the class is k, we need only compute P_k with the additional traffic included and check if

it is still less than the desired BBP. If it is, then the current number of virtual servers is adequate. Otherwise, an additional virtual server will be required to admit the connection.

The straightforward method for directly computing P_k from ρ requires two divisions, one multiplication, and one addition per step for k steps. Instead of computing P_k, one can compute its reciprocal $(1/P_k) = r_k$ obtained recursively from

$$r_0 = 1$$

$$r_i = i(1/\rho)r_{i-1} + 1$$

This approach requires only two multiplications and one addition per step for k steps to compute $(1/P_k)$ from $(1/\rho)$. Furthermore, it is computationally more accurate.

A better method is to use a table lookup to find the required number of virtual servers. For a fixed BBP, precompute the maximum possible ρ for each value of k and store the values in a table,[4] e.g., $\rho_{\max}[1{:}10000]$. Then, given a particular ρ, simply search the table for the minimum $\rho_{\max}[k]$ that is greater than or equal to the given ρ. The index k is the required number of servers. In practice, the CAC can simply keep the current $\rho_{\max}[k]$ in a variable. Then, when adding a new connection, it only has to check if the resulting ρ exceeds the current $\rho_{\max}[k]$.

In summary, the CAC procedure is as follows: Compute the new bandwidth requirements of the class to which the call belongs if it is added. This amounts to determining if the class needs an additional virtual server. Then check if the additional bandwidth required is available from the residual bandwidth not allocated to any class. The burst admission procedure is simply to check if the bandwidth required is available from the bandwidth allocated to the class but not currently reserved for any burst. These procedures are sketched in Figure 5.2.

[4] A sparser table containing the utilizations ρ/k for staggered k can be used if the large table size is a problem or if large values of k are anticipated. Values of ρ/k not represented in the table can be obtained via linear interpolation.

Relevant variables

For switch:	`MaxLdTbl`	table of maximum loads ($\rho_{\max}[1{:}10000]$)
	`BurstOv`	burstwise overhead (h)
For link:	`BwAvail`	bandwidth not allocated to any class
For class:	`BwServ`	bandwidth of virtual server (R)
	`NumServ`	number of virtual servers (k)
	`ServInUse`	number of servers in use by bursts
	`TotNormLd`	total normalized load for class ($\sum_i \rho_i$)
	`MaxAdmLd`	maximum admissible load ($\rho_{\max}[k]$)
For call:	`ArrRate`	burst arrival rate (λ_c)
	`MeanDur`	mean burst duration (b_c)
	`NormLd`	normalized load

Portions of interest from admission procedures

A. ALLOCATE BANDWIDTH FOR CALL:

```
NormLd = ArrRate × (MeanDur + BurstOv)
IF TotNormLd + NormLd ≤ MaxAdmLd
THEN TotNormLd = TotNormLd + NormLd
        ACCEPT CALL
ELSE  IF BwAvail ≥ BwServ
        THEN BwAvail = BwAvail − BwServ
                NumServ = NumServ + 1
                MaxAdmLd = MaxLdTbl[NumServ]
                TotNormLd = TotNormLd + NormLd
                ACCEPT CALL
        ELSE  REJECT CALL
```

B. RESERVE BANDWIDTH FOR BURST:

```
IF ServInUse < NumServ
THEN ServInUse = ServInUse + 1
        ACCEPT BURST
ELSE  REJECT BURST
```

Figure 5.2: Bandwidth allocation procedures. Procedure A is a portion of the call admission procedure that checks the bandwidth availability for a new connection. It is based on the table lookup alternative and ignores the subtleties with $\rho_{\min}$ described in the text. Procedure B is a portion of the corresponding burst admission procedure.

5.4 Multi-Class Generalization of the Erlang Loss System

A number of researchers have investigated the generalization of the Erlang loss system that allowed resource sharing among different classes of customers [2, 6, 41, 64]. Like the classical Erlang loss system, this system assumes that the arrival process is Poisson and that an arrival that cannot be allocated its required resources is blocked (i.e., dropped from the system). Furthermore, the service time distribution may or may not be exponential; only its mean has to be known. However, the multi-class system allows resource sharing among different classes of customers with different resource demands. The multi-class model can be summarized as follows.

- There are Q independent classes for a constant but arbitrary Q.
- Class q has a Poisson arrival rate of λ_q.
- Class q customers have a "spatial" resource demand of R_q integral resource units.
- Class q customers have a mean service time of B_q time units.

In short, each class has its own arrival rate, resource demand, and mean service time.

All the customers from all classes are to share the same resource pool according to the resource sharing policy. A variety of resource sharing approaches are possible. For example, in the *full sharing* approach, all customers have full access to the resources such that no customer is blocked as long as its resource demand is available when it arrives. In the *dedicated access* approach, the resources are partitioned among the classes, effectively eliminating dynamic sharing of resources among the different classes. In between these two approaches are the *partial sharing* approaches in which some of the resources are dedicated and the rest are shared. The partial sharing

approaches may also be viewed as limiting the access of some classes, and may be used as a priority scheme to control the blocking probabilities of the various classes.

The general formulas for the state probability distributions and the blocking probabilities are given in [2, 6, 41, 64]. However, in their natural forms or their multidimensional recursion equivalents, these solutions are not computationally practical for reasonably large systems. Only in the full sharing approach is there a practically useful one-dimensional recursion [41]. However, various approximations and numerical techniques are available. See [2, 6, 41, 64] for more information.

5.5 Performance Evaluation

In this section, we will evaluate the burst-oriented approach for handling bursty traffic in ATM networks. In particular, we will study the utilizations achievable by the call admission scheme based on the Erlang loss system given in this chapter. We will also compare the eager transmission approach to patient approaches over a wide range of network scenarios. In addition, we will compare the scheme given in this chapter to another scheme based on a more general traffic model. Finally, simulation results are given that validate the effectivity of the scheme in controlling the BBP of the system.

5.5.1 General Results

Here, we briefly examine the maximum utilizations achievable using the admission scheme presented in Section 5.3 for the thin logical connections (TLCs) presented in Section 4.1. Recall that in the given scheme, TLCs are grouped into classes according to their peak rates. Bandwidth is allocated to whole classes by way of *virtual server* units. We assume that the peak rates are small relative to the total available bandwidth. For example, classes of 64 Kbps, 2 Mbps, and 10 Mbps can support a wide variety of applications (cf. [20, 27]). In comparison, ATM link bandwidths are in the

range of hundreds of Mbps to several Gbps.

Figure 5.3 plots the maximum achievable utilizations for a wide range of capacities for different choices of the burst blocking probability (BBP). The utilizations were obtained using Equation (5.2) by computing, for each k, the maximum aggregate ρ for which P_k is no greater than the BBP. The maximum ρ divided by k gives the maximum utilization for the k servers. Note that these maximum utilizations apply to all cases where there are more connections than virtual servers in a class.[5] Furthermore, if we ignore the burstwise overhead or consider it as part of the aggregate ρ, then the utilizations do not depend on the burst durations or the burstiness (peak to mean rate) ratios of the connections; they only depend on the aggregate ρ.

The results in Figure 5.3 clearly show that the burst-oriented reservation scheme with eager transmission is able to achieve moderately high utilizations for a wide range of bursty traffic. Note that to obtain the corresponding statistical gains, simply multiply the utilization value by the burstiness ratio. For example, if the utilization is 50% and the burstiness is 20, then the statistical gain is 10; this means that with the same bandwidth, the network is able to support 10 times as many connections with the burst-oriented scheme than with deterministic (peak-rate) allocation.

5.5.2 Network Scenarios

In this section, we analyze the difference in the achievable utilizations between the eager and patient transmission schemes in different network scenarios. Recall that the eager transmission scheme, unlike the patient transmission scheme, does not wait for the required bandwidth to be reserved (and acknowledged) before transmitting a burst. Therefore, to compare the performance of eager transmission with the patient

[5] If the number of virtual servers is equal to the number of connections, then the maximum possible utilization with no burst blocking is 100%, which is the case when all the connections are active all of the time.

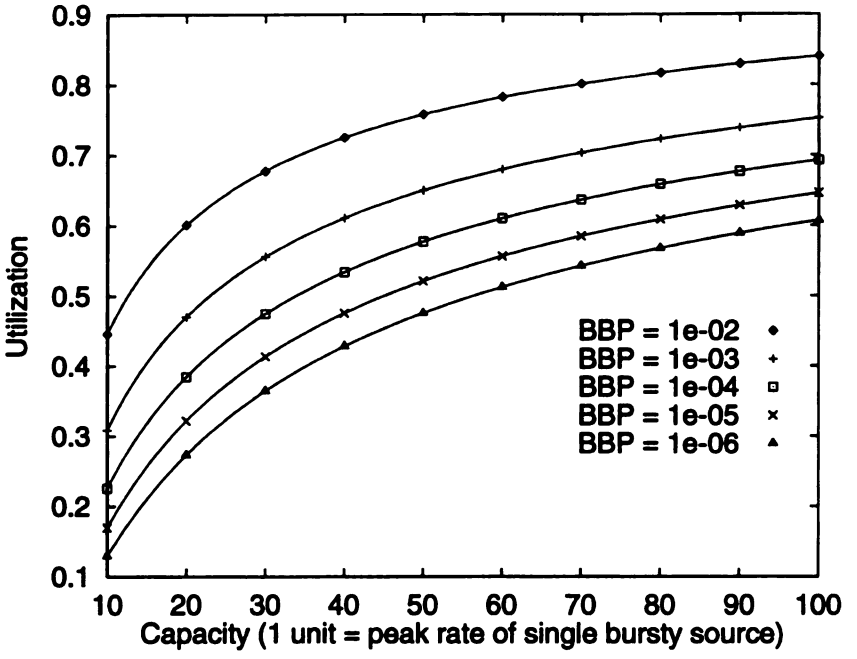

Figure 5.3: Maximum utilizations for the burst-oriented reservation scheme. The **x-axis** is the capacity available to a class of bursty connections of the same peak rate. **The** data points are the maximum achievable utilizations for the different choices of **BBP**. These maximum bounds apply when the number of connections exceeds the **number** of capacity units (*virtual servers*) in a class.

Table 5.1: Parameters and settings for the network scenarios.

Parameter	Setting
traffic burstiness	10
available bandwidth	100 Mbps
BBP (per hop)	10^{-4}
number of hops	LAN: 4 switches MAN: 6 switches WAN: 10 switches
total link distance	LAN: 5 miles (8 km) MAN: 30 miles (48 km) WAN: 3,000 miles (4,800 km)
propagation speed (speed of light)	186,000 miles/s (300,000 km/s)
average switch queueing delay	0.1 ms/switch
average connection setup delay and average connection tear down delay (excluding bandwidth reservation/release)	0.5 ms/switch
average bandwidth reservation delay and average bandwidth release delay (with preestablished virtual channel)	0.01 ms/switch

approaches, we simply have to increase the overhead portion of the mean service time in the formula to include the round trip delay and any additional overhead. We analyze the utilizations for three networks scenarios and three bandwidth classes. The network scenarios are the local area (LAN), metropolitan area (MAN), and wide area (WAN) networks. The bandwidth classes are 64 Kbps, 2 Mbps, and 10 Mbps. The various parameters used in the performance studies are summarized in Table 5.1. The maximum achievable utilizations are plotted in Figures 5.4 to 5.6, where each curve represents one of the following approaches:

- **eager**—thin logical connections with eager transmission.
- **fast**—fast reservation: thin logical connections with patient transmission.
- **reconn**—reconnection without preestablished logical connections. Here, a connection is established and disconnected for each burst. The offered load is limited so that the BBP is maintained.

- **noGOS**—no guarantee of service. Same as **reconn** but with an unlimited offered load and no guaranteed BBP. In this case, the only limiting factor is the connection setup and tear down overhead for each burst. For theoretical comparison only.

- **PVCs**—permanent virtual circuits with permanently reserved bandwidth. For comparison.

Note that all the above approaches except **noGOS** maintains the BBP.

In Figure 5.4, we see that for low bandwidth classes with many connections, any burst-oriented approach (**eager**, **fast**, or **reconn**) is able to achieve high bandwidth utilizations. However, in the WAN scenario (Figure 5.4(c)), the eager approach has a slight advantage. This is because of the greater round trip delay that is effectively an additional overhead for the patient approaches (**fast** and **reconn**).

For the medium-sized class of 2 Mbps, we see a different story. First, note that since the total class bandwidth is set to 100 Mbps in this study,[6] there are fewer virtual servers available for this class compared to the 64 Kbps class. This is the reason why the utilizations are in general not as high as in the 64 Kbps class. In any case, we see in Figure 5.5, that the eager approach has a significant utilization advantage over the patient approaches. In the LAN and MAN scenarios, the advantage is significant for short mean burst lengths. However, in the WAN scenario, the advantage is significant even for long mean burst lengths. Note also that in the WAN scenario, the eager approach is better than **noGOS** for short to moderate mean burst lengths, even though **noGOS** does not guarantee any level of service (and is totally impractical because of the uncontrolled congestion).

[6] In no way does this imply that the class bandwidth allocation scheme is static. In the scheme, the bandwidth allocated to each class is increased and decreased as connections are established and torn down.

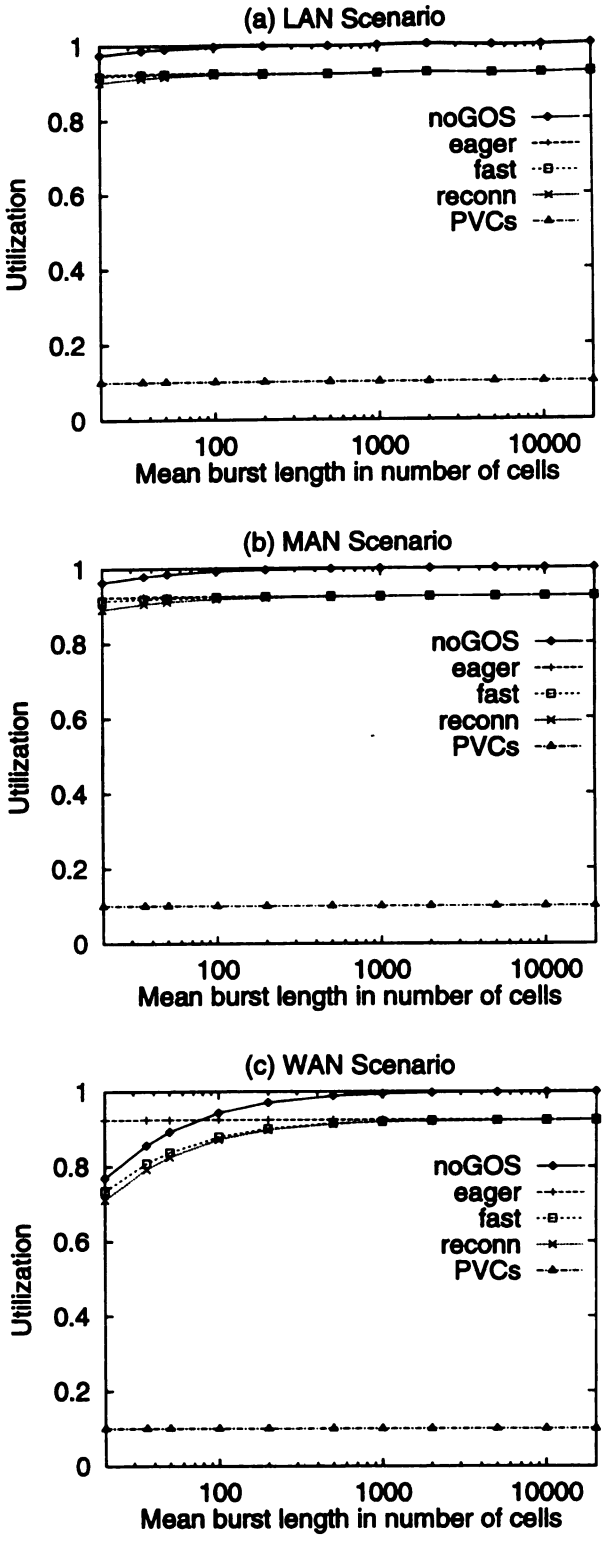

Figure 5.4: Maximum utilizations for the 64 Kbps class.

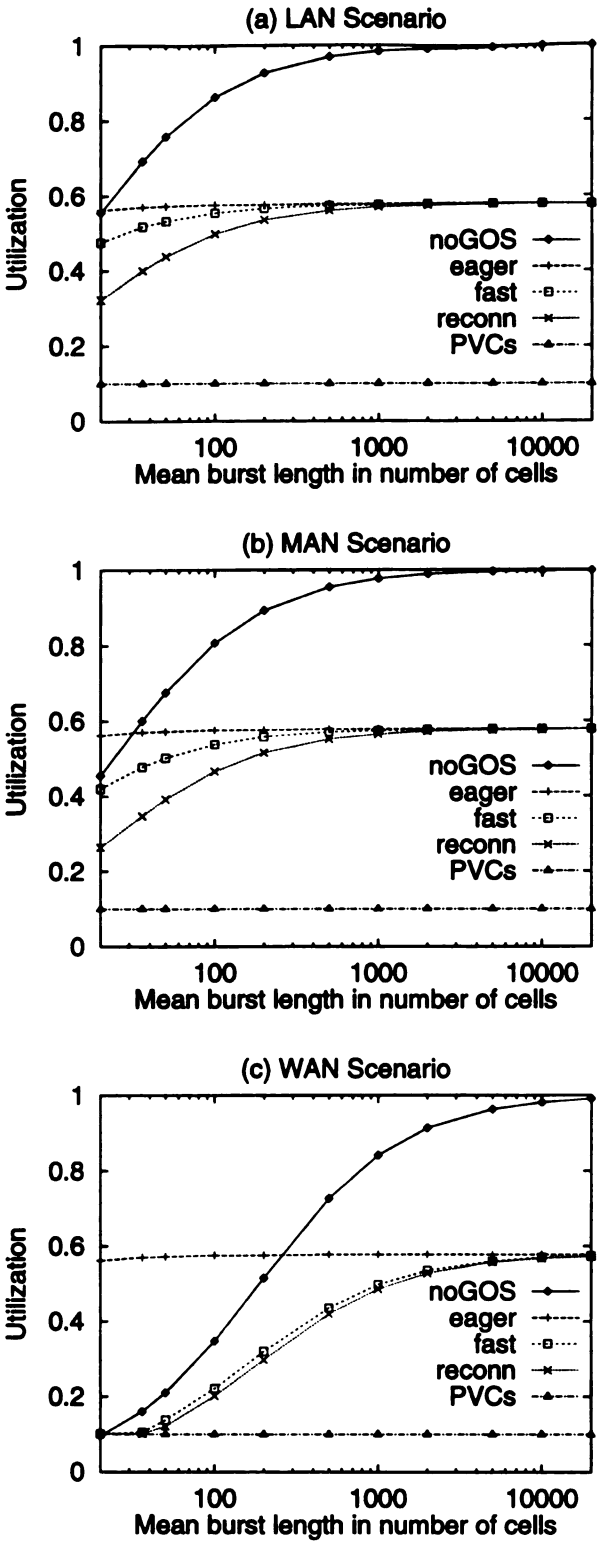

Figure 5.5: Maximum utilizations for the 2 Mbps class.

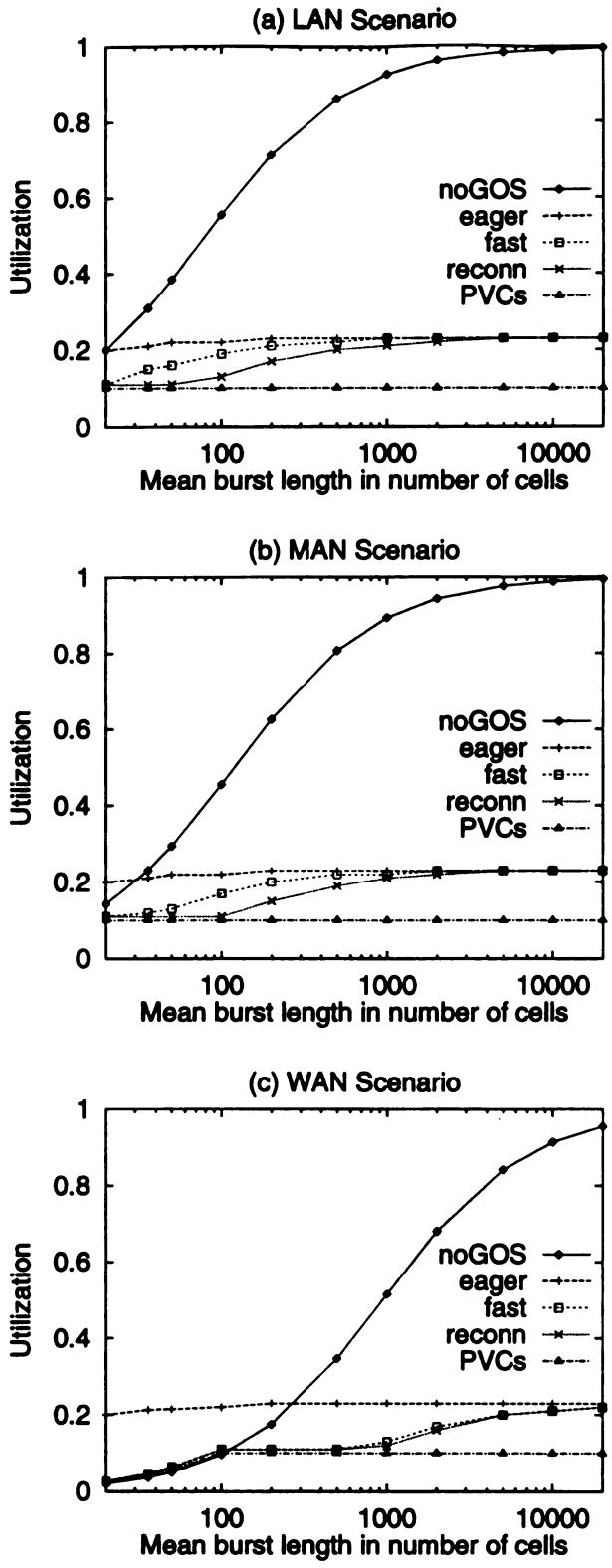

Figure 5.6: Maximum utilizations for the 10 Mbps class.

Figure 5.6 presents the results for the 10 Mbps class. Note that there are only 10 virtual servers in this case. However, for the most part the eager approach still achieves a utilization of about 23%. This is equivalent to a statistical gain of 2.3 since the burstiness is equal to 10 (2.3 is also the ratio of the utilization of **eager** to the utilization of **PVCs**). All the observations for the 2 Mbps class apply except that the range in which the eager approach is better extends to longer mean burst lengths.

Note that in the case of Figure 5.6(c), when obtaining the results for low mean burst lengths, the arrival rate was deliberately decreased so that there is only one burst on the path per source per round trip time.[7] This is to simulate a protocol (such as *stop-and-wait*) that requires an acknowledgment between bursts. The performance of the eager approach was not affected by this. However, it further increased the overhead and decreased the utilization of the patient approaches.

Overall, we see that the eager approach is better than the patient approaches in the cases where either the mean burst length is relatively short, or the round trip delay is long, or both. That the eager approach is better for shorter mean burst lengths can be explained by the fact that the overhead *per cell* incurred by waiting for the bandwidth to be reserved becomes larger when the number of cells per burst becomes fewer. As already explained, the round trip delay is additional overhead to the patient approaches, but this is not the case for the eager approach.

Furthermore, note that the performance gain of the eager approach is more significant for traffic classes of higher peak rates. This is due to the fact that the same number of cells corresponds to a shorter burst duration (i.e., holding time) when transmitted at a higher peak rate. In general, the amount of benefit of the eager approach over the patient approaches depends on the round trip delay to burst duration ratio. An analysis of this relationship was briefly presented in [13]. It is most

[7] To obtain the results without this adjustment, note that the utilizations will just bottom out at 10%.

interesting to note that the performance of the eager approach is only a little affected by the burst duration as long as the reservation overhead is small. The most important thing that affects the performance of the eager approach is the number of sources or virtual servers in the class—the more, the better.

Finally, note that the burstiness used in the studies was equal to 10, and that the gains of the eager burst-oriented approach will be even greater for higher burstiness values.

5.5.3 Comparison with a Non-Poisson Model of Sources

A concern that must be evaluated is how well does a model with an assumption of Poisson sources limit the utilization of the network safely within target BBP requirements even if the sources themselves do not follow a Poisson generation process. Although the computation of BBP and the determination of accepting new requests in the Erlang loss system are simple and efficient, it must be determined whether the utilization of the network using the Erlang loss system will exceed the amount that can be safely accepted under a more general model of burst sources. Therefore, to provide a comparison to the Erlang loss system model, we investigate a more general model that only assumes that the sources are independent. The model does not require the arrivals from the sources to be Poisson. What it requires is that the state of each source (i.e., whether it is active or inactive) is independent of states of the other sources.

In the basic form of the general model to be used in this section, we will assume that the burst sources are homogeneous. For each source, we assume that p is the fraction of time that the source is active at its peak rate, and the $1-p$ is the fraction of time that it is silent. We assume that there are k virtual servers at the network link of concern, and that each virtual server has bandwidth equal to the peak rate of

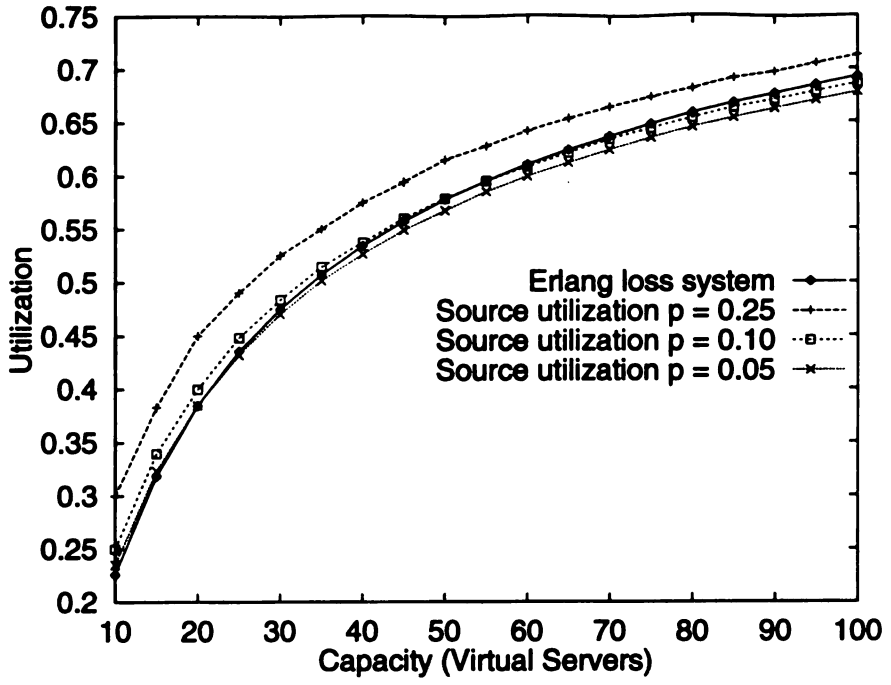

Figure 5.7: Comparison of Erlang loss system to non-Poisson model. The BBP is set to 10^{-4}.

each source. The number of sources is s.

For this general model, which does not assume any (inter)arrival process, the probability that a burst from some source will be blocked will be bounded above by the probability that all the servers are being used by bursts from the competing $s-1$ sources. Therefore, the BBP for the source is given by (cf. the *stationary approximation* for homogeneous sources used in [28])

$$\text{BBP} \leq \sum_{i=k}^{s-1} \binom{s-1}{i} p^i (1-p)^{s-1-i} \tag{5.3}$$

The offered load to the servers will be $p * s$, which is the individual source utilization of each source (normalized with respect to its peak rate) times the number of sources.

We now compare the utilizations allowable by this non-Poisson model with the utilizations allowable by the Erlang loss system. Figure 5.7 plots the achievable utilizations using the Erlang loss system model and the non-Poisson model for selected

values of the individual source utilization p. For a small number of sources, or a relatively large value of the source utilization p, the Erlang loss system model safely constrains the number of sources. For $p = 0.25$, the Erlang loss formula is safe for all numbers of servers shown. For $p = 0.1$, it appears to be completely safe up to about 55 to 60 servers. However, as the number of independent sources increases, it is possible that the aggregate load of the sources behaves more like a Poisson process, and the Erlang loss formula may be safe even at larger numbers of servers (and sources).[8] These results show that as long as the loads of the sources operate within an acceptable degree of burstiness, the simplicity and efficiency of the Erlang loss formula makes it attractive for use in a CAC strategy, even in the cases of non-Poisson sources.

5.5.4 Simulation Results

Simulations have been performed to validate the effectivity of the CAC scheme based on the Erlang loss system in controlling the BBP experienced by the admitted connections. Figure 5.8 shows the measured BBPs obtained from simulations using different types of burst source processes. The target BBP was set to 10^{-4}. The simulation runs were set up as follows.

Each run consisted of 10^7 bursts from independent sources of the same type, each with an individual source load (i.e., mean burst arrival rate multiplied by the mean burst length) equal to 0.1. The number of sources for each capacity value of interest was determined by dividing the maximum utilization allowed by the CAC scheme (i.e., from Figure 5.3) by the individual source load. Seven different types of source processes were used: Poisson process (PP), Markov-modulated PP (MMPP),

[8]Note that in the case of heterogeneous sources (i.e., same peak bandwidth but varying p) with the same aggregate average p, the general model will actually allow a higher utilization for reasonably small enough BBP values. This is the result of a demand density curve that will tend to have a higher peak and a flatter tail.

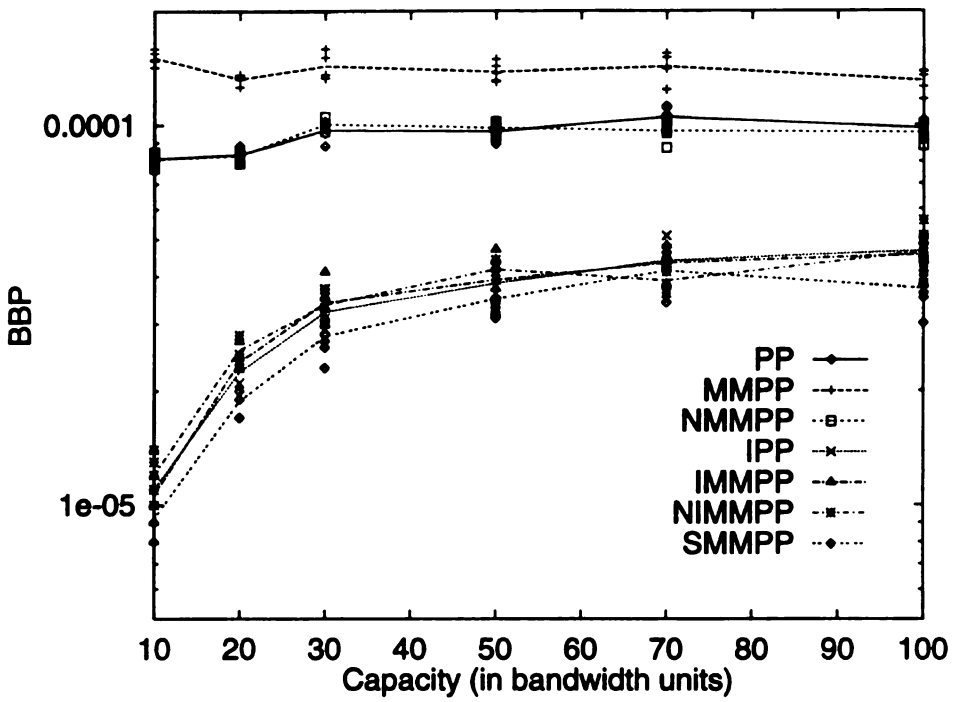

Figure 5.8: Simulation results using different burst source models. The target BBP was set to 10^{-4}. The individual points plot the observed BBPs of individual runs, and the lines connect the corresponding averages of five runs per source type. The legend acronyms indicate the source types as follows: "PP" stands for Poisson process; "MM" stands for Markov modulated; "N" stands for normalized; "I" stands for interrupted; and "S" stands for serialized. These are described in the text.

normalized MMPP (NMMPP), interrupted PP (IPP), interrupted MMPP (IMMPP), normalized IMMPP (NIMMPP), and serialized MMPP (SMMPP). The burst lengths (i.e., service times) were exponentially distributed.

The PP was simply a homogeneous Poisson process. The MMPP was a non-homogeneous Poisson process whose arrival rate was determined by a two-state continuous-time Markov chain. The IPP was a Poisson process whose arrival rate temporarily dropped to zero (i.e., got interrupted) while the process still had a burst in the system. The IMMPP was the similarly interrupted version of the MMPP. The NMMPP was an MMPP whose burst length distribution was normalized with respect to the interarrival time distribution dictated by the Markov chain so as to maintain the same offered load in the two Markov chain states. The NIMMPP was the analogously normalized version of the IMMPP. Finally, the SMMPP was an MMPP whose bursts were serialized through a single server queue before they entered the multi-server system. Note that in the simulations, these processes were used to model single sources, and that as many such independent processes were simulated as was allowed by the CAC scheme. None of the above processes were used to model the aggregate traffic from multiple sources.

The IPP, IMMPP, NIMMPP, and SMMPP are *on-off* processes which can have at most one burst in the system. In contrast, the PP, MMPP, and NMMPP are not *on-off* sources since they can have more than one burst in the system. Strictly speaking, they are inappropriate for the given burst-oriented framework which allows only one burst at a time. However, they are included for comparison.

The interarrival times of the PP and IPP are independent. On the contrary, the interarrival times of the Markov-modulated processes are autocorrelated. From Figure 5.8, it is apparent that in this system which does not queue bursts as such, the introduction of autocorrelation through Markov chain modulation does not adversely affect the BBP as long as one of two conditions holds: either the process is of the

on-off type, or the process has a relatively stable source load.

The first condition is demonstrated by the relative closeness of the observed BBPs of the different *on-off* processes, which include the SMMPP. Note that without serialization, the MMPP simulations resulted in BBPs that were higher than allowed. However, after serializing the bursts from each source, which can be viewed as a form of traffic policing, the SMMPP simulations resulted in BBPs that were below the target BBP and were in the neighborhood of the BBPs of the other simulations with *on-off* sources. That the BBPs were significantly below the target BBP can be attributed to the fact that the CAC scheme was derived from a model of the aggregate traffic of non-*on-off* PP sources, effectively making the BBP computations conservative for *on-off* sources (with equivalent effective source loads).

The second condition is demonstrated by the NMMPP simulations. Note that as is, the MMPP effectively resulted in an individual source load that fluctuated together with the arrival rate that varied according to the modulating Markov chain. In the NMMPP, the burst length distribution also varied according to the Markov chain state so that the source load was effectively the same for both Markov chain states. This resulted in BBPs that were in the neighborhood of the BBPs of the PP simulations. Naturally, since the CAC scheme used was based on the Erlang loss system which assumed PP sources,[9] the BBPs of the PP simulations were in the neighborhood of the target BBP of 10^{-4}.

5.6 Summary

A simple call admission control scheme to control the burst blocking probability has been developed for the framework presented in Chapter 4. It is based on *Erlang's loss*

[9]Technically, it assumes a single Poisson process source. However, as already pointed out, combining Poisson processes results in an aggregate Poisson process.

formula and is shown to be computationally efficient. The scheme allows connections to have arbitrary burst length distributions. It groups the burst-oriented connections into classes of the same peak rate and is shown to achieve high utilizations when used with eager transmission and when there are many connections in a class. The scheme has been evaluated for a wide range of parameters in three hypothetical network scenarios and compared with a more general approach using a non-Poisson source model. Furthermore, simulations have validated the effectivity of the scheme in controlling the BBP. For the most part, the CAC scheme appears to be a viable alternative to computationally more complex approaches even for non-Poisson traffic.

Chapter 6

A Burst-Level Priority CAC Scheme for Bursty Traffic

Statistical gain is achieved in ATM networks by making bursty connections share resources stochastically. A CAC scheme is used to limit the number of admitted connections to guarantee their QOS requirements. When connections with different QOS requirements share the same resources, the highest QOS requirements would typically be the limiting factor in determining the admissible load at a link. This may lead to connections with low QOS requirements getting better service than they require, leading to an underutilization of the resources. To alleviate this problem, a burst-level priority scheme is proposed. In the proposal, priorities are associated with connections and bursts rather than individual cells. This chapter presents the details and the evaluation of a two-level priority CAC scheme for controlling the burst-level blocking rates of independent heterogeneous on-off sources.

6.1 Motivation

One of the primary attractions of the ATM network is its statistical multiplexing capability. It allows VBR and bursty connections to share the same limited resources (e.g., bandwidth and buffers) stochastically. This, in turn, allows more connections to be supported simultaneously than in a circuit-switched or a *synchronous transfer mode* (STM) network.

If all traffic from different connections with different QOS requirements are treated identically in the network, then the connections with lower QOS requirements will tend to receive better service than they require. This is the case because the higher QOS requirements of the other connections would tend to limit the available competition (i.e., cause the CAC scheme to admit fewer competing connections). Therefore, without priority handling of traffic within the network, the resources may become underutilized.

Different connections may have different levels of tolerance for data loss. Furthermore, a connection may tolerate different levels of loss rates depending on the nature of the data it is currently transmitting. On the ATM cell level, the CLP bit is provided in the header to distinguish between the more loss-sensitive portion of the traffic and the less sensitive. In this chapter, an approach is proposed to provide priority handling of bursts of cells (rather than individual cells) inside the network.

The proposal consists of a burst admission priority policy and a CAC procedure that controls the BBPs of the two priority levels. The proposal provides a way for bursty connections with different loss requirements to more efficiently share resources, specifically bandwidth. The correctness of the CAC procedure (i.e., that it controls the BBPs to the proper levels) has been verified using simulation techniques.

Using analytical techniques, the performance of the two-priority scheme is compared with that of the alternative approach where connections with different loss

requirements do not share resources. The results demonstrate that a significant increase in utilization is achieved by using the priority scheme in a wide variety of configurations.

The rest of the chapter is organized as follows. Section 6.2 defines the type of traffic sources that the priority CAC scheme is intended to handle. Section 6.3 describes the burst-level admission procedure and the CAC scheme for connections with identical peak rates. The relevant portions of the CAC scheme are then generalized to handle connections with non-identical peak rates in Section 6.4. The performance benefit of the priority scheme is evaluated in Section 6.5. A summary of the chapter is given in Section 6.6.

6.2 Traffic Model

The traffic sources are assumed to satisfy the following conditions:

- Each source is of the *on-off* type that transmits at its peak rate between periods of silence. The distributions of the *on* and *off* periods are unspecified. Note that the peak cell transmission rate can be easily enforced with a *leaky bucket* mechanism (see Section 3.4.1). Furthermore, when a source transmits at a non-zero rate less than the peak rate, it would be treated as if it were transmitting at the peak rate.[1]

- The long-term bandwidth demand of a source is characterized by the probability that the source is active. This probability, which we will call the *source load*, may be different for each source. Note that for random *on-off* sources that always transmit at the peak rate during the *on* periods, this probability is

[1] Appropriate buffering at the source may be employed to reduce the underutilization resulting from this condition.

equivalent to the mean cell transmission rate of the source normalized to the peak rate, i.e., the mean rate divided by the peak rate.

Like the peak rate, the mean rate can also be enforced using a leaky bucket mechanism [33] (separate from the leaky bucket mechanism enforcing the peak rate). However, for maximum effectivity, this leaky bucket mechanism may have to be enhanced so as to take into account the burst groupings of cells.

- The sources are independent in the sense that at any instant, the state of a source (i.e., whether it is active or inactive) is independent of the states of the others. This is a reasonable assumption that is commonly made for unrelated connections whose transmissions are typically initiated by non-conspiring parties.[2] Note that the independence of the transmissions of admitted connections is the issue here, not the independence of the connection establishment request arrivals.

Observe that no assumption is made on the burst arrival process. Furthermore, note that if the sources are homogeneous (i.e., with the same peak rate and source load), then this model reduces to the general non-Poisson model discussed briefly in Section 5.5.3. Here, we are concerned with heterogeneous sources.

As required by the burst-oriented traffic control framework, the beginning and end of bursts should be marked as such to allow the "on-the-fly" allocation and deallocation of bandwidth to connections as bursts come and go. When a burst arrives at a switch, bandwidth equal to the peak rate of the source, if available, is allocated to the connection. This bandwidth is deallocated when the end of the burst is detected. If the required bandwidth is not available when a burst arrives, then the whole burst is dropped.

[2]The case of (known) cooperating sources is handled in Chapter 7.

6.3 CAC Scheme for Connections with Identical Peak Rates

In this section, a call admission control scheme for connections with identical peak rates is presented. The CAC scheme guarantees two levels of service: a low burst blocking probability (BBP) for high-priority bursts and a higher BBP for low-priority bursts. For now, we will assume that all bursts belonging to the same connection have the same priority (i.e., each connection has a fixed BBP requirement). However, allowing connections to carry bursts of different priorities is also possible and is discussed in Section 6.3.5.

6.3.1 Burst Admission Priority Policy

Connections using the same outgoing link at a switch are to partially share a set of *bandwidth units* each equal to the peak rate of the individual connections. This collection of bandwidth units is allocated to the whole set of connections—individual connections will "hold" the bandwidth units only during their admitted bursts. A high-priority burst is dropped only when there is no bandwidth unit available when it arrives. In contrast, a low-priority burst is also dropped whenever the number of low-priority bursts in the system has reached a (lower) threshold, even if there may be an unused bandwidth unit available.

More formally, we define the following variables:

- Let n = total number of bandwidth units allocated to the set of connections.
- Let m = low-priority threshold, $m \leq n$.
- Let h = number of bandwidth units currently used by high-priority bursts.
- Let ℓ = number of bandwidth units currently used by low-priority bursts.

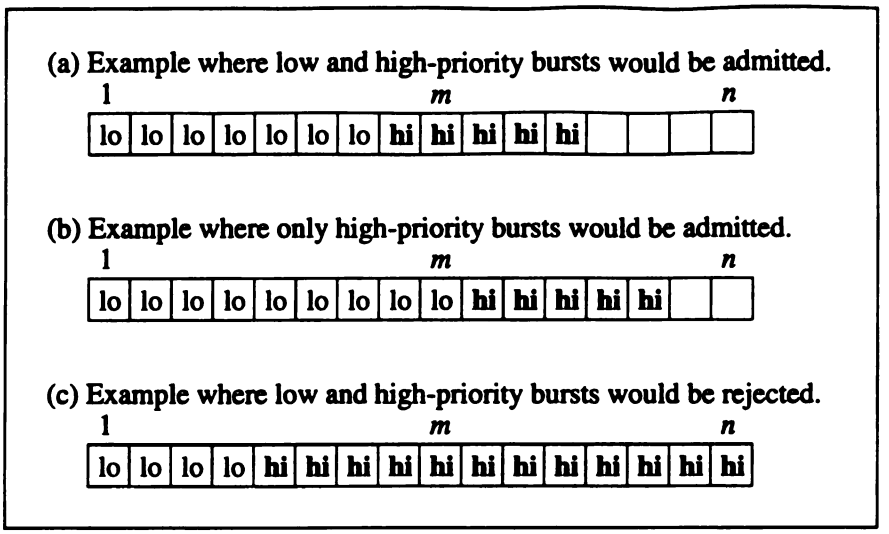

Figure 6.1: Burst admission priority policy. Each box represents a bandwidth unit. A box marked "lo" has been allocated to an admitted low-priority burst and a box marked "hi" has been allocated to an admitted high-priority burst. The policy allows for only up to m bandwidth units to be used by low-priority bursts but up to n bandwidth units to be used by high-priority bursts at the same time.

A high-priority burst is admitted if $h+\ell < n$ when it arrives; otherwise, it is dropped. In contrast, a low-priority burst is admitted if both $\ell < m$ and $h + \ell < n$ when it arrives; otherwise, it is dropped. The burst admission priority policy is illustrated in Figure 6.1.

It is the objective of the CAC scheme to find the minimum n (and the corresponding m) that would achieve the required BBPs for the low and high-priority connections. In practice, connections will be incrementally added and removed from the set. The CAC scheme then dynamically adjusts m and n as connections are established and torn-down.

6.3.2 Bandwidth Demand Probability Tables

To calculate the BBPs for the set of low-priority and the set of high-priority connections, two tables are maintained. These tables will store the discrete probability mass functions of the simultaneous bandwidth demands of each set of connections.

In theory, this will require a maximum table size equal to the maximum number of connections; but in practice, a table size equal to the maximum number of bandwidth units will suffice for the BBP computations.

First, we define the following parameters and notation:

- Let N = maximum number of bandwidth units that could be allocated to the set of bursty connections.
- Let the *source loads* of the s connections, $c_1, c_2, \ldots, c_s$, be $p_1, p_2, \ldots, p_s$, respectively. For convenience, let $q_i = 1 - p_i$, for all i. We will assume that $0 < p_i < 1$ for all i.
- Let $\mathcal{L}$ = set of low-priority connections.
- Let $\mathcal{H}$ = set of high-priority connections.
- Let $P_{\mathcal{C}}(k)$ = probability that exactly k connections in some set $\mathcal{C}$ (e.g., $\mathcal{L}$ or $\mathcal{H}$) are simultaneously active. This is given by

$$P_{\mathcal{C}}(k) = \sum_{\mathcal{S} \subseteq \mathcal{C}, |\mathcal{S}| = k} \left(\prod_{c_i \in \mathcal{S}} p_i \prod_{c_j \in \mathcal{C} - \mathcal{S}} q_j \right) \tag{6.1}$$

  where the product in the summation is simply the probability that all and only the connections in $\mathcal{S}$ are active. The summation, in turn, is over all the subsets of $\mathcal{C}$ that are of size k.

The values of $P_{\mathcal{C}}(k)$ for $k = 0, 1, \ldots, N-1$ may be stored in a table and computed incrementally as connections are added to or removed from an arbitrary set $\mathcal{C}$ by using the following recurrence relations (cf. [82]).

- For an empty set of connections, we simply have

$$P_{\emptyset}(k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}$$

- When adding a connection to a set $\mathcal{C}$, we have, for $c_i \notin \mathcal{C}$

$$P_{\mathcal{C}\cup\{c_i\}}(k) = \begin{cases} q_i P_{\mathcal{C}}(k) & \text{if } k = 0 \\ p_i P_{\mathcal{C}}(k-1) + q_i P_{\mathcal{C}}(k) & \text{if } k > 0 \end{cases} \tag{6.2}$$

Note that $P_{\mathcal{C}}(k)$ vanishes for $k > |\mathcal{C}|$. From the equations above, computing the new table of probabilities,[3] when adding a new connection to a set $\mathcal{C}$, will require $2|\mathcal{C}| + 1$ multiplications and $|\mathcal{C}| + 1$ additions. If we limit the table to N entries (i.e., for $k = 0, 1, ..., N-1$), then this will require at most $2N - 1$ multiplications and $N - 1$ additions.

- When removing a connection from a set $\mathcal{C}$, we have, for $c_j \in \mathcal{C}$

$$P_{\mathcal{C}-\{c_j\}}(k) = \begin{cases} \frac{P_{\mathcal{C}}(k)}{q_j} & \text{if } k = 0 \\ \frac{P_{\mathcal{C}}(k) - p_j P_{\mathcal{C}-\{c_j\}}(k-1)}{q_j} & \text{if } k > 0 \end{cases} \tag{6.3}$$

which can be obtained from Equation 6.2 using a "backward substitution" approach. From Equation 6.3 above, computing the new table of probabilities,[4] when removing a connection from a set $\mathcal{C}$, will require 1 division (to compute $1/q_j$), $2|\mathcal{C}| - 1$ multiplications, and $|\mathcal{C}| - 1$ subtractions. If we limit the table to N entries (i.e., for $k = 0, 1, ..., N-1$), then this will require at most $2N - 1$ multiplications and $N - 1$ subtractions.

[3] which can be done *in place*, if necessary or convenient, by computing the new entries from *right to left*

[4] which can be done *in place* by computing the new entries from *left to right*

Note that in general, $P_{\mathcal{C}}(k)$ is non-zero for values of k up to $|\mathcal{C}|$. However, the CAC scheme will not use values of $P_{\mathcal{L}}(k)$ or $P_{\mathcal{H}}(k)$ for $k \geq N$.

6.3.3 Computing the BBP Bounds

For convenience, the following notation is introduced: Let $P^{+}_{\mathcal{C}}(k)$ = probability that k *or more* connections in set $\mathcal{C}$ are simultaneously active. This is given by

$$P^{+}_{\mathcal{C}}(k) = 1 - \sum_{i=0}^{k-1} P_{\mathcal{C}}(i) \tag{6.4}$$

Using this notation, the BBP for a low-priority connection c_i, given the parameters m and n (as defined in Section 6.3.1 above), is bounded above by

$$\mathrm{BBP}_{c_i} = \left(\sum_{k=0}^{m-1} P_{\mathcal{L}-\{c_i\}}(k) P^{+}_{\mathcal{H}}(n-k) \right) + P^{+}_{\mathcal{L}-\{c_i\}}(m) \tag{6.5}$$

where the summation covers the cases where there are less than m active low-priority connections but there are sufficient active high-priority connections to make up a total of n or more active connections of both type. The remaining term covers the cases where there are m or more active low-priority connections.

Similarly, for a high-priority connection c_j, the BBP is bounded above by

$$\mathrm{BBP}_{c_j} = \left(\sum_{k=0}^{m-1} P_{\mathcal{L}}(k) P^{+}_{\mathcal{H}-\{c_j\}}(n-k) \right) + P^{+}_{\mathcal{L}}(m) P^{+}_{\mathcal{H}-\{c_j\}}(n-m) \tag{6.6}$$

where the summation covers the cases where there are less than m active low-priority connections but there are sufficient active high-priority connections to make up a total of n or more active connections of both type. The remaining term covers the cases where there are m or more active low-priority connections (of which at most m could be *activated* and holding bandwidth) and sufficient active high-priority connections to make up a total of n or more active connections (when only m *activated* connections

are counted among the active low-priority connections).

Normally, we are just concerned with the worst-case BBP experienced by each set of connections. For the low-priority connections, this will be experienced by the connection c_i with the minimum p_i in $\mathcal{L}$. For the high-priority connections, this will be experienced by the connection c_j with the minimum p_j in $\mathcal{H}$. Therefore, it would be sufficient to compute the BBP bounds for those connections rather than all connections.

For large $|\mathcal{L}|$ and $|\mathcal{H}|$, including or excluding c_i and c_j from the computations will make little difference, in which case it is preferable just to include them in the (conservative) approximations of the BBP bounds. Therefore, for large $|\mathcal{L}|$ and $|\mathcal{H}|$, when determining c_i and c_j can be costly, we can use the following upper bound for the BBP of the low-priority connections

$$B_1(\mathcal{L},\mathcal{H},m,n) = \left(\sum_{k=0}^{m-1} P_{\mathcal{L}}(k)P_{\mathcal{H}}^{+}(n-k)\right) + P_{\mathcal{L}}^{+}(m)$$

where the right hand side is identical to that in Equation 6.5 above except that c_i is left in $\mathcal{L}$ in the computation. Similarly, we can use the following upper bound for the BBP of the high-priority connections

$$B_2(\mathcal{L},\mathcal{H},m,n) = \left(\sum_{k=0}^{m-1} P_{\mathcal{L}}(k)P_{\mathcal{H}}^{+}(n-k)\right) + P_{\mathcal{L}}^{+}(m)P_{\mathcal{H}}^{+}(n-m)$$

where the right hand side is identical to that in Equation 6.6 above except that c_j is left in $\mathcal{H}$ in the computation.

Using Equation 6.4, we can compute $B_1(\mathcal{L},\mathcal{H},m,n)$ or $B_2(\mathcal{L},\mathcal{H},m,n)$ from $P_{\mathcal{L}}$ and $P_{\mathcal{H}}$ with m multiplications, m additions, and $m+n$ subtractions. Finally, note that we are only interested in computing $B_1(\mathcal{L},\mathcal{H},m,n)$ and $B_2(\mathcal{L},\mathcal{H},m,n)$ for values of m and n satisfying $m \leq n \leq N$.

6.3.4 Summary of CAC Scheme and Computational Requirements

Let $\hat{B}_1$ and $\hat{B}_2$ be the maximum tolerable BBPs for the low-priority and high-priority connections, respectively, where $\hat{B}_1 \geq \hat{B}_2$. Given $\mathcal{L}$, $\mathcal{H}$, m, and n, the procedure for accepting a new connection c_z is as follows.

1. If c_z is a low-priority connection, then let $\mathcal{L}_z = \mathcal{L} \cup \{c_z\}$ and let $\mathcal{H}_z = \mathcal{H}$. Otherwise, let $\mathcal{L}_z = \mathcal{L}$ and let $\mathcal{H}_z = \mathcal{H} \cup \{c_z\}$. In other words, add the connection to the appropriate set.

2. If $|\mathcal{L}_z|$ is *small* (e.g., if $|\mathcal{L}_z|$ is less than some small multiple of N), then find the $c_{\hat{x}}$ that has the smallest $p_{\hat{x}}$ in $\mathcal{L}_z$, and let $\mathcal{L}'_z = \mathcal{L}_z - \{c_{\hat{x}}\}$. Otherwise, let $\mathcal{L}'_z = \mathcal{L}_z$. That is, only if the set of low-priority connections is small will we bother to find the particular low-priority connection that is expected to experience the highest BBP.

3. Similarly, if $|\mathcal{H}_z|$ is small, then find the $c_{\hat{y}}$ that has the smallest $p_{\hat{y}}$ in $\mathcal{H}_z$, and let $\mathcal{H}'_z = \mathcal{H}_z - \{c_{\hat{y}}\}$. Otherwise, let $\mathcal{H}'_z = \mathcal{H}_z$.

4. Find the required m and n.

   (a) If $B_1(\mathcal{L}'_z, \mathcal{H}_z, m, n) \leq \hat{B}_1$ and $B_2(\mathcal{L}_z, \mathcal{H}'_z, m, n) \leq \hat{B}_2$, then let $m' = m$ and $n' = n$. In this case, adding the new connection does not cause any of the BBPs to exceed the targeted values with the current m and n.

   (b) Otherwise, if $B_1(\mathcal{L}'_z, \mathcal{H}_z, m+1, n) \leq \hat{B}_1$ and $B_2(\mathcal{L}_z, \mathcal{H}'_z, m+1, n) \leq \hat{B}_2$, then let $m' = m + 1$ and $n' = n$. In this case, increasing the low-priority threshold m (without increasing the size of the bandwidth pool n) is sufficient to keep the BBPs in check.

(c) Otherwise, if $B_1(\mathcal{L}'_z, \mathcal{H}_z, m, n+1) \leq \hat{B}_1$ and $B_2(\mathcal{L}_z, \mathcal{H}'_z, m, n+1) \leq \hat{B}_2$, then let $m' = m$ and $n' = n+1$. In this case, increasing the bandwidth pool (without increasing the low-priority threshold) is sufficient.

(d) Otherwise, let $m' = m+1$ and $n' = n+1$. Since the new connection can use up at most one bandwidth unit, then this should suffice.

5. If $n' - n$ bandwidth units can be added to the collection of bandwidth units for $\mathcal{L} \cup \mathcal{H}$, then accept the connection c_z and let $\mathcal{L} = \mathcal{L}_z$, $\mathcal{H} = \mathcal{H}_z$, $m = m'$, and $n = n'$. Otherwise, reject the connection (at the link in question) and leave $\mathcal{L}$, $\mathcal{H}$, m, and n as they were.

As long as we have $m \leq n \leq N$, then all the above steps can be done in $O(N)$ time. The corresponding procedure to release a connection c_w is given below. Note the similarities with the steps given above.

1. If c_w is a low-priority connection, then let $\mathcal{L}_{\bar{w}} = \mathcal{L} - \{c_w\}$ and let $\mathcal{H}_{\bar{w}} = \mathcal{H}$. Otherwise, let $\mathcal{L}_{\bar{w}} = \mathcal{L}$ and let $\mathcal{H}_{\bar{w}} = \mathcal{H} - \{c_w\}$.

2. If $|\mathcal{L}_{\bar{w}}|$ is small, then find the $c_{\hat{x}}$ that has the smallest $p_{\hat{x}}$ in $\mathcal{L}_{\bar{w}}$, and let $\mathcal{L}'_{\bar{w}} = \mathcal{L}_{\bar{w}} - \{c_{\hat{x}}\}$. Otherwise, let $\mathcal{L}'_{\bar{w}} = \mathcal{L}_{\bar{w}}$.

3. If $|\mathcal{H}_{\bar{w}}|$ is small, then find the $c_{\hat{y}}$ that has the smallest $p_{\hat{y}}$ in $\mathcal{H}_{\bar{w}}$, and let $\mathcal{H}'_{\bar{w}} = \mathcal{H}_{\bar{w}} - \{c_{\hat{y}}\}$. Otherwise, let $\mathcal{H}'_{\bar{w}} = \mathcal{H}_{\bar{w}}$.

4. Find the required m and n.

(a) If $B_1(\mathcal{L}'_z, \mathcal{H}_z, m-1, n-1) \leq \hat{B}_1$ and $B_2(\mathcal{L}_z, \mathcal{H}'_z, m-1, n-1) \leq \hat{B}_2$, then let $m' = m-1$ and $n' = n-1$.

(b) Otherwise, if $B_1(\mathcal{L}'_z, \mathcal{H}_z, m, n-1) \leq \hat{B}_1$ and $B_2(\mathcal{L}_z, \mathcal{H}'_z, m, n-1) \leq \hat{B}_2$, then let $m' = m$ and $n' = n-1$.

(c) Otherwise, if $B_1(\mathcal{L}'_z, \mathcal{H}_z, m-1, n) \leq \hat{B}_1$ and $B_2(\mathcal{L}_z, \mathcal{H}'_z, m-1, n) \leq \hat{B}_2$, then let $m' = m-1$ and $n' = n$.

(d) Otherwise, let $m' = m$ and $n' = n$.

5. Release $n - n'$ bandwidth units from the collection of bandwidth units for $\mathcal{L} \cup \mathcal{H}$, and let $\mathcal{L} = \mathcal{L}_{\bar{w}}$, $\mathcal{H} = \mathcal{H}_{\bar{w}}$, $m = m'$, and $n = n'$.

Again, all the above steps can be done in $O(N)$ time.

6.3.5 Allowing Connections to Use Both Priorities

This section briefly discusses a technique for admitting a connection that transmits bursts of different priorities. The idea is simply to consider such a connection as having two burst streams, one consisting of the low-priority bursts and the other consisting of the high-priority bursts. The low-priority component is incorporated into $\mathcal{L}$ (and $P_{\mathcal{L}}$) and the high-priority component is incorporated into $\mathcal{H}$ (and $P_{\mathcal{H}}$). Assuming that the two components may be considered as independent with each of them having its own *source load*, then the CAC scheme will apply. If it is the case that such connections transmit at most one burst at a time, then the CAC scheme would effectively be a little conservative.

6.4 CAC Scheme for Connections with Non-Identical Peak Rates

For simplicity, we will assume that the peak rates of the sources during their *on* state are multiples of a basic bandwidth unit (e.g., 64 Kbps). For connection c_i, we will use the symbol R_i to indicate its peak rate in number of bandwidth units. We now generalize the definition of $P_{\mathcal{C}}(k)$ to be based on total simultaneous bandwidth

demand. That is, we let $P_{\mathcal{C}}(k)$ = probability that the total bandwidth demand of set $\mathcal{C}$ (e.g., $\mathcal{L}$ or $\mathcal{H}$) is exactly k bandwidth units. This is given by

$$P_{\mathcal{C}}(k) = \sum_{\{\mathcal{S} \subseteq \mathcal{C} \mid \sum_{c_i \in \mathcal{S}} R_i = k\}} \left(\prod_{c_i \in \mathcal{S}} p_i \prod_{c_j \in \mathcal{C} - \mathcal{S}} q_j \right)$$

where the product in the summation is simply the probability that all and only the connections in $\mathcal{S}$ are active. The summation, in turn, is over all subsets $\mathcal{S}$ whose total bandwidth demand is k when all connections in $\mathcal{S}$ are simultaneously active (cf. Equation 6.1). The recurrence relation for adding a new connection c_i to set $\mathcal{C}$ is now

$$P_{\mathcal{C} \cup \{c_i\}}(k) = \begin{cases} q_i P_{\mathcal{C}}(k) & \text{if } 0 \leq k < R_i \\ p_i P_{\mathcal{C}}(k - R_i) + q_i P_{\mathcal{C}}(k) & \text{if } k \geq R_i \end{cases} \tag{6.7}$$

(cf. Equation 6.2). Finally, the corresponding recurrence relation for removing a connection c_j from the set $\mathcal{C}$ is

$$P_{\mathcal{C} - \{c_j\}}(k) = \begin{cases} \frac{P_{\mathcal{C}}(k)}{q_j} & \text{if } 0 \leq k < R_j \\ \frac{P_{\mathcal{C}}(k) - p_j P_{\mathcal{C} - \{c_j\}}(k - R_j)}{q_j} & \text{if } k \geq R_j \end{cases} \tag{6.8}$$

(cf. Equation 6.3). The above generalization is not new as it is equivalent to the technique proposed by Turner in [82] for calculating the total instantaneous buffer demand of a set of bursty connections with varying buffer requirements (but with the same priority).

To handle connections with non-identical peak rates, the multi-priority CAC scheme described in the previous section has to be modified in the following ways:

- The more general method of computing $P_{\mathcal{C}}$ (for $\mathcal{L}$ and $\mathcal{H}$) as given in Equations 6.7 and 6.8 is to be used.

- The CAC steps testing the values of $B_1(\mathcal{L}, \mathcal{H}, m, n)$ and $B_2(\mathcal{L}, \mathcal{H}, m, n)$ will have

to be modified to test the corresponding values of $B_1(\mathcal{L}, \mathcal{H}, m - R_i + 1, n)$ and $B_2(\mathcal{L}, \mathcal{H}, m, n - R_j + 1)$ for values of R_i (for low-priority connections) and R_j (for high-priority connections) of interest.

- Adjusting m and n may require several steps (i.e., adding one to either or both of them might not be sufficient if the new connection has a peak rate greater than one bandwidth unit).

Note that an even more general approach for computing P_C is possible, one in which a connection is allowed to have multiple levels of bandwidth demand. This has two applications. First, it could be used to handle connections that transmit bursts at different burst rates. Second, it could be used to handle sets of traffic sources that transmit bursts in a correlated manner. The latter is the topic of Chapter 7.

6.5 Performance of the Priority Scheme

First, simulations have been performed to validate the CAC procedure. Various configurations have been tested with different capacities, source loads, low and high-priority target BBP values, and source traffic generators. The source traffic generators included renewal on-off processes and autocorrelated processes such as Markov-modulated on-off processes. The simulations consistently resulted in measured BBPs that were around or below the target BBP. In general, various simulation studies have shown that *non-queued* burst-oriented CAC schemes based on the same traffic model, appear to be immune to (intra-source) autocorrelation, as long as the sources are independent.

The results of a set of simulations are shown in Figure 6.2.[5] In these simulations,

[5] In this set of simulations, the parameters were selected so as to facilitate the collection of many data points in a reasonable amount of computing time. Other configurations (not included here) were also simulated, with a single or a few data points, giving results that were also as expected.

the target low and high-priority BBPs were set to 10^{-2} and 10^{-4}, respectively. Each source had an individual offered load of 0.2 and the number of high-priority sources was set equal to the number of low-priority sources (± 1). The CAC scheme presented in this chapter was used to determine the number of connections that can be accommodated for each capacity value of interest. Each simulation run consisted of a total of 2×10^6 bursts from the admitted connections, resulting in approximately 10^6 low-priority bursts and 10^6 high-priority bursts. Five runs were executed for each source type at each capacity value of interest.

Four different types of *on-off* sources were used. The interrupted Poisson process (IPP) was essentially a Poisson process whose arrival rate temporarily dropped to zero (i.e., was interrupted) while the process had a burst in the system. The burst lengths were exponentially distributed. The interrupted Markov-modulated Poisson process (IMMPP) was a Markov-modulated Poisson process (MMPP) that was interrupted (like the IPP) while it had a burst in the system. Recall that a non-interrupted MMPP was simply a non-homogeneous Poisson process whose arrival rate was determined by a two-state continuous-time Markov chain. The normalized IMMPP (NIMMPP) was a variation of the IMMPP in which the burst length distribution was adjusted together with the interarrival time distribution so that the effective source load at each state was the same. The Markov-modulated on-off process (MMOOP) was a Markov-modulated version of the IPP in which the burst length distribution (rather than the arrival rate) was modulated by a two-state continuous-time Markov chain. This has the effect of making the source load of fluctuate according to the Markov chain, without the arrival rate itself fluctuating.

From Figure 6.2, it is evident that the CAC scheme did control the low and high-priority BBPs as intended. Moreover, it appears that it did not matter if the interarrival times and/or the burst lengths were autocorrelated.

Note that the measured BBPs were lower than the target BBPs. Furthermore,

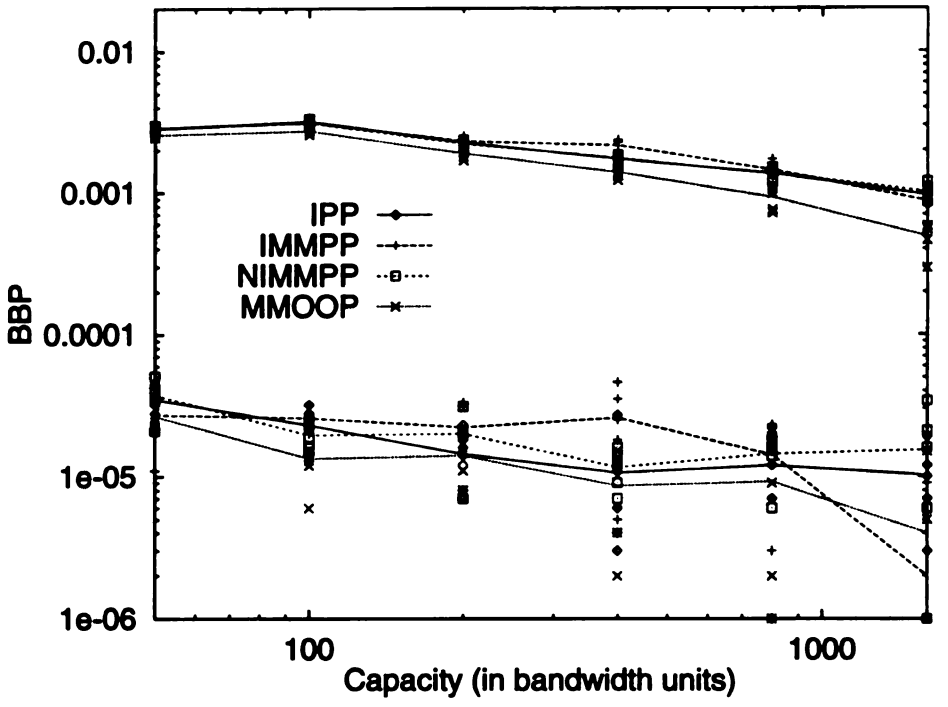

Figure 6.2: Simulation results using different *on-off* traffic sources. The target low and high-priority BBPs were set to 10^{-2} and 10^{-4}, respectively. The upper band consists of the observed low-priority BBPs and the lower band consists of the observed high-priority BBPs. Individual points plot the results of individual runs and the lines connect the corresponding averages. Interrupted Poisson process (IPP), interrupted Markov-modulated Poisson process (IMMPP), normalized IMMPP (NIMMPP), and Markov-modulated *on-off* process (MMOOP), described in the text, were the types of sources used.

the BBPs tended to decrease as the capacity increased. These phenomena can be attributed to the conservative nature of the CAC scheme which does not take burst blocking into account in its bandwidth demand calculations.[6] In reality (and in the simulations), burst blocking tends to come in bunches and dropping blocked bursts helps to reduce the BBP. This BBP reduction effect becomes more pronounced in a higher capacity system which effectively has a larger number of states (i.e., number of bandwidth units in use by bursts). The large number of states results in long periods without threat of burst blocking between short periods of high likelihood of burst blocking. Dropping blocked bursts effectively decreases the load during those periods where it would help the most.

In order to provide evidence for the explanations put forth in the previous paragraph, additional simulations were performed. The same simulations (i.e., the same sets of source processes) which produced the results in Figure 6.2 were repeated with a single modification. In the new simulations, the blocked bursts were made to be "persistent," meaning that instead of being dropped completely, they were allowed to stay in the system and later use bandwidth released by departing bursts. This does not mean that the blocked bursts were queued; rather, only the front part of a blocked burst was effectively dropped until a bandwidth unit was released by a departing burst. The new results are shown in Figure 6.3. Note that the measured BBPs were closer to the target BBPs and that there was no noticeable decreasing trend of the BBPs as the capacity increased.

In the rest of this section, we will concentrate on evaluating the performance benefit (in terms of increased utilization) offered by the two-level priority scheme using analytical methods. The priority scheme is compared with two alternative approaches. In one approach, the two classes of traffic are treated identically, i.e.,

[6]This is not to say that the scheme has to be modified to take burst blocking into account as it is not clear if it can be done at all. The scheme, with its conservative nature, is useful as is.

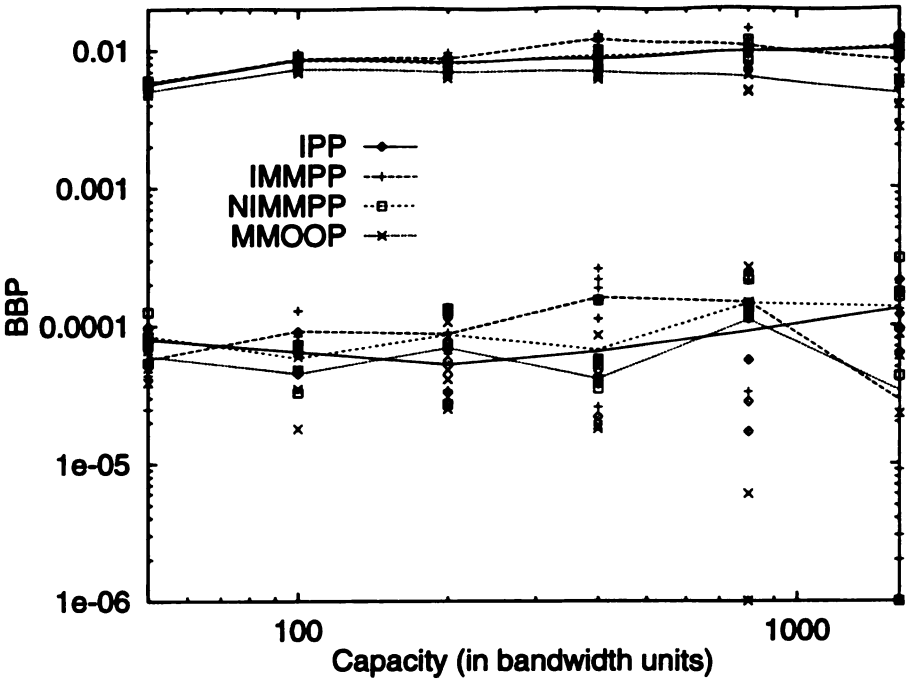

Figure 6.3: Results of the same simulations in the previous figure, but with bursts that were not dropped when blocked. In these simulations, blocked bursts were allowed to persist in the system and later use bandwidth released by departing bursts.

with the same (high) priority (low BBP). In the other approach, the two classes are managed separately and do not share a common bandwidth pool. The results are obtained analytically (i.e., using the CAC scheme formulas) assuming homogeneous sources for simplicity. Various parameters are varied in order to test the different factors that affect the utilization increase.

Figure 6.4 compares the utilizations obtained by using the two-priority scheme with those obtained by using the two alternative approaches mentioned above. The sources are homogeneous, each with a source load of 0.1 and a peak rate of 1 bandwidth unit. The target low-priority and high-priority BBPs are 10^{-4} and 10^{-8}, respectively. In this comparison, the number of high-priority sources is kept at approximately 10% of the total number of sources. The figure shows that for a wide range of capacities, the two-priority scheme offers a non-trivial improvement in the bandwidth utilization. However, the *relative* improvement decreases as the capacity

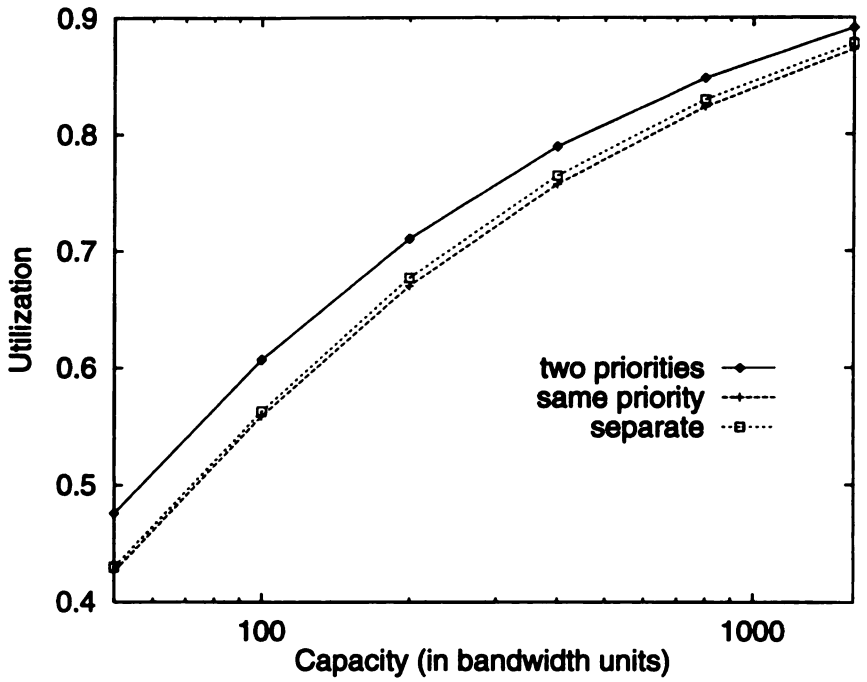

Figure 6.4: Utilizations for the three approaches for different total capacities. The fixed parameters are: low-priority BBP = 10^{-4}, high-priority BBP = 10^{-8}, individual source load $p = 0.1$, and individual peak rate = 1 bandwidth unit. The high-priority sources constitute approximately 10% of the total load.

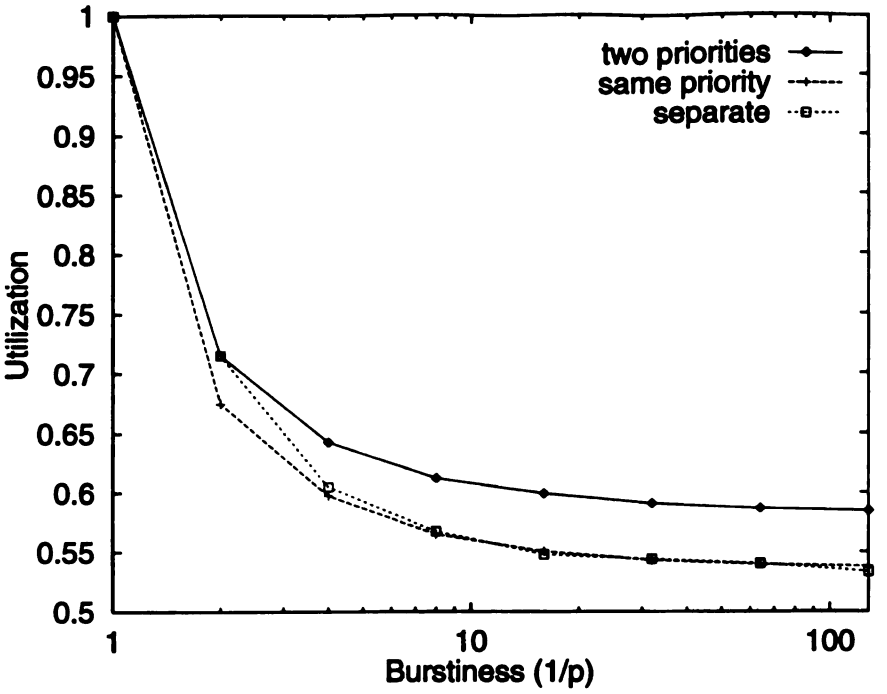

Figure 6.5: Utilizations for the three approaches for different burstiness values. The fixed parameters are: low-priority BBP = 10^{-4}, high-priority BBP = 10^{-8}, and capacity = 100 bandwidth units. The high-priority sources constitute approximately 10% of the total load.

increases. This is to be expected, as it is the case that for very high capacities, the attainable utilizations are high for any (reasonable) target BBPs. Alternatively, one can measure the benefit of the priority scheme in terms of the increase in the number of admitted connections. For example, in the case of the 50 bandwidth unit capacity, the 0.046 increase in the utilization over the separate approach represents 23 additional connections. In the case of the 1600 bandwidth unit capacity, the 0.013 increase represents 203 additional connections.

Figure 6.5 shows the effect of *burstiness* (which we define simply as $\frac{1}{p}$) on the utilizations of the three approaches. Note that for values of burstiness greater than or equal to 4 ($p \leq 0.25$), the curves are relatively flat and the two-priority scheme consistently offers a significant utilization increase in the neighborhood of 0.05.

In the previous figures, the number of high-priority sources was kept at

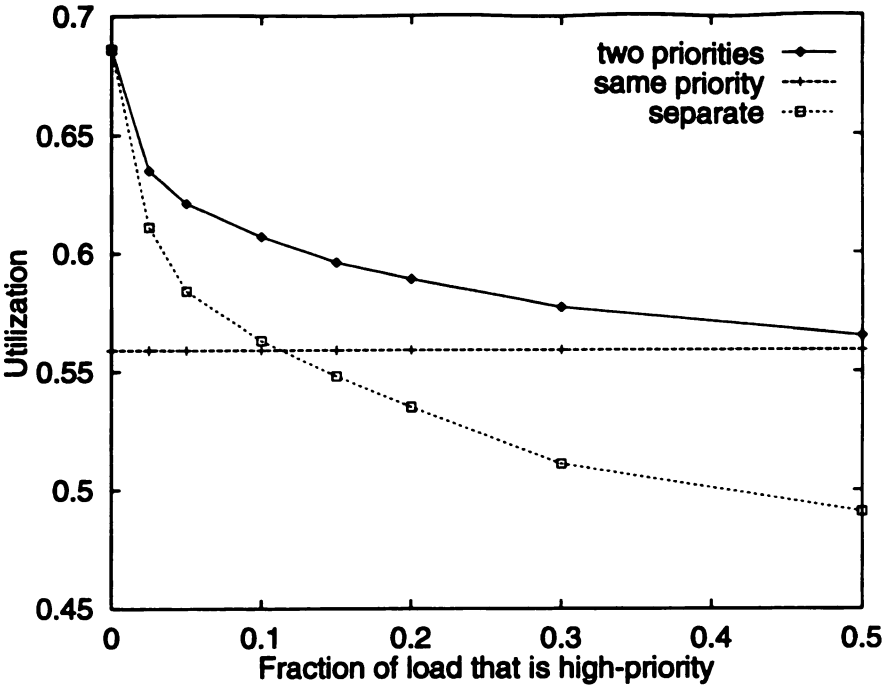

Figure 6.6: Utilizations for the three approaches for different relative amounts of high-priority traffic. The fixed parameters are: low-priority BBP = 10^{-4}, high-priority BBP = 10^{-8}, individual source load $p = 0.1$, and capacity = 100 bandwidth units.

approximately 10% of the total number of sources. In Figure 6.6, the fraction of the total load that is high-priority is varied. Evidently, the utilization gain offered by the two-priority scheme over *both* alternative approaches is most significant when the high-priority traffic constitutes about 5% to 20% of the total load. Furthermore, when compared with each of the two other approaches *one at a time* (after all, they cannot be used at the same time), the two-priority scheme offers a significant utilization gain for a wider range of relative amounts of high-priority traffic.

So far, we have seen results only for a selected pair of low-priority and high-priority BBP values, 10^{-4} BBP for low-priority bursts and 10^{-8} BBP high-priority bursts. Figure 6.7 compares the utilizations of the three approaches for different values of the low-priority BBP. In the figure, the low-priority BBP is varied across the x-axis while the high-priority BBP is fixed at 10^{-8}. The utilization gain offered by the two-priority scheme over *both* alternative approaches is most significant when the

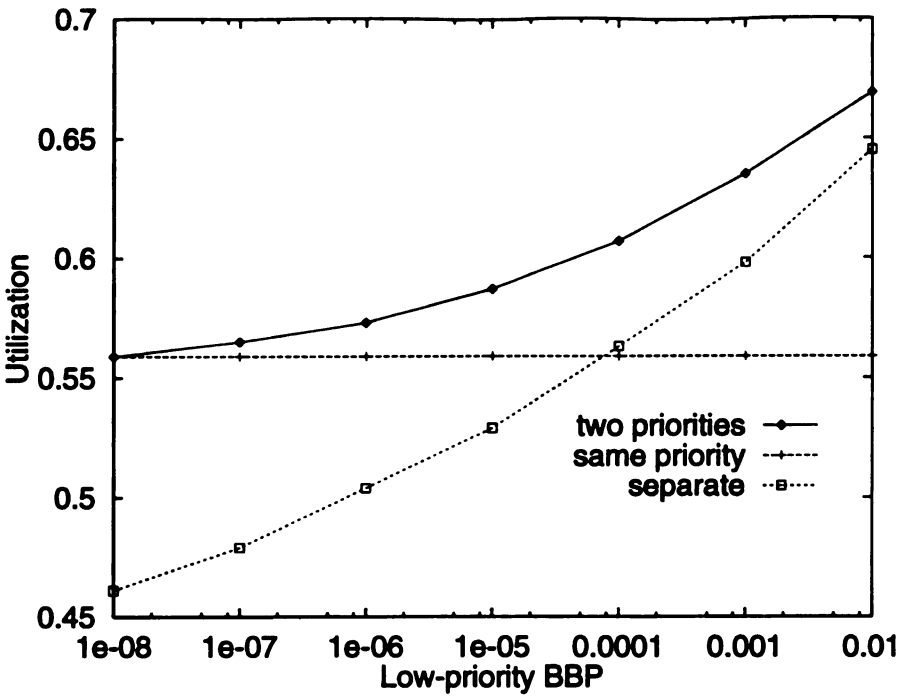

Figure 6.7: Utilizations for the three approaches for different low-priority BBP values. The fixed parameters are: individual source load $p = 0.1$ and capacity = 100 bandwidth units. The high-priority sources constitute approximately 10% of the total load.

low-priority BBP is between 10^{-5} and 10^{-3}. However, just like in the previous figure, when compared with each of the two other approaches *one at a time*, the two-priority scheme offers a significant increase in utilization over a wider range of values of the low-priority BBP.

6.6 Summary

In this chapter, a burst-level priority scheme for ATM networks has been presented. In the proposal, bursty traffic connections are to be allocated bandwidth on the fly on a burst-by-burst basis. Connections (or bursts within connections) can be assigned one of two priority levels with high-priority connections experiencing a lower burst blocking probability than the low-priority connections. The low and high-priority connections share the same bandwidth pool. However, the low priority connections

are constrained to use at most only a subset of the bandwidth pool. The cell buffering required by the approach is equivalent to the cell buffering required to support similarly jitter-impaired continuous bit rate (CBR) connections with the same bandwidth.

Burst-level (as opposed to strictly cell-level) traffic control has its merits. These include its capability of intentionally dropping practically useless cells from bursts that can not be transmitted integrally, its property of protecting admitted and ongoing bursts from subsequent congestion, and with this proposal, its capability for loss-prioritizing traffic on the burst (rather than the cell) level, which may be more appropriate for most data traffic.

The algorithm for the burst-level priority CAC scheme for general bursty sources has been given. The CAC scheme handles independent heterogeneous on-off sources without any specification of the burst length or interarrival time distributions. Using simulation techniques, the CAC scheme has proved to properly control the two levels of BBPs of the independent on-off sources, including renewal and autocorrelated source models (e.g., Markov-modulated processes).

Analytical performance studies show that the two-priority scheme offers significant utilization gains over two alternative approaches, one in which the two classes are treated identically (i.e., effectively with the same priority) and another in which the two classes have separate bandwidth pools. The utilization gain of the two-priority approach over the two alternative approaches is most significant when the total available capacity is moderate, the burstiness is moderate to high, the load is about 10% high-priority, and the low-priority BBP is several orders of magnitude higher than the high-priority BBP.

Finally, the CAC procedure may be extended to handle more than two priority levels but only at a considerable increase in computational complexity. Furthermore, having more than two priority levels is not expected to bring much additional performance benefit. This the case because the BBPs of *adjacent* priority levels will tend

to be close together. This happens to be a situation where the two-priority scheme does not offer much benefit over the "same priority" approach as evidenced by the results shown in Figure 6.7.

Chapter 7

A Correlation-Aware CAC Scheme for Bursty Traffic

As network capacities increase, wide-area distributed parallel computing may become feasible. This chapter addresses one of the issues involved in using an ATM network for such a purpose—that of developing an appropriate admission control procedure for such applications given the special nature of their traffic. In this proposal, connections belonging to the same application and sharing the same link are allowed to utilize the link bandwidth in a strongly correlated manner. However, connections belonging to different applications are still assumed to be independent. This allows the development of a tabular approach for keeping track of the aggregate bandwidth demand of the applications sharing the same link. The proposed approach is compared with two related approaches (one more conservative and another overly optimistic) and is shown to strike a balance between utilization and loss rate.

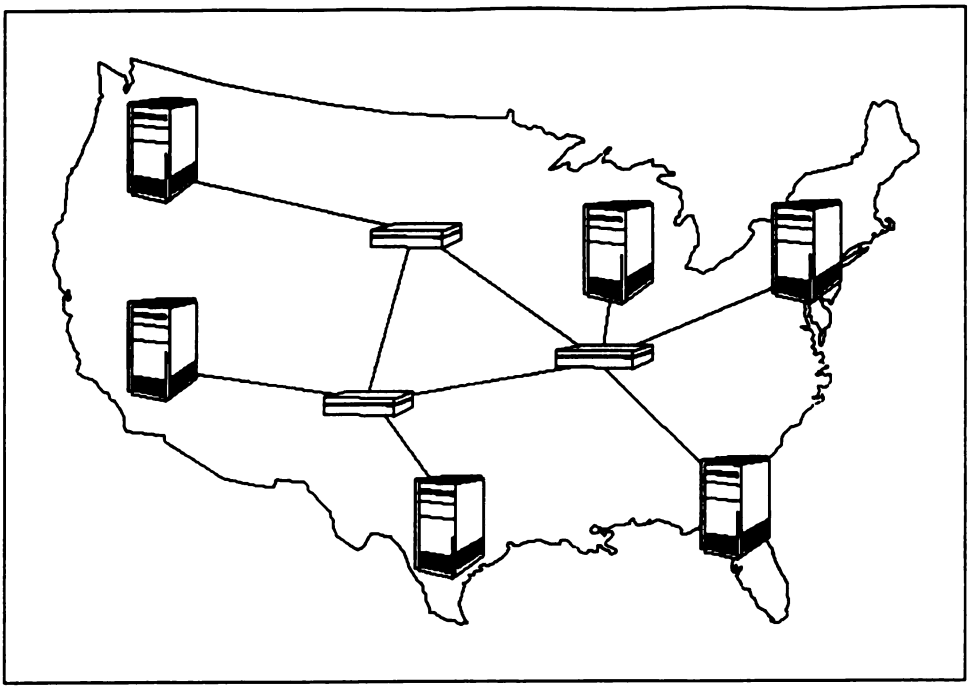

Figure 7.1: As network capacities increase, distributed or parallel computing over a wide-area network may become desirable and feasible.

7.1 Scenario

In the future, high-speed networks may make it feasible to solve "grand challenge" or other problems on a network of computers distributed over a large area (see Figure 7.1). The ATM network would be an appropriate network for such applications because of its high capacity, scalability, guaranteed quality of service (QOS), and ability to handle bursty traffic.

Recall that in ATM networks, all communication is connection-oriented. A call admission control (CAC) procedure is applied to connection establishment requests in order to determine whether the new call can be accommodated without violating the guaranteed QOS of the preexisting connections using the same links and buffers. Many different models of an individual bursty source's traffic have been proposed. However, it is typically assumed (not necessarily explicitly) that the individual sources are independent. This assumption is not likely to hold for distributed (or parallel)

computing applications.

Parallel or distributed computing typically requires some form of *collective communication*. This includes many-to-many, one-to-many, and many-to-one communication patterns. Various issues involved in supporting collective communication operations in ATM networks are discussed in [34].

Whether or not the collective communication primitives are implemented through hardware (or some lower protocol layer) operations or collections of unicast operations, does not affect the fact that the timing and volume of the traffic from the cooperating sources will likely be correlated. For example, in the *scatter-gather* operation, where a node collects results from a set of nodes to which it has sent requests simultaneously, it is expected that the gathered messages would utilize the network resources in a correlated manner. The same is true in the *reduction* operation, where a set of nodes performs a global operation on data that is distributed among the nodes in the network.

In this chapter, a simple approach for call admission control is proposed. The CAC scheme is designed to take into account the fact that some connections sharing the same link will have correlated bursty traffic. Typically, these correlated connections belong to the same distributed application. Therefore, in this proposal, a set of correlated connections sharing a link is called an *application*. Separate applications are assumed to be independent, however. This allows the development of a tabular method of keeping track of the overall bandwidth demands of all applications sharing the same link.

There are other traffic models that have been proposed that are able to capture some of the correlation in network traffic (see Section 3.1). The most notable example is the *Markov-modulated Poisson process* (MMPP) model. The MMPP has been used to model the aggregate arrival process of a theoretical integrated network carrying voice and data traffic [32]. It has also been used to model the traffic characteristics of

an actual wide-area network [10]. The main difference between MMPP modeling and the approach proposed in this chapter is that while the former is based on modeling an arrival process and a queueing system, the latter is not concerned about these things. Another difference is that the proposed approach attempts to capture the correlation among the known correlated traffic sources while still taking advantage of the likely non-correlation (independence) of other traffic sources. Another example of a correlated traffic model is the *packet train* [38] model. However, it is not clear how this model can be used in the context of call admission control.

An alternative approach that can be used to handle highly correlated traffic is to transmit such traffic through *available bit rate* (ABR) or *best-effort* service (see Section 3.4.4). The idea behind ABR service is to transmit the data, which in this case is assumed not to be delay-sensitive, across communication links only when the bandwidth is not otherwise used by bandwidth-reserved traffic. There are two general approaches that have been proposed for supporting ABR traffic: rate-based flow control [9] and credit-based flow control [45]. In addition, some have proposed hybrid approaches [36, 61]. The main disadvantage of using ABR services is their (inherent) lack of QOS guarantees in terms of delay and throughput. This is undesirable when performing parallel distributed computing over a large (public) network, because the unlimited network competition may introduce long delays in the system and defeat the purpose of distributing the work around the network.

The rest of this chapter is organized as follows. Section 7.2 presents the proposed *application-level* traffic model that will be used in the CAC scheme presented in Section 7.3. Section 7.4 applies the CAC scheme to an illustrative example and evaluates the performance of the proposed scheme vis-a-vis two related approaches. Section 7.5 gives a summary of the chapter.

7.2 Traffic Model

In this model, the traffic sources are organized into *applications* within which the burst generation may be correlated. The applications and traffic sources are assumed to satisfy the following conditions:

- The sources are of the *on-off* type with burst rates that are multiples of a fundamental bandwidth unit defined by the network (e.g., 64 Kbps). The peak rate of each source is to be policed at the entrance to the network.

- The long-term bandwidth demand of a set of correlated sources belonging to an application and sharing a link can be characterized by an empirical discrete probability distribution of the number of bandwidth units simultaneously required by the application. This can be represented in a table like the following:

r:	0	1	2	$\cdots$	R_i
$Q_i(r)$:	·	·	·	$\cdots$	·

  where $Q_i(r)$ is the proportion of time the application a_i requires r bandwidth units from the link, and R_i is the maximum number of bandwidth units that application a_i will require simultaneously from the link. Policing the aggregate bandwidth demand of an application inside the network is not practical. However, the individual contributions of the sources (i.e., their mean rates) may be policed at the boundaries of the network.

- The applications are independent in the sense that at any instant, the bandwidth demand of an application is independent of the bandwidth demands of the other applications. This is a reasonable assumption for non-conspiring applications.

Note that there are no assumptions on the burst arrival process (i.e., on the interarrival time distribution).

Not absolutely necessary but highly convenient in the development is the assumption of the existence of an underlying *burst-oriented* traffic control mechanism like that described in detail in Chapter 4.

7.3 CAC Scheme for Applications with Intra-Application Correlation

In this section, a call admission control scheme for independent applications is presented. The CAC scheme guarantees a level of service (i.e., burst blocking probability) for each burst class (i.e., burst peak rate in number of bandwidth units).

7.3.1 Burst Admission Policy

Applications using the same outgoing link at a switch are to partially share a set of *bandwidth units*. This collection of bandwidth units is allocated to the whole set of connections belonging to all applications—individual connections will "hold" the bandwidth units only during their admitted bursts. A burst is dropped when there is an insufficient number of available bandwidth units when it arrives.

It is the objective of the CAC scheme to find the minimum number of bandwidth units for the set that would achieve the required burst blocking probabilities (BBPs) for the different burst classes. In practice, connections will be incrementally added and removed from the set. The CAC scheme then dynamically adjusts the bandwidth allocation to the set as connections are established and torn-down.

7.3.2 Probability Tables

To calculate the BBPs for the different burst classes for the set of applications, a table is maintained. The table will store the discrete probability mass function of the aggregate bandwidth demands of the whole set of applications. In theory, this will require a maximum table size equal to the maximum possible number of bandwidth units that can be used by all connections simultaneously; but in practice, a table size equal to the maximum number of bandwidth units available in the link (i.e., the link capacity) will suffice for the BBP computations.

First, we define the following parameters and notation:

- Let M = maximum number of bandwidth units that could be allocated to the set of bursty applications.
- Let N = number of bandwidth units allocated to the set of bursty applications.
- Let $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ be the set of admitted applications.
- Let the maximum demands of the n applications, $a_1, a_2, \ldots, a_n$, be given by $R_1, R_2, \ldots, R_n$, respectively, and let their bandwidth demand probability mass functions be given by

$$Q_i(j) \text{ for } j = 0, 1, 2, \ldots, R_i$$

for $i = 1, 2, \ldots, n$, respectively. Obviously, for any i in $1, 2, \ldots, n$, it must be the case that

$$\sum_{j=0}^{R_i} Q_i(j) = 1$$

- Let $P_{\mathcal{A}}(k)$ = probability that the aggregate simultaneous bandwidth demand of the set $\mathcal{A}$ is k. This is given by

$$P_{\mathcal{A}}(k) = \sum_{\{r_1,r_2,\ldots,r_n \mid \sum r_i = k\}} \prod_{i=1}^{n} Q_i(r_i)$$

  which is the sum of the probabilities of all possible disjoint ways that exactly k total bandwidth units are required by the n applications.

The values of $P_{\mathcal{A}}(k)$ for $k = 0, 1, \ldots, M-1$ may be stored in a table and computed incrementally as connections are added to or removed from an arbitrary set $\mathcal{A}$ by using the following recurrence relations.

- For an empty set of connections, we simply have

$$P_{\emptyset}(k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}$$

- When adding a connection to a set $\mathcal{A}$, we have, for $a_i \notin \mathcal{A}$

$$P_{\mathcal{A}\cup\{a_i\}}(k) = \sum_{j=0}^{\min(k,R_i)} Q_i(j) P_{\mathcal{A}}(k-j) \tag{7.1}$$

  where $Q_i(j)P_{\mathcal{A}}(k-j)$ is simply the probability that the total bandwidth demand of $\mathcal{A} \cup \{a_i\}$ is exactly k.

  From equation (7.1) above, computing the new table of probabilities,[1] when adding a new connection to a set $\mathcal{A}$, will require less than $\hat{M} \times (R_i + 1)$ multiplications and fewer additions, where $\hat{M}$ is equal to the maximum aggregate bandwidth demand of all the applications in $\mathcal{A} \cup \{a_i\}$. If we limit the table

[1] which can be done *in place*, if necessary or convenient, by computing the new entries from *right to left*

to M entries (i.e., compute $P_{\mathcal{A}}(k)$ for $k = 0, 1, ..., M-1$ only), then this will require less than $M \times (R_i + 1)$ multiplications and fewer additions.

- When removing a connection from a set $\mathcal{A}$, we have, for $a_i \in \mathcal{A}$

$$P_{\mathcal{A}-\{a_i\}}(k) = \begin{cases} \frac{P_{\mathcal{A}}(0)}{Q_i(0)} & \text{if } k = 0 \\ \frac{P_{\mathcal{A}}(k) - \sum_{j=1}^{\min(k,R_i)} Q_i(j) P_{\mathcal{A}}(k-j)}{Q_i(0)} & \text{if } k > 0 \end{cases} \tag{7.2}$$

which can be solved from equation (7.1).

From equation (7.2) above, computing the new table of probabilities,[2] when removing a connection from a set $\mathcal{A}$, will require 1 division (to compute $1/Q_i(0)$), less than $\hat{M} \times (R_i + 1)$ multiplications, and fewer subtractions. If we limit the table to M entries, then this will require less than $M \times (R_i + 1)$ multiplications and fewer subtractions.

Note that in general, $P_{\mathcal{A}}(k)$ is non-zero for values of k up to $\hat{M}$. However, the CAC scheme will only use values of $P_{\mathcal{A}}(k)$ for $k < N \leq M$.

7.3.3 Computing the BBP Bounds

For convenience, we introduce the following notation: Let $P^{\dagger}_{\mathcal{A}}(k)$ = probability that the connections in set $\mathcal{A}$ simultaneously require (demand) k *or more* bandwidth units. This is given by

$$P^{\dagger}_{\mathcal{A}}(k) = 1 - \sum_{j=0}^{k-1} P_{\mathcal{A}}(j)$$

Using this notation, an upper bound for the BBP of a source belonging to an application a_i requiring at most R_i bandwidth units is given by

[2] which can be done *in place* by computing the new entries from *left to right*

$$P^{+}_{\mathcal{A}-\{a_i\}}(N - R_i + 1) = 1 - \sum_{j=0}^{N-R_i} P_{\mathcal{A}-\{a_i\}}(j)$$

The above formula is derived as follows. Given that we are given only the aggregate bandwidth probability table of application a_i, and given that we are interested in maintaining a *source-wise* BBP (for fairness), we have to consider the worst-case situation for a source within the application. Without loss of generality, select a source s in application a_i. Assume that it produces bursts requiring r bandwidth units. Since application a_i as a whole uses up at most R_i bandwidth units simultaneously, then the other sources in a_i (besides s) can use up at most $R_i - r$ bandwidth units at the same time, leaving $N - (R_i - r)$ free for source s and the other applications. In the worst-case but perfectly conceivable scenario, it is possible that whenever a burst from source s arrives, the other sources in a_i are already using their $R_i - r$ bandwidth units. This means that the burst from source s will be blocked unless the other applications in $\mathcal{A} - \{a_i\}$ are using less than $N - (R_i - r) - r + 1 = N - R_i + 1$ bandwidth units. The probability of this situation is given by $P^{+}_{\mathcal{A}-\{a_i\}}(N - R_i + 1)$.

Computing this value for each a_i in $\mathcal{A}$ can be computationally expensive when $|\mathcal{A}|$ is large. Therefore, when $|\mathcal{A}|$ is large, it is recommended that the upper bound $P^{+}_{\mathcal{A}}(N - \hat{R} + 1)$ be used for all applications instead. Here, $\hat{R}$ is the maximum number of bandwidth units required by any single application in $\mathcal{A}$.

7.3.4 Summary of CAC Scheme and Computational Requirements

Let $\hat{B}$ be the maximum tolerable BBP. Given $\mathcal{A}$, M, and N, the procedure to accept a new application $a_i \notin \mathcal{A}$ with the bandwidth demand table, $Q_i(r)$ for $r = 0, 1, 2, \ldots, R_i$,

is as follows.

1. Let $\mathcal{A}' = \mathcal{A} \cup \{a_i\}$ and let $\hat{R}$ be the maximum number of bandwidth units required by any single application in $\mathcal{A}'$.

2. Compute the new aggregate probability table, $P_{\mathcal{A}'}(k)$ for $k = 0, 1, \ldots, M-1$.

3. Find the minimum N' such that $P^{\dagger}_{\mathcal{A}'}(N' - \hat{R} + 1) \leq \hat{B}$.

4. If $N' - N$ additional bandwidth units are available for allocation to the set $\mathcal{A}$, then increase N to N' and accept the application. Otherwise, reject it.

Step 2 above requires less than $M \times (R_i + 1)$ multiplications and fewer subtractions[3] as long as we limit the table P to M entries, which is sufficient. Step 3 requires at most M subtractions and R_i comparisons, since we are only interested in values of N' not exceeding M, and since N' can be at most R_i more than N. As long as the value of R_i tends to be much less than M, then the algorithmic complexity is approximately linear with respect to M. However, if a value of R_i approaching M is quite common, then we suggest that the size (number of entries) of the Q_i table be scaled down appropriately as follows.

1. Determine an appropriate scale factor s_i that would make the size of the new table, $\lceil R_i/s_i \rceil + 1$, acceptable for computational purposes.

2. Construct a new table Q'_i consisting of the entries

$$Q'_i(0) = Q_i(0)$$

$$Q'_i(j) = \sum_{k=s_i \times (j-1)+1}^{s_i \times j} Q_i(k) \text{ for } j = 1, 2, \ldots, \lceil R_i/s_i \rceil$$

[3]Note that we are only counting the required floating point operations. The additions and comparisons associated with loop control are ignored.

3. Equation (7.1) now becomes

$$P_{\mathcal{A}\cup\{a_i\}}(k) = \sum_{j=0}^{\min(\lfloor k/s_i \rfloor, \lceil R_i/s_i \rceil)} Q_i'(j) P_{\mathcal{A}}(k - s_i \times j)$$

Finally, note that each application can have its own scale factor, and that only "large" applications need to be scaled.

7.4 Illustration and Performance Evaluation

In this section, a simple example demonstrating the use of the proposed model is presented. In the example, we consider the traffic load that could be offered by a hypothetical application at some link. For simplicity in the analysis, we will assume a periodic traffic behavior and ignore traffic jitter and computation-time variations at the hosts. These issues will be addressed later.

The scenario is diagrammed in Figure 7.2. In the figure, host M periodically sends a computation request to the slave hosts S. The request passes through the intermediate node N which multicasts the message to the slave hosts. After the slave hosts finish their computations (which we will assume to take uniform time), they send their identically-sized result messages to M via N. The link of concern is the first link emanating from N heading toward M. Obviously, the arrivals of the result messages at N and their demands for bandwidth at L will be strongly correlated.

The complete characterization of the traffic pattern from the slaves S to M involves many parameters. However, we can construct the bandwidth demand "probabilities" from the parameters given in Table 7.1. From the given parameters, the resulting "probability" table is the following:

r:	0	1	2	3	4	5
$Q(r)$:	$\frac{2660}{3000}$	$\frac{25}{3000}$	$\frac{15}{3000}$	$\frac{15}{3000}$	$\frac{25}{3000}$	$\frac{260}{3000}$

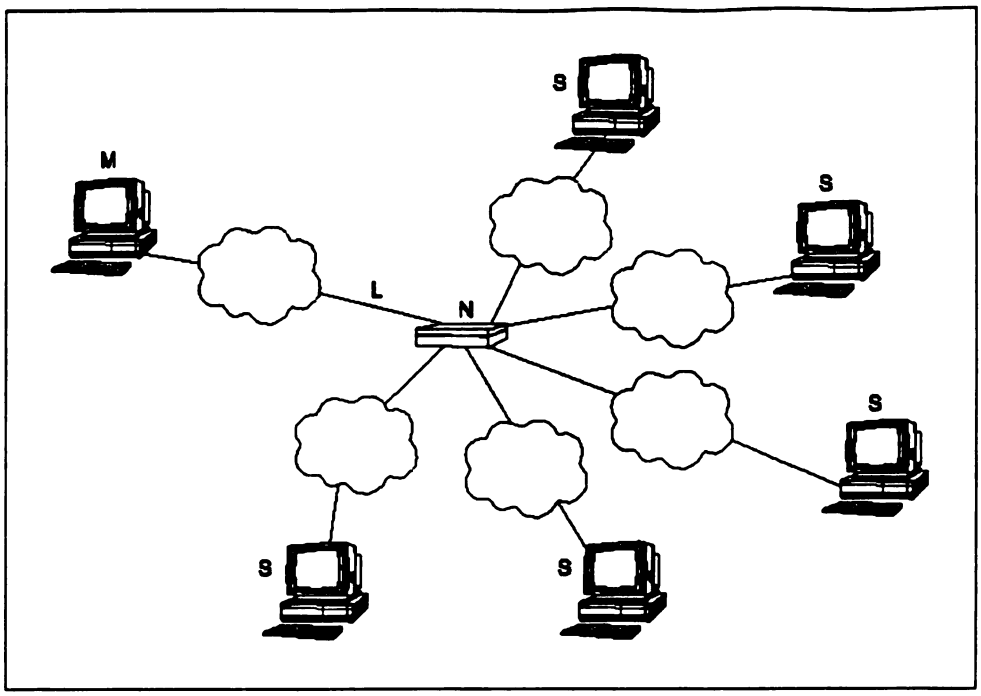

Figure 7.2: Example distributed application diagram. M is the master host, N is an intermediate node, and S is the set of slave hosts sending messages to M via N. The link of concern is marked L.

Even though the above table was obtained with the assumption of perfect synchronization (in the network and among the hosts), we believe that it is an adequate example for illustration purposes. The variations in network delay and computing time would more likely decrease the "perceived burstiness" by the link because the arrivals will be more scattered in time.[4]

To evaluate the CAC scheme, we compare its performance in terms of maximum utilization gained to the performance of two alternative approaches. The first alternative approach is the "optimistic assumption" approach, where the individual sources within an application are assumed to be independent of each other, in addition to the applications being independent among themselves. The second alternative approach is the "conservative assumption" approach, which assumes the maximum possible

[4]In contrast, consider the opposite extreme in which the timings are perfectly "random." In this case, the variations in the network delay and computing time are likely to reduce the "randomness" and increase overlapping, and thus be harmful to the network (if not taken into account).

Table 7.1: Table of parameters for the given example application.

Parameter	Value
number of slaves	5
cycle length (period)	3000 time units
S-to-M message length	300 time units
number of bandwidth units required	1 per message
time between earliest and second arrival	5 time units
time between second and third arrival	5 time units
time between third and fourth arrival	10 time units
time between fourth and last arrival	20 time units

overlap among the messages belonging to the same application.

To compare the approaches, heterogeneous applications similar to the given example application (i.e., Figure 7.2) were generated. The applications were given different numbers of sources (ranging from 2 to 10), different degrees of message overlap, and different cycle lengths. The amount of overlap ranged from almost complete overlap (where the messages arrived almost simultaneously) to minimal all-sources overlap (where the last message barely overlapped with the first). For simplicity, each source was assumed to be active 10% of the time (i.e., each cycle). For each approach, the appropriate CAC procedure was applied to determine the maximum number of independent applications admissible given a certain number of bandwidth units and a target BBP of 0.001. The resulting bandwidth utilizations are summarized in Figure 7.3.

From the figure, observe that the "correlated" approach results in a utilization between the "conservative" and "optimistic" approaches, as expected. In theory, larger utilizations closer to the optimistic approach would be attainable by the correlated approach if the applications had lower correlation (i.e., less expected overlap). Conversely, the utilizations would be lower for the correlated approach if the applications had higher correlation.

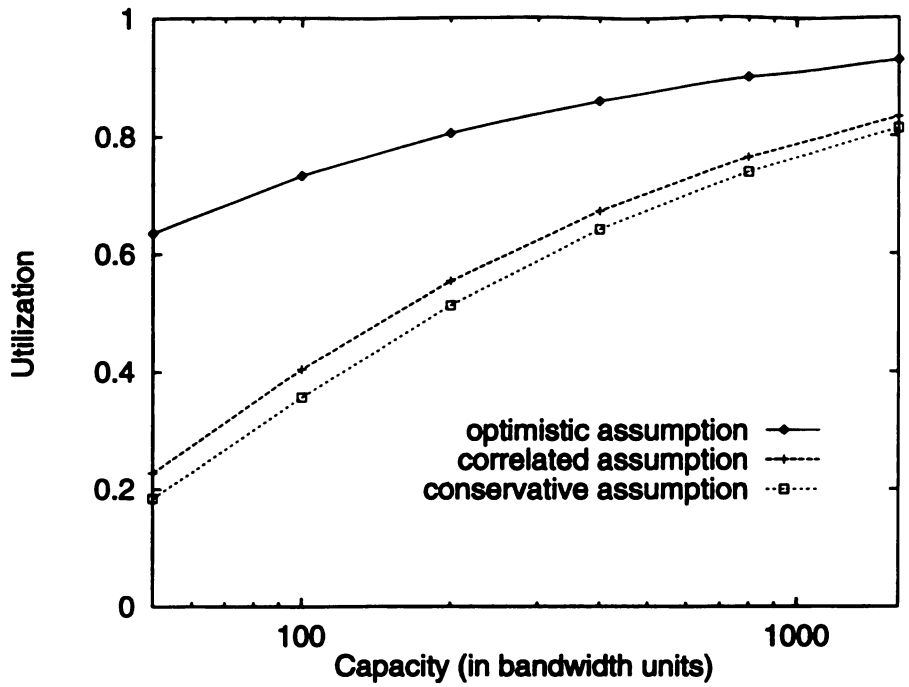

Figure 7.3: Utilizations allowed by the three CAC approaches. The target BBP was set to 0.001. The "correlated assumption" approach uses the proposed scheme; the "optimistic" approach assumes that all sources are independent; and the "conservative" approach assumes maximal overlap among messages from the same application.

Figure 7.3 gives only half the story. One benefit of the "correlated" approach is that it allows us to achieve a higher utilization than the simpler "conservative" approach would allow. The other benefit is that it is more likely to maintain the correct BBP compared to the optimistic approach that assumes that all the sources are independent. This is illustrated in Figure 7.4 which shows that the optimistic approach results in much higher BBPs if the correlation is indeed there.

Note that the correlation of concern in the model is that on the likelihood that one source is active given that another source (in the same application) is also active. In the burst-oriented traffic control approach, where bursts from "on-off" sources are not queued as such, the (intra-source) correlation affecting the interarrival times (from the same source) does not appear to be relevant.

Using simulations with a wide range of parameters, it has been demonstrated that

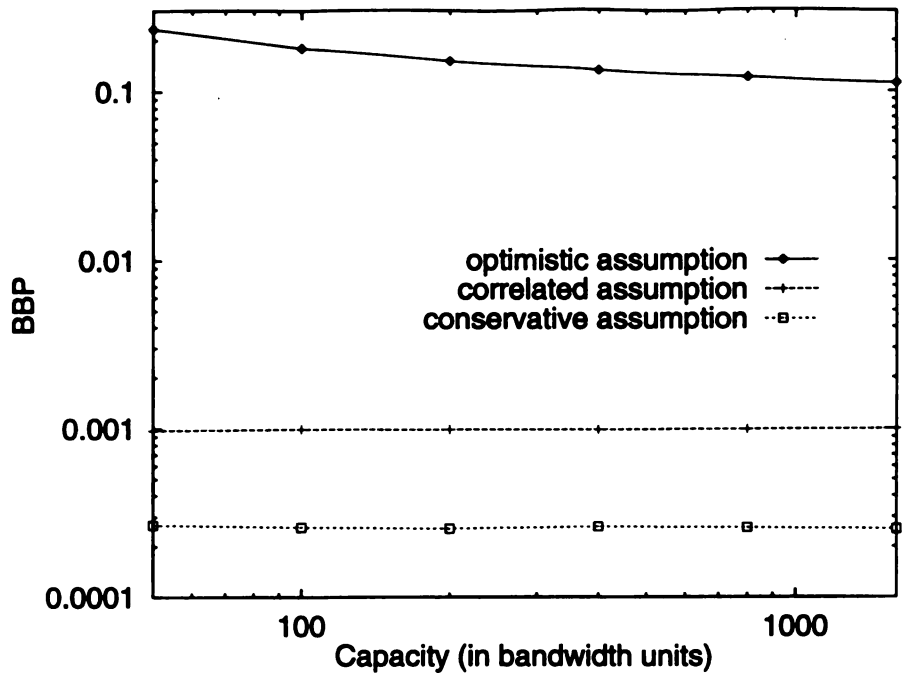

Figure 7.4: Calculated BBPs for the three CAC approaches assuming that the actual traffic follows the correlated model.

the proposed CAC scheme does indeed control the BBP for the given traffic model. Figure 7.5 shows the simulation results for a set of synthetic workloads consisting of heterogeneous applications similar to the given example application (i.e., Figure 7.2). To facilitate the data collection, the applications in the simulations had five sources each. This way, the trailing burst from each application cycle experienced the same highest BBP which was supposed to be controlled by the CAC scheme. The applications, however, had different cycle lengths and message overlap patterns.

For each capacity value of interest, a set of admitted applications was generated and then ten independent runs (with different initial states/starting points) were executed from that workload. Three BBPs were monitored in each run: the BBP of the leading burst of each application cycle, the BBP of the trailing burst of each application cycle, and the BBP of all bursts. The individual results and the averages are plotted in the figure, which shows that the BBPs were kept below the target BBP of 0.001 as intended. The figure also shows that the BBPs of the trailing bursts were

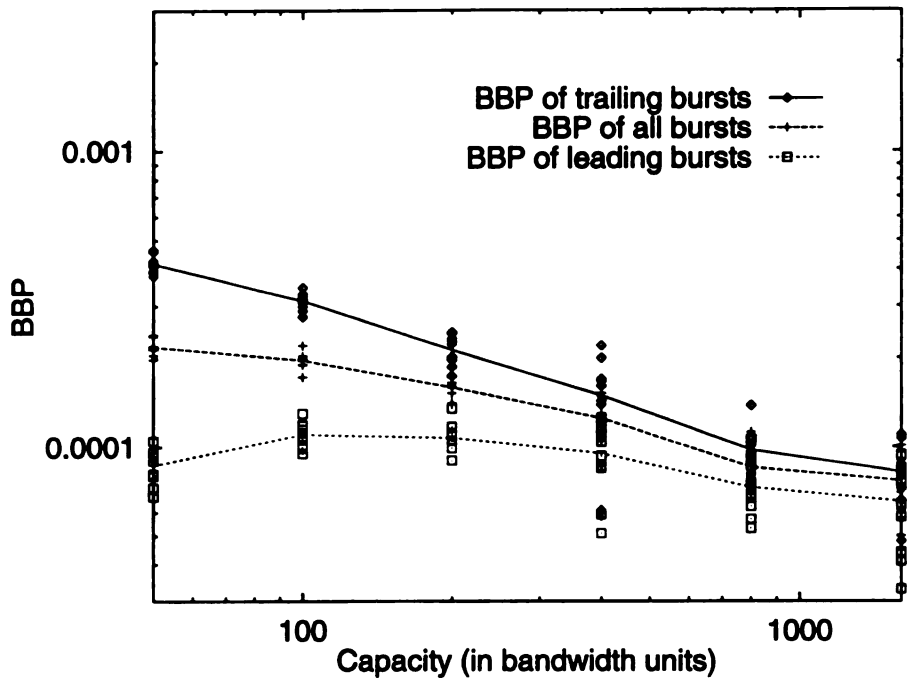

Figure 7.5: Measured BBPs from simulations using workloads admitted by the correlation-aware CAC scheme. The target BBP was set to 0.001 as in the previous figures. The points plot the results of individual runs and the lines connect the corresponding averages. "Leading" and "trailing" bursts refer to the first and last bursts of each application cycle.

generally higher than the BBPs of the other bursts.

Note that the measured BBPs were all below the target BBP and that the BBPs tended to decrease as the capacity increased. These observations were also made from the simulation results described in Section 6.5. These phenomena can be attributed to the conservative nature of the CAC scheme, which does not take burst blocking into account. Burst blocking tends to occur in bunches, and dropping blocked bursts helps to reduce the likelihood of burst blocking in the near future. In a higher capacity system, this effect becomes more pronounced because of the larger number of states (i.e., number of bandwidth units in use by bursts) that it has. The large number of states results in long periods without threat of burst blocking between short periods of high likelihood of burst blocking. Dropping blocked bursts effectively decreases the load during those periods where it would help the most.

Just like in Section 6.5, additional simulations were performed in order to provide evidence that the lower-than-expected BBP and the decreasing trend of the BBPs were mainly due to the dropping of blocked bursts. Figure 7.6 shows the results using the same workloads as in Figure 7.5, except for the use of "persistent" bursts. That is, in the new simulations, blocked bursts were not necessarily dropped completely but instead allowed to stay in the system and later use bandwidth released by departing bursts. This does not mean that the blocked bursts were queued; rather, only the front part of a blocked burst was effectively dropped until a bandwidth unit was released by a departing burst. Note that the measured BBPs of the trailing bursts were closer to the target BBP in the new simulations.

Finally, we examine two sets of simulation results evaluating the sensitivity of the CAC scheme to the accuracy of the input application demand probability tables.

Figure 7.7 shows the effect of increasing the traffic load given to the system beyond that "admitted" by the CAC scheme. The results were obtained by simply increasing all the message lengths by the given percentage (without increasing the total cycle

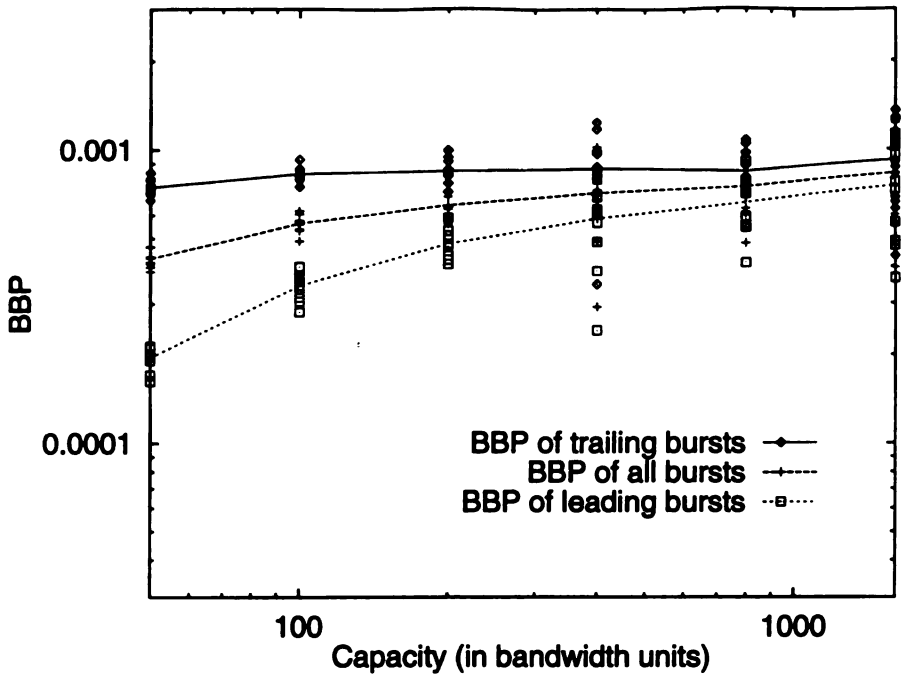

Figure 7.6: Results of the same simulations in the previous figure, but with bursts that were not dropped when blocked. In these simulations, blocked bursts were allowed to persist in the system and later use bandwidth released by departing bursts.

time) in the simulations. The "load" seems to be the parameter to which the BBP is most sensitive. Fortunately, the offered load *per source* is relatively easy to control with appropriate traffic policing.

Figure 7.8 shows the effect of increasing the amount of message overlap within applications *without increasing the load*. The BBP appears to be sensitive to such increase in overlap but not much. The results were obtained by reducing the delays between messages (in the same cycle) by the given percentage, making their arrival times closer together. This has the effect of increasing the probability of more messages overlapping without increasing the load.

For example, if the cycle time for the scenario in Figure 7.9(a) is 100 time units, then its bandwidth demand probability table is

r:	0	1	2	3
$Q(r)$:	0.84	0.06	0.06	0.04

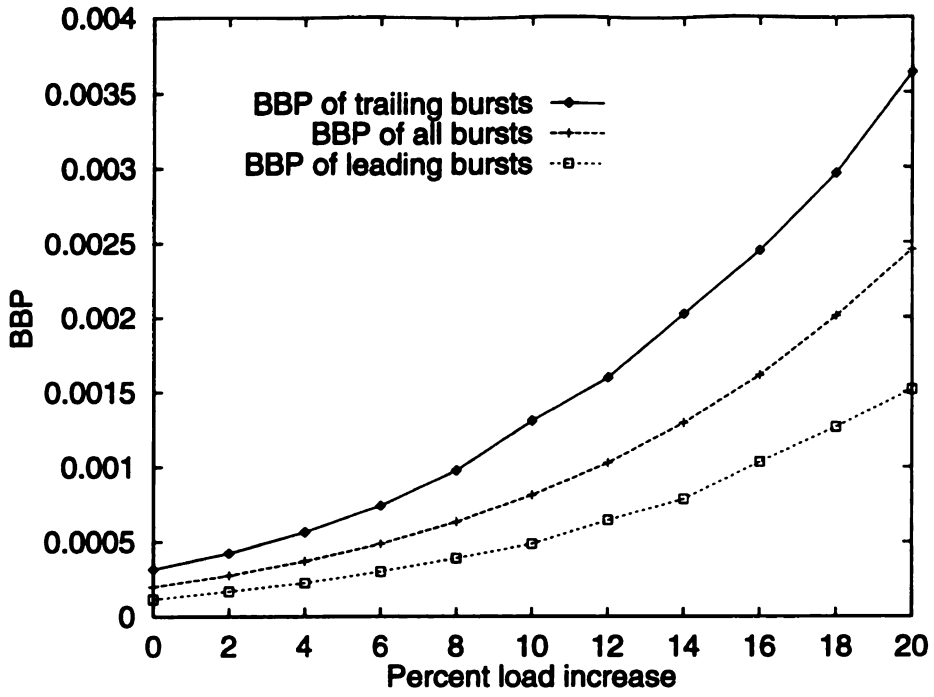

Figure 7.7: BBPs resulting from simulations where the traffic load was effectively increased beyond that specified to the CAC scheme. The capacity was 100 bandwidth units, the target BBP was 0.001, and the (unadjusted) workload was that admitted by the "correlated" approach in the previous simulations.

This table would be used by the application generator (and CAC scheme) in the simulation. However, the (burst/message) traffic generator would use the adjusted delays shown in Figure 7.9(b), which corresponds to a demand probability table of

r:	0	1	2	3
$Q(r)$:	0.87	0.03	0.03	0.07

which is effectively more bursty.

It should be noted that other types of variations in the traffic patterns were experimented with, but the above were the only ways in which the resulting BBP was observed to increase. In general, the BBP appears to be tolerant of variations to the delays between messages, message lengths, and cycle durations, as long as the averages of these were maintained in the long run.[5]

[5]Needless to say, the independence between applications has to be maintained in all cases.

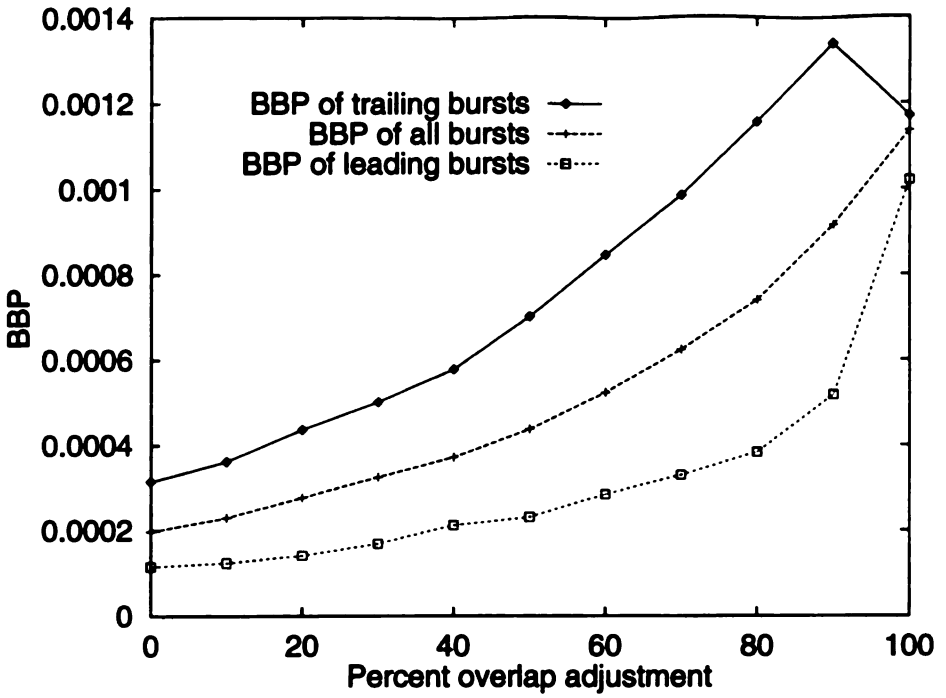

Figure 7.8: BBPs resulting from simulations where the intra-application overlap was effectively increased beyond that specified to the CAC scheme. The capacity was 100 bandwidth units, the target BBP was 0.001, and the (unadjusted) workload was that admitted by the "correlated" approach in the previous simulations.

7.5 Summary

In this chapter, a general approach for modeling the bandwidth demands of correlated multiple sources belonging to a distributed application is presented. The model is based on specifying (or estimating) the aggregate bandwidth demand distribution of a set of sources over a shared communication link. The distribution is specified as an empirical probability mass function in tabular form. This table is then combined with other such tables using the tractable algorithm given in Section 7.3. The result is a global (i.e., including all pertinent applications using the link) bandwidth demand distribution from which the burst blocking probability can be computed.

Using a simple example of a correlated application, the model and CAC scheme are shown to allow a utilization that falls between a more conservative but less bandwidth-efficient approach and a more overly optimistic approach that does not take any

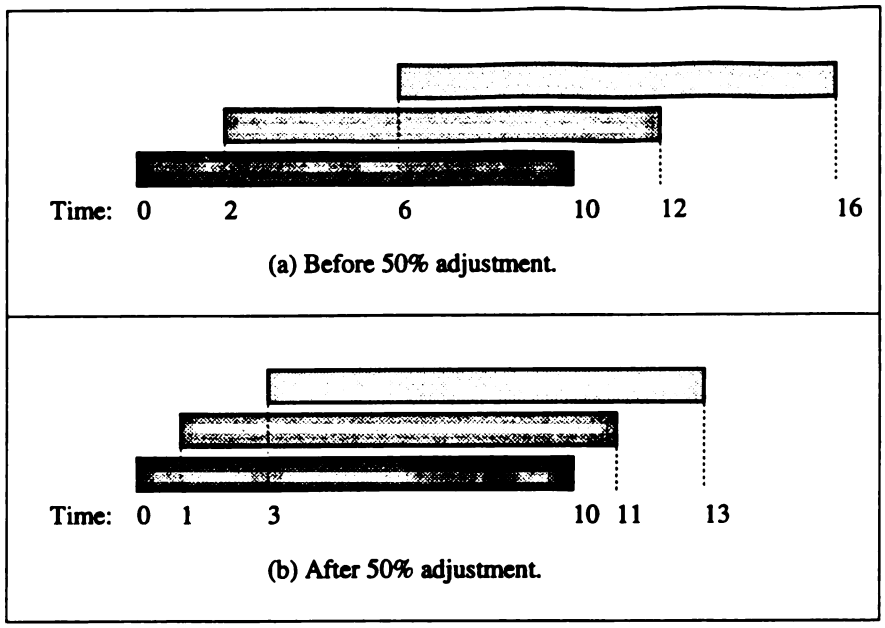

Figure 7.9: Diagram of the message overlap profile of an application, with three sources using the same link of concern. Figure (b) shows the effect of decreasing the relative message delays in (a) by 50%.

correlation into account. The penalties for using these other approaches are lower bandwidth utilizations for the conservative approach, on the one hand, and higher burst loss rates for the optimistic approach, on the other hand. These penalties are shown to be significant in Section 7.4.

That the CAC scheme works in controlling the BBP as intended has been demonstrated using simulations. In addition, additional simulations provide evidence that the CAC scheme is quite tolerant to variations in the traffic patterns as long as long-term averages in the traffic parameters are maintained. However, the technique appears to be specially sensitive to *increases* in the traffic load (which is to be expected from all but the most conservative approaches). In addition, it appears to be sensitive (to a lesser extent) to *increases* in intra-application overlap.

Chapter 8

Conclusion

8.1 Summary of Contributions

A comprehensive framework for burst-level traffic control in ATM networks has been presented. The framework provides mechanisms for supporting burst-oriented connections in which bursts of cells are granted (or denied) bandwidth on-the-fly. This allows such connections to start transmitting bursts without having to fully reserve bandwidth beforehand. Two types of burst-oriented connections are supported: thin logical connections (TLCs) which have zero bandwidth between bursts and fat logical connections (FLCs) which have non-zero reserved bandwidth between bursts (i.e., high-activity periods). The framework includes mechanisms to support the bundling of these connections into virtual paths.

Congestion control in this kind of framework is performed at two levels. Burst-level congestion control consists of properly forwarding or dropping bursts based on the availability of bandwidth and optionally on the priority of the bursts. To get any reasonable performance at this level, hardware support will be required from the ATM switching elements. Call-level congestion control is performed at connection establishment by a call admission control (CAC) scheme. The CAC scheme either

admits or rejects connection establishment requests based on whether the anticipated burst-level congestion will be acceptable to all parties concerned.

Three new CAC schemes have been presented and evaluated in this dissertation. The first is a simple scheme based on the Erlang loss system model. Its primary merit is its low computational requirement compared to other schemes using more general traffic models. Despite its simplicity, it appears to be potentially adequate under a wide range of conditions.

The second CAC scheme is a burst-level priority scheme based on a more general (non-Poisson and non-renewal) traffic source model. The scheme as presented supports two priority levels for the connections (or bursts). The low and high-priority connections share the same pool of resources but with the low-priority bursts constrained to use only a subset of the shared bandwidth allocation units. This results in a higher blocking rate for the low-priority connections. The sharing of resources this way results in higher bandwidth utilization (i.e., more admitted connections) than when the two classes of connections are either treated identically (i.e., with the same *low* BBP) or managed separately (i.e., do not share resources). This two-level scheme can be extended to handle more than two priority levels, but the result would be significantly more complex (computationally) with little expected additional performance benefit.

The third CAC scheme is an approach that can take into account known traffic correlation among sources. Such correlation is likely to exist among a set of traffic sources associated with a parallel distributed computing application. Most CAC schemes (burst-oriented or otherwise) assume independent traffic sources and this type of correlation will likely degrade the performance of the network unacceptably. The proposed "correlation-aware" scheme is based on a general "set-of-sources" model to capture the correlation and an algorithm similar to that used in the second CAC scheme to compute the aggregate bandwidth demand probabilities. When compared

to two alternative approaches, one based on an optimistic "independent sources" assumption and another based on a conservative "maximum overlap" assumption, the proposed scheme appears to strike the proper balance in terms of loss rate and bandwidth utilization. This scheme may be extended to support burst-level priorities like in the second CAC scheme but with the restriction that each set of correlated sources have the same priority.[1]

The following is a summary of the advantages of the burst-oriented approach over the more common cell-oriented approach in handling bursty traffic.[2]

- The loss rate at the burst level may be a more useful performance metric than the cell loss rate as it has direct impact on the performance of end-to-end protocols handling protocol data units larger than a cell.

- In periods of congestion, cell losses will be shared by all active sources sharing a link (cell loss priority aside) in the cell-oriented approach. In contrast, cell losses will be experienced by the same (blocked) bursts in the burst-oriented approach. In fact, the burst-oriented framework provides a form of "congestion relief" by deliberately dropping the cells of blocked bursts (cf. [5, 65]).

- The burst-oriented approach with "on-the-fly" bandwidth reservation requires only enough buffers to support equivalent CBR connections (circuit emulation) in the presence of cell delay jitter. However, if cell queueing beyond that required for absorbing jitter is to be considered in the cell-oriented approach, then the bandwidth demand analysis would be considerably more complicated. This is the case because it would depend on buffer sizes, burst length distributions, and cell interarrival processes.

[1] Of course, it is mathematically possible to remove this restriction and also to increase the number of priority levels, but the resulting procedure would be computationally impractical.

[2] Note than in the following, we are primarily concerned with cell losses due to congestion. Cell losses due to physical layer and switch hardware memory bit errors will be experienced equally by both approaches.

8.2 Future Work

The original motivation and the main emphasis of this dissertation in the use of the burst-oriented approach in the transmission of bursty *data* traffic. However, with additional work, multimedia traffic, specifically VBR compressed packet video traffic, may also be supported by the framework. The two primary benefits of the burst-oriented approach are its low buffer requirements and its low (queueing) latency. As already pointed out, burst-oriented connections behave much like CBR connections (circuit emulation), except for the head and tail processing. However, a few issues have to be addressed as follows.

First, the inherent periodicity of video traffic has to be handled properly. Periodicity may be good or bad. In the best case, interleaving of bursts from different sources may occur, resulting in very little and fair burst losses. In the worst case, a set of sources could be caught in a "lock-step" pattern of synchronized arrivals that would result in excessive and unfair burst losses. Note that a connection would typically have to pass through multiple switches and compete with a different set of connections at each switch. Therefore, for a particular connection, a favorable timing ("phase") at one switch may be disastrous in another switch in the path. Beware of Murphy's Law.

One general solution to the periodicity problem is to introduce randomness in the phase or timing of the frames. In some video coding-decoding schemes (codecs) such as MPEG [26], compressed frames come in different types and sizes. Naturally the bandwidth allocated to the connection would be dictated by the size of the largest frames. Therefore, for most of the smaller frames, the time to transmit them would be smaller than the frame period. This gives the sender some room for randomizing the frame transmission times. This approach, however, does not solve the more subtle problem of periodicity within larger time scales. For example, in MPEG, if the larger

I-frames are sent every 12 frames, then a higher-level 12-frame-long period of high activity will result. After the first frame has been transmitted, the source does not have much flexibility in positioning the subsequent I-frames (unless it has control over the encoder which would not be the case in the transmission of precompressed, prerecorded video).

Another issue of concern is the debatable (in)appropriateness of burst-level discarding in video (as opposed to cell-level discarding during periods of congestion). Depending on the encoding scheme, it is possible that losing a cell in a frame may not necessarily invalidate the whole frame. That is, the remaining cells in the frame may still be useful to the decoder. For example, this would be the case if the data in the cell applied to a region of the picture rather than the picture as a whole. However, tolerance to single cell losses requires the additional overhead of sequence numbers in the cells in order for the decoder to identify the lost cells. In any case, if whole bursts are to be discarded in periods of congestion, then an effort must be made to minimize the perceptability of the losses.

One approach is to use burst-level priorities. Multi-layered MPEG coders have been designed that separate the output components of the standard coder into high-priority and low-priority streams (e.g., [58]). The high-priority stream carries most of the "energy" in the picture, and the scheme is able to tolerate higher loss rates in the low-priority stream. These streams may then be carried in different bursts with different priorities. Alternatively or in addition to the above, the more important frames (such as I-frames which have longer temporal significance) can be given higher priority than the less important frames. Moreover, to further minimize the losses of high-priority frames, preemption (of low-priority bursts) may be supported. The above may further be enhanced with limited buffering such as the technique (intended for voice) used in burst-switching [31]. This allows the bulk of a burst to get through even if part of it gets clipped.

The above suggestions assume the use of a TLC to carry the video traffic. However, it is also possible to use an FLC instead. One possible approach is to start with a video codec that is designed to produce CBR output such as the $p \times 64$ *kbit/s* video standard [48]. The constant bit rate output is obtained by the coding scheme by dynamically controlling the degree of quantization (least-significant-bit truncation) in the compressor. The drawback of this approach is that it results in a *variable quality* output. To produce constant quality output, an FLC may be used to carry the truncated information in a burst-by-burst fashion in addition to the CBR component. This general idea of sending the high-priority stream in a bandwidth-reserved manner while sending the low-priority stream in a less resource-demanding manner has been suggested in [58].

Bibliography

Bibliography

[1] Bandula W. Abeysundara and Ahmed E. Kamal. High-speed local area networks and their performance: A survey. *ACM Computing Surveys*, 23(2):221–264, June 1991.

[2] J. M. Aein. A multi-user-class, blocked-calls-cleared, demand access model. *IEEE Trans. Commun.*, COM-26(3):378–385, March 1978.

[3] Stanford R. Amstutz. Burst switching—an introduction. *IEEE Commun. Mag.*, pages 36–42, November 1983.

[4] Heinrich Armbrüster and Klaus Wimmer. Broadband multimedia applications using ATM networks: High-performance computing, high-capacity storage, and high-speed communication. *IEEE J. Sel. Areas Commun.*, 10(9):1382–1396, December 1992.

[5] G. Armitage and K. Adams. Packet reassembly during cell loss. *IEEE Network*, 7(5):26–34, September 1993.

[6] E. Arthurs and J. S. Kaufman. Sizing a message store subject to blocking criteria. In M. Arato, A. Butrimenko, and E. Gelenbe, editors, *Performance of Computer Systems*, pages 547–564. North-Holland, 1979.

[7] Jaime Jungok Bae and Tatsuya Suda. Survey of traffic control schemes and protocols in ATM networks. *Proc. of the IEEE*, 79(2):170–189, February 1991.

[8] Krishna Bala, Israel Cidon, and Khosrow Sohraby. Congestion control for high speed packet switched networks. In *Proc. INFOCOM '90*, pages 520–526. IEEE, 1990.

[9] Flavio Bonomi and Kerry W. Fendick. The rate-based flow control framework for the available bit rate ATM service. *IEEE Network*, pages 25–39, March 1995.

[10] Laura J. Bottomley and Arne A. Nilsson. Traffic characterization in a wide area network. In Harry Perros, editor, *High-Speed Communication Networks*, pages 213–224. Plenum Press, 1992.

[11] Jean-Yves Le Boudec. The Asynchronous Transfer Mode: a tutorial. *Computer Networks and ISDN Systems*, 24:279–309, 1992.

[12] Andreas D. Bovopoulos. Performance evaluation of a traffic control mechanism for ATM networks. In *Proc. INFOCOM '92*, pages 469–478. IEEE, 1992.

[13] Pierre E. Boyer and Didier P. Tranchier. A reservation principle with applications to the ATM traffic control. *Computer Networks and ISDN Systems*, 24:321–334, 1992.

[14] Milena Butto', Elisa Cavallero, and Alberto Tonietti. Effectiveness of the "leaky bucket" policing mechanism in ATM networks. *IEEE J. Sel. Areas Commun.*, 9(3):335–342, April 1991.

[15] Ramón Cáceres et al. Characteristics of wide-area TCP/IP conversations. *Computer Commun. Review*, 21(4):101–112, September 1991.

[16] Nim K. Cheung. The infrastructure for gigabit computer networks. *IEEE Commun. Mag.*, pages 60–68, April 1992.

[17] Israel Cidon, Inder Gopal, and Adrian Segall. Fast connection establishment in high speed networks. *Computer Commun. Review*, 20(4):287–296, September 1990.

[18] Douglas E. Comer. *Internetworking with TCP/IP*, volume I: Principles, Protocols, and Architecture. Prentice-Hall, second edition, 1991.

[19] Simon Crosby. In-call renegotiation of traffic parameters. In *Proc. INFOCOM '93*, pages 638–646. IEEE, March 1993.

[20] M. Decina et al. Bandwidth assignment and virtual call blocking in ATM networks. In *Proc. INFOCOM '90*, pages 881–888. IEEE, 1990.

[21] W. A. Doeringer et al. Fast connection establishment in large-scale networks. In *Proc. INFOCOM '93*, pages 489–496. IEEE, 1993.

[22] Bharat Doshi. Performance of in-call buffer/window allocation schemes for short intermittent file transfers over broadband packet networks. In *Proc. INFOCOM '92*, pages 2463–2471. IEEE, May 1992.

[23] Bharat Doshi and Harry Heffes. Performance of an in-call buffer-window reservation/allocation scheme for long file transfers. *IEEE J. Sel. Areas Commun.*, 9(7):1013–1023, September 1991.

[24] Bharat T. Doshi and Pravin K. Johri. Communication protocols for high speed packet networks. *Computer Networks and ISDN Systems*, 24:243–273, 1992.

[25] Fore Systems, Inc., 1000 Gamma Drive, Suite 504, Pittsburgh, PA 15238. *ForeRunner(TM) ASX-100 ATM Switch Architecture Manual (Release 2.1)*, 1993.

[26] Didier Le Gall. MPEG: A video comnpression standard for multimedia applications. *Commun. of the ACM*, 34(4):46–58, April 1991.

[27] G. Gallassi, G. Rigolio, and L. Fratta. ATM: Bandwidth assignment and bandwidth enforcement policies. In *Proc. GLOBECOM '89*, pages 1788–1793. IEEE, 1989.

[28] Roch Guérin, Hamid Ahmadi, and Mahmoud Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE J. Sel. Areas Commun.*, 9(7):968–981, September 1991.

[29] Levent Gün and Roch Guérin. An overview of bandwidth management procedures in high-speed networks. In Harry Perros, editor, *High-Speed Communication Networks*, pages 35–45. Plenum Press, 1992.

[30] Ibrahim W. Habib and Tarek N. Saadawi. Controlling flow and avoiding congestion in broadband networks. *IEEE Commun. Mag.*, pages 46–53, October 1991.

[31] E. Fletcher Haselton. A PCM frame switching concept leading to burst switching network architecture. *IEEE Commun. Mag.*, pages 13–19, September 1983.

[32] Harry Heffes and David M. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Sel. Areas Commun.*, SAC-4(6):856–867, September 1986.

[33] Douglas S. Holtsinger and Harry G. Perros. Performance of the buffered leaky bucket policing mechanism. In Harry Perros, editor, *High-Speed Communication Networks*, pages 47–69. Plenum Press, 1992.

[34] Chengchang Huang and Philip K. McKinley. Communication issues in parallel computing across ATM networks. *Parallel & Distributed Technology*, 2(4):73–86, 1994.

[35] Joseph Y. Hui. Resource allocation for broadband networks. *IEEE J. Sel. Areas Commun.*, 6(9):1598–1608, December 1988.

[36] A. Iwata et al. ATM connection and traffic management schemes for multimedia internetworking. *Commun. of the ACM*, 38(2):72–89, February 1995.

[37] Raj Jain. FDDI: Current issues and future plans. *IEEE Commun. Mag.*, pages 98–105, September 1993.

[38] Raj Jain and Shawn A. Routhier. Packet trains—measurements and a new model for computer network traffic. *IEEE J. Sel. Areas Commun.*, 4(6):986–995, September 1986.

[39] Per Jomer. Connection caching of traffic adaptive dynamic virtual circuits. *Computer Commun. Review*, 19(4):13–24, September 1989.

[40] Takashi Kamitake and Tatsuya Suda. Evaluation of an admission control scheme for an ATM network considering fluctuations in cell loss rate. In *Proc. GLOBECOM '89*, pages 1774–1780. IEEE, 1989.

[41] Joseph S. Kaufman. Blocking in a shared resource environment. *IEEE Trans. Commun.*, COM-29(10):1474–1481, October 1981.

[42] Gary C. Kessler and David A. Train. *Metropolitan Area Networks: Concepts, Standards, and Services.* McGraw-Hill, 1992.

[43] Leonard Kleinrock. The latency/bandwidth tradeoff in gigabit networks. *IEEE Commun. Mag.*, pages 36–40, April 1992.

[44] H. T. Kung and Alan Chapman. The FCVC (flow-controlled virtual channels) proposal for ATM networks. In *Proc. 1993 Int. Conf. on Network Protocols*, pages 116–127, 1993.

[45] H. T. Kung and Robert Morris. Credit-based flow control for ATM networks. *IEEE Network*, pages 40–48, March 1995.

[46] Jim Kurose. Open issues and challenges in providing quality of service guarantees in high-speed networks. *Computer Commun. Review*, 23(1):6–15, January 1993.

[47] Will E. Leland et al. On the self-similar nature of Ethernet traffic. *Computer Commun. Review*, 23(4):183–193, October 1993.

[48] Ming Liou. Overview of the px64 kbit/s video coding standard. *Commun. of the ACM*, 34(4):59–63, April 1991.

[49] Basil Maglaris et al. Routing of voice and data in burst-switched networks. *IEEE Trans. Commun.*, 38(6):889–896, June 1990.

[50] Peter Newman. ATM local area networks. *IEEE Commun. Mag.*, pages 86–98, March 1994.

[51] Ioanis Nikolaidis and Raif O. Onvural. A bibliography on performance issues in ATM networks. *Computer Commun. Review*, 22(5):8–23, October 1992.

[52] Zhisheng Niu and Haruo Akimaru. Analysis of statistical multiplexer with selective cell discarding control in ATM systems. *IEICE Trans.*, E74(12):4069–4079, December 1991.

[53] Katia Obraczka, Peter B. Danzig, and Shih-Hao Li. Internet resource discovery services. *IEEE Computer*, pages 8–22, September 1993.

[54] Hirokazu Ohnishi, Tadanobu Okada, and Kiyohiro Noguchi. Flow control schemes and delay/loss tradeoff in ATM networks. *IEEE J. Sel. Areas Commun.*, 6(9), December 1988.

[55] Robert T. Olsen and Lawrence H. Landweber. Design and implementation of a fast virtual channel establishment method for ATM networks. In *Proc. INFOCOM '93*, pages 617–627. IEEE, March 1993.

[56] Raif O. Onvural and Ioanis Nikolaidis. Routing in ATM networks. In Harry Perros, editor, *High-Speed Communication Networks*, pages 139–150. Plenum Press, 1992.

[57] Teunis Ott, T. V. Lakshman, and Ali Tabatabai. A scheme for smoothing delay-sensitive traffic offered to ATM networks. In *Proc. INFOCOM '92*, pages 776–785. IEEE, 1992.

[58] P. Pancha and M. El Zarki. Prioritized transmission of variable bit rate MPEG video. In *Proc. GLOBECOM '92*, pages 1135–1139. IEEE, December 1992.

[59] Pramod Pancha and Magda El Zarki. Bandwidth requirements of variable bit rate MPEG sources in ATM networks. In *Proc. INFOCOM '93*, pages 902–909. IEEE, 1993.

[60] M. De Prycker, R. Peschi, and T. Van Landegem. B-ISDN and the OSI protocol reference model. *IEEE Network*, 7(2):10–18, March 1993.

[61] K. K. Ramakrishnan and Peter Newman. Integration of rate and credit schemes for ATM flow control. *IEEE Network*, pages 49–56, March 1995.

[62] Erwin P. Rathgeb. Modeling and performance comparison of policing mechanisms for ATM networks. *IEEE J. Sel. Areas Commun.*, 9(3):325–334, April 1991.

[63] Michael J. Rider. Protocols for ATM access networks. *IEEE Network*, pages 17–22, January 1989.

[64] James W. Roberts. A service system with heterogeneous user requirements—application to multi-services telecommunications systems. In G. Pujolle, editor, *Performance of Data Communications Systems*, pages 423–431. North-Holland, 1981.

[65] Allyn Romanow and Sally Floyd. Dynamics of TCP traffic over ATM networks. *Computer Commun. Review*, 24(4), October 1994.

[66] Reza Rooholamini, Vladimir Cherkassky, and Mark Garver. Finding the right ATM switch for the market. *IEEE Computer*, pages 16–28, April 1994.

[67] Jonathan Rosenberg et al. Multimedia communications for users. *IEEE Commun. Mag.*, pages 20–36, May 1992.

[68] Sheldon M. Ross. *Introduction to Probability Models.* Academic Press, fourth edition, 1989.

[69] Matthew N. O. Sadiku and A. S. Arvind. Annotated bibliography on Distributed Queue Dual Bus (DQDB). *Computer Commun. Review*, 24(1):21–35, January 1994.

[70] Hiroshi Saito. Call admission control in an ATM network using upper bound of cell loss probability. *IEEE Trans. Commun.*, 40(9):1512–1521, September 1992.

[71] Kiyoshi Shimokoshi. Evaluation of policing mechanisms for ATM networks. *IEICE Trans.*, E76-B(10):1341–1351, November 1993.

[72] John D. Spragins, Joseph L. Hammond, and Krzysztof Pawlikowski. *Telecommunications: Protocols and Design.* Addison-Wesley, 1991.

[73] William Stallings. *Data and Computer Communications.* Macmillan, third edition, 1991.

[74] William Stallings. *Local and Metropolitan Area Networks.* Macmillan, fourth edition, 1993.

[75] William Stallings. *ISDN and Broadband ISDN with Frame Relay and ATM.* Prentice-Hall, third edition, 1995.

[76] Hiroshi Suzuki, Tutomu Murase, Syohei Sato, and Takao Takeuchi. A burst traffic control strategy for ATM networks. In *Proc. GLOBECOM '90*, pages 874–878. IEEE, 1990.

[77] Toshikazu Suzuki. ATM adaptation layer protocol. *IEEE Commun. Mag.*, pages 80–83, April 1994.

[78] Lajos Takács. On Erlang's formula. *The Annals of Mathematical Statistics*, 40(1):71–78, 1969.

[79] Andrew S. Tanenbaum. *Computer Networks.* Prentice-Hall, second edition, 1988.

[80] Don Tolmie and John Renwick. HIPPI: Simplicity yields success. *IEEE Network*, 7(1):28–32, January 1993.

[81] Jonathan S. Turner. New directions in communications (or which way to the information age?). *IEEE Commun. Mag.*, 24(10):8–15, October 1986.

[82] Jonathan S. Turner. Managing bandwidth in ATM networks with bursty traffic. *IEEE Network*, pages 50–58, September 1992.

[83] Faramak Vakil and Hiroshi Saito. On congestion control in ATM networks. *IEEE LTS*, pages 55–65, August 1991.

[84] Jean Walrand. *Communication Networks: A First Course.* Aksen Associates, 1991.

[85] Patricia E. Wirth and Kathleen S. Meier-Hellstern. Traffic models for ISDN and B-ISDN users. In Harry Perros, editor, *High-Speed Communication Networks*, pages 205–211. Plenum Press, 1992.

[86] Nanying Yin, San-Qi Li, and Thomas E. Stern. Congestion control for packet voice by selective packet discarding. *IEEE Trans. Commun.*, 38(5):674–683, May 1990.