The University of Kansas

Information and Telecommunication Technology Center

A Technical Report of ITTC's
Networking and Distributed Systems Laboratory

Design and Performance Analysis of a Self-Configuring CAC for AAL2 with Load Estimation (SCALE)

Gopi Krishna Vaddi
and
David W. Petr

ITTC-FY2000-TR-15664-04

January 2000

Project Sponsor:
Sprint Corporation

Copyright © 2000:
The University of Kansas Center for Research, Inc.,
2291 Irving Hill Road, Lawrence, KS 66044-7541;
and Sprint Corporation.
All rights reserved.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The evolution of diverse, bandwidth hungry applications like high speed data, real time video and voice have created a demand for the design of an integrated network that would, for each application, emulate an application specific network. The emergence of faster DSPs (Digital Signal Processors), Optical Fiber and High-Speed Integrated Circuits have made the design of such networks possible. Such a network should be designed to satisfy the varied Quality of Service (QoS) requirements for each of the applications. Towards an effort to create such an integrated network evolved new types of networks like B-ISDN ATM and enhancements to traditional IP. Efforts are on to make these networks more application friendly.

Efficient resource allocation is among the most important design aspects in a integrated network. A good resource allocation scheme avoids congestion at the switches and routers in the network. Voice is an important part of today's networks because of the demand and revenue associated with it. It is expected to have the single largest traffic volume in future integrated networks. Also, voice traffic has the most stringent delay QoS requirement. Thus we need an effective and efficient means of carrying packet voice.

It has been demonstrated that ATM (Asynchronous Transfer Mode) can be used as an effective medium to transport integrated services, including voice. As part of the effort to make ATM effective for voice transport, a new ATM Adaptation Layer AAL2 has been designed specifically for voice applications.

1.1 Motivation and Problem Statement

In this work, we propose an effective resource (bandwidth) allocation scheme for Voice and Telephony over ATM (VToA) when AAL2 is used as the adaptation layer. Using AAL2, information from more than one user can be carried on a single ATM connection. This makes the resource allocation different from the traditional methods. *Call Admission Control* (CAC) has been recognized as one of the most important methods to prevent congestion and provide satisfactory QoS for all applications in ATM networks. Since AAL2 does user multiplexing, CAC for AAL2 users should take care of various different issues (explained later) that are not present when using other AALs. There arises a need for an effective Call Admission Control (CAC) strategy at the AAL2 level. In designing an effective CAC at the AAL2 level, we need to find answers to questions like *"What is*

the exact procedure to be employed for the CAC", "What is the nature of the Virtual Circuit (VC) to be used", "Given a load distribution, what is the size of the Virtual Circuit (VC) that needs to be used" etc. In this work, we try to understand the factors to be considered before answering the above questions and then proceed to answering them.

1.2 Organization of Report

- *Chapter 2* : We start with a background on packet voice. A brief discussion of voice over IP and voice over ATM is given and is followed by a discussion of AAL2.
- *Chapter 3* Chapter 3 speaks about the requirements for the AAL2 CAC. The answers to all the above questions are discussed in detail.
- *Chapter 4* The methods that can be used for bandwidth calculation for AAL2 users are discussed.
- *Chapter 5* The actual AAL2 CAC algorithm is explained with the flow chart and algorithm in Chapter 5. The simulations that have been done as proof of concept are discussed.
- *Chapter 6* Results showing the effectiveness of the algorithm are discussed here.
- *Chapter 7* We conclude the work and provide possible future work that is relevant to the work done.

Chapter 2

Background

In this chapter we give an introduction to voice transport over packet networks. We briefly discuss the protocol stack used to carry voice in IP/ATM networks followed by a discussion of AALs. We provide more details about AAL2. It is assumed that the reader has an introduction to IP and ATM.

2.1 Voice over Packet Networks

Towards providing an integrated voice, video and data network, a lot of research has been done on packetized voice. Carrying voice over packet networks is bandwidth efficient. Traditional voice is carried as 64 kb/s PCM. With the emergence of low bit rate coders, 64 kb/s bandwidth is not required for voice any longer. One of the latest coders (G.723.1) uses a coding rate of 5.3-6.3 kb/s, which is less than 10 percent of the 64 kb/s PCM. Due the asynchronous nature, packet networks can be used to carry voice irrespective of the coding rate, thus exploiting the advantage of low bit rate coders. Another important advantage of carrying voice over packet networks is the ability to incorporate *silence elimination*. It is a well known fact that telephone conversations consist of only 50 % active voice on either end [22] [6]. The transmission efficiency can be increased by transmitting only the active voice across the network. The two most popular packet networks carrying voice today are ATM and IP (Internet Protocol). We first go through voice over IP (VoIP) briefly and then proceed towards voice over ATM (VToA), which is the focus of this report.

2.1.1 Voice over IP Networks

Voice transport over IP networks mainly uses the well defined H.323 protocol stack [9]. Figure 2.1 shows the different layers and protocols that are part of the H.323 specification. The H.323 protocol stack uses the H.245 standard for passing control information and H.225 for passing RAS (registration, admission and signaling) information. The media channel is carried using RTP (real time protocol). RTCP (real time control protocol) is used as the media control channel. The problem with using IP for voice comes from the fact that IP is inherently connectionless and provides only best effort service. There have been a lot of developments in designing new protocols and adding new functionality to

the existing protocol stack for efficient voice transport. Also, efforts in incorporating QoS into IP networks have reached a matured stage. Discussions on VoIP standards are taking place in *avt* (Audio/Video Transport), *iptel* (IP Telephony), *mmusic* (Multiparty Multimedia Session Control) and *pint* (PSTN and Internet Internetworking) working groups of IETF (Internet Engineers Task Force). Though this is an interesting area of study, we will not discuss about this further as it is out of the scope of this report.

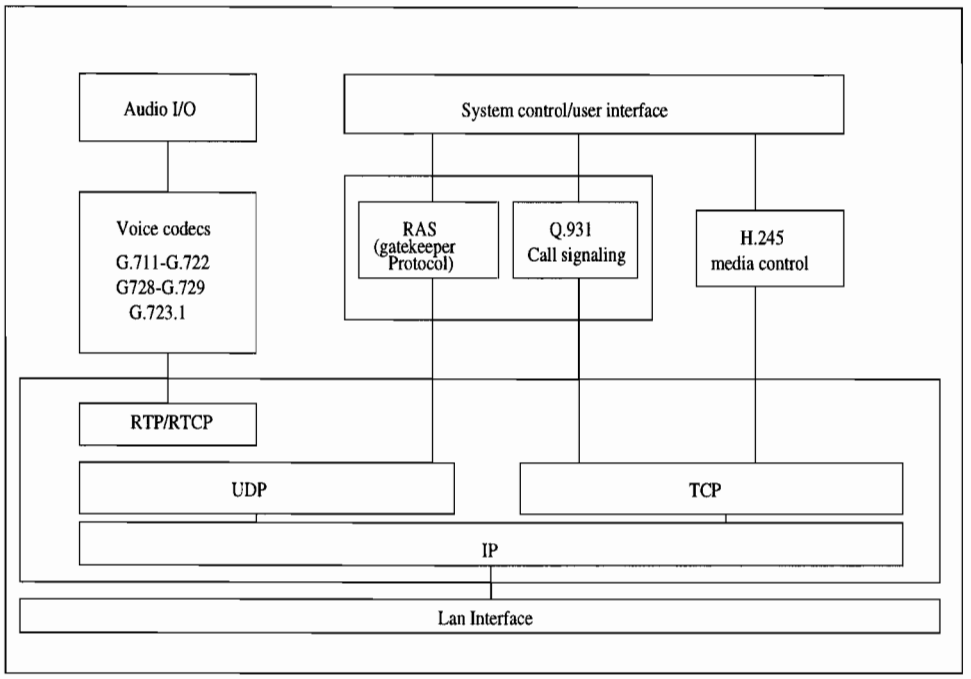


Figure 2.1: H.323 Protocol Stack

2.1.2 Voice over ATM

The well defined QoS capabilities of ATM make it ideal for voice transport. ATM promises to deliver a cost-effective solution for heterogeneous types of services by promoting dynamic network resource sharing, supporting bandwidth on demand and exploiting the statistical multiplexing property of aggregated traffic. ATM networks carry fixed-size (53 bytes) cells within the network irrespective of the applications being carried over it. The B-ISDN ATM protocol stack is shown in Figure 2.2. An ATM adaptation layer (AAL) maps the services offered by the ATM network to those required by the application. Different AALs have been defined [4], specifically AAL1, AAL3/4, AAL5, AAL2 (both old and new). Each of these AALs has been designed for a different purpose keeping a particular application in mind. Only AAL1, AAL5 and the latest AAL2 are discused here as reasonable choices for voice transport. AAL1 and AAL5 are briefly explained, and we then shift our focus onto AAL2.

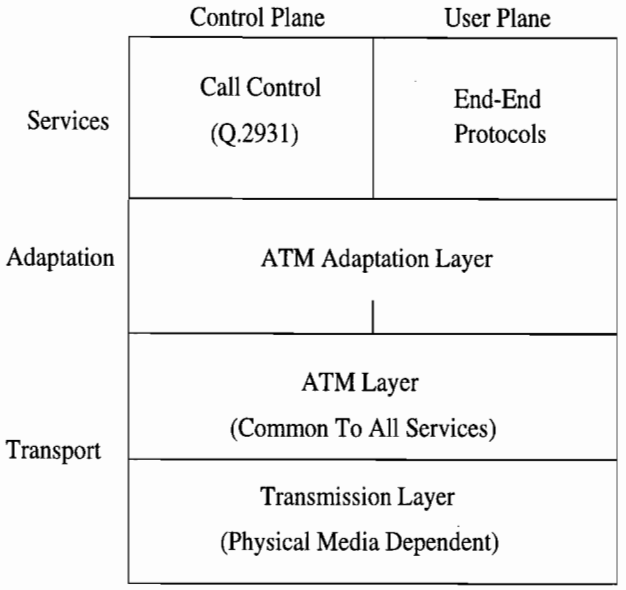


Figure 2.2: B-ISDN/ATM Protocol Stack

2.1.2.1 ATM Adaptation Layer 1

AAL1 provides the most basic functionality required of an adaptation layer. This layer is divided into Segmentation and Reassembly (SAR) sublayer and Convergence Sublayer (CS). The SAR sublayer header consists of 3 fields: a 1-bit Convergence Sublayer Indication (CSI), a sequence number of 3 bits and a 4-bit Sequence Number (and CSI) Protection (SNP). The CSI bit indicates the presence of the optional convergence sublayer. The structure of the AAL1 PDU is shown in Figure 2.3. The AAL1 payload can be a maximum of 47 bytes in the 48 byte ATM payload. AAL1 is good enough for carrying 64 kb/s PCM but performs badly for more sophisticated low bit rate coders [7].

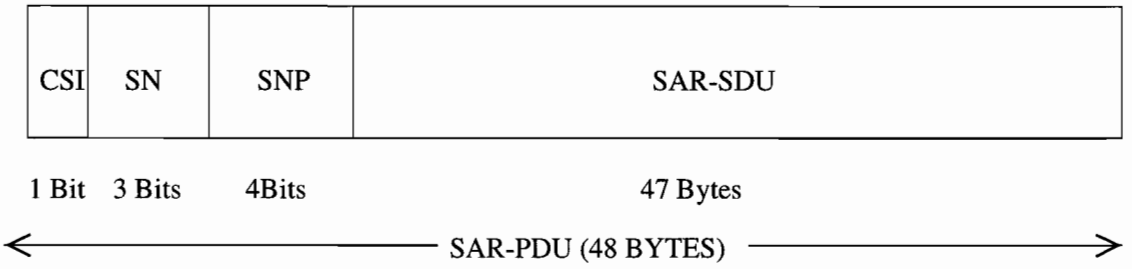


Figure 2.3: AAL1-PDU Structure

2.1.2.2 ATM Adaptation Layer 5

For data, an overhead of 1 byte per 48 byte payload is a wastage of about 2 percent of the bandwidth. For data applications, which typically have a particularly large payload, an

adaptation layer AAL5 was proposed. AAL5 has very small protocol overhead, hence its initial name: Simple and Efficient Adaptation Layer (SEAL). This layer is further divided into Segmentation and Reassembly (SAR) sublayer and Common Part Convergence Sublayer (CPCS). The Payload Type Indicator (PTI) field in the ATM header is used by SAR sublayer and hence it doesn't add any additional overhead. The structure of an AAL5 CPCS Payload Data Unit (PDU) is shown in Figure 2.4. It consists of 3 fields: AAL5 CPCS Service Data Unit (SDU) which is the data to be transported (1-65535 bytes), PAD which is the padding done to guarantee the CPCS PDU to be a multiple of 48 bytes and a CPCS PDU trailer having control information. The CPCS PDU trailer consists of 1 byte of UUI (User to User Indication), 1 byte of Common Part Indicator (CPI), 2 bytes of Length Indicator (LI) and 4 bytes of Cyclic Redundancy Check (CRC) used for error protection. AAL5 is designed for large data bursts, which incur very small percentage overhead. But if the data bursts are smaller, the percentage overhead gets larger. Typically voice packets from low bit rate coders are around 20 bytes long and hence AAL5 is also not particularly suited for voice [7].

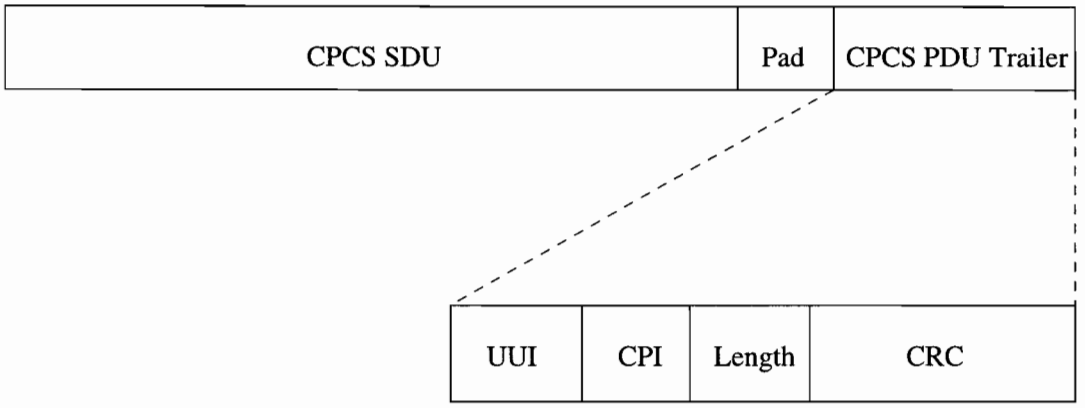


Figure 2.4: AAL5-PDU Structure

2.1.2.3 ATM Adaptation Layer 2

The AAL type 2 [1] provides for the bandwidth efficient transmission of low-rate, short, and variable length packets in delay sensitive applications. More than one AAL type 2 user can be supported on a single ATM connection. To understand the problem of not more than one user using a single ATM connection, consider the example of a low-bit rate coder G.728 which has a 16 kb/s coding rate. The coder takes 23.5 ms (in addition to its algorithmic delay of 2.5 ms) to generate a payload of 47 bytes allowed using AAL1. This is a huge delay considering the fact that we aim at an end-to-end delay of less than 200 ms. To reduce the cell formation delay, we need to pad the cell with null bytes. This is a waste of bandwidth. We see from this example that there is a need for the cell to be shared between users. AAL2 was designed to solve this problem by supporting more than one user data on single ATM connection. This would solve the problem of cell formation delay while retaining high bandwidth efficiency. The AAL2 layer takes care of the complexity due to multiplexing within an ATM Virtual Channel (VC). Like other

AALs, AAL2 is divided into SAR, CPS (Common Part Sublayer) and SSCS (Service Specific Convergence Sublayer),as shown in Figure 2.5. Different SSCS protocols are defined to support different AAL2 user services. SSCS may also be NULL, merely providing for the mapping. The CPCS takes of multiplexing in AAL2. The following four types of SSCS PDUs are recognized by ATM Forum:

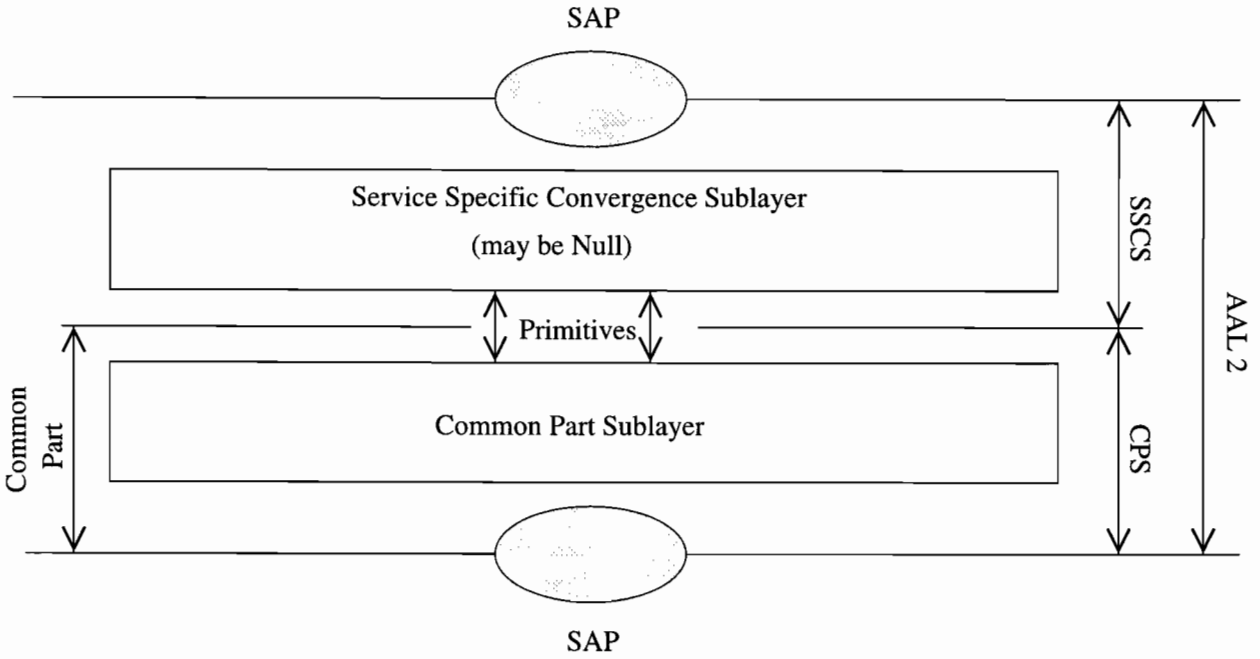


Figure 2.5: Structure of AAL2

- *Unprotected*: The payload is not protected. This format type is used by default for all SSCS packets unless an alternate type is explicitly specified. This type carries User to User Information (UUI) and Length Indicator (LI) in its header.

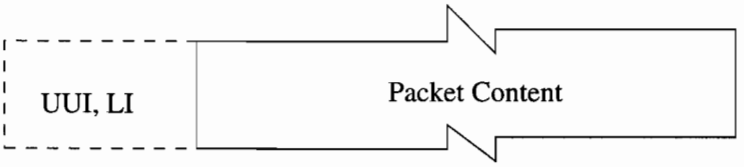


Figure 2.6: AAL2 SSCS Packet Format : Unprotected

- *Partially Protected*: The payload for this type of SSCS packet begins with UUI, LI and 19 bits of additional header information. A 5 bit Cyclic Redundancy Code (CRC) is used to protect the 19 bits of additional information. The rest of the payload is unprotected.
- *Fully Protected*: The entire payload in the SSCS packet is protected using a 10 bit CRC. A 6 bit Message Type field is also present in the trailer along with the CRC. UUI and LI fields are present in the header.

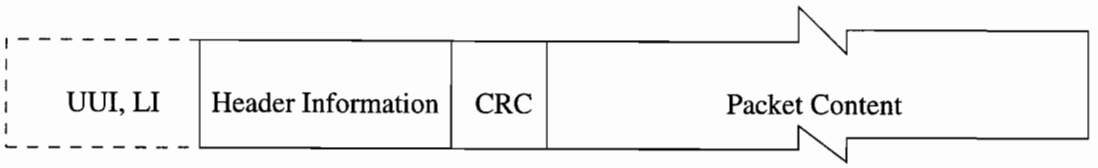


Figure 2.7: AAL2 SSCS Packet Format : Partially Protected

- *Error Detection Trailer*: This is attached as the last segment of the framed data unit when error detection option in the SSCS is selected.

The CPS packet format is shown in Figure 2.9. The CPS header occupies 3 bytes and contains the following fields within it:

- *Channel Identifier (CID)*: This field is used to distinguish between AAL2 users within a single VC. CID values from "8" to "255" are used to identify the users of AAL2 CPS. The value "0" is used only when an all zero octet is used for the padding function. The value "1" is reserved for layer management procedures. The values "2" to "7" are reserved for future use. The UUI field helps distinguish between SSCS and Layer management.

- *Length Indicator (LI)*: The LI field is binary encoded with a value one less than the number of octets in the CPS-Packet Payload. The default maximum length of CPS-Packet Payload is 44 bytes. The maximum length can otherwise be set at 64 bytes. The maximum length an be set for each channel (CID). Within a channel, the maximum value is constant. The maximum value is set by signaling or management procedures.

- *User-to-User Indication (UUI)*: The UUI field can be used to distinguish between the SSCS entities and Layer Management users of the CPS or to convey specific information between CPS users. It occupies 5 bits and provides for 32 code-points. "0" to "27" are available for SSCS entities, code-points "30" and "31" are available for Layer Management and code-points "28" and "29" are reserved for future use.

- *Header Error Control (HEC)*: The AAL2 transmitter calculates the remainder of the division (modulo 2), by the generator polynomial x^5+x^2+1, of the product of x^5 and the contents of the first 19 bits of the CPS-Packet header (CPS-PH). The coefficients of the remainder polynomial are inserted in the HEC field with the coefficient of the x^4 term in the most significant bit of the HEC field. The AAL2 receiver uses the contents of the HEC field to detect errors in the CPS-PH.

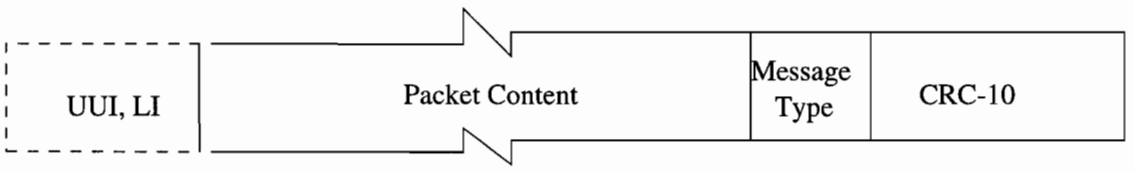
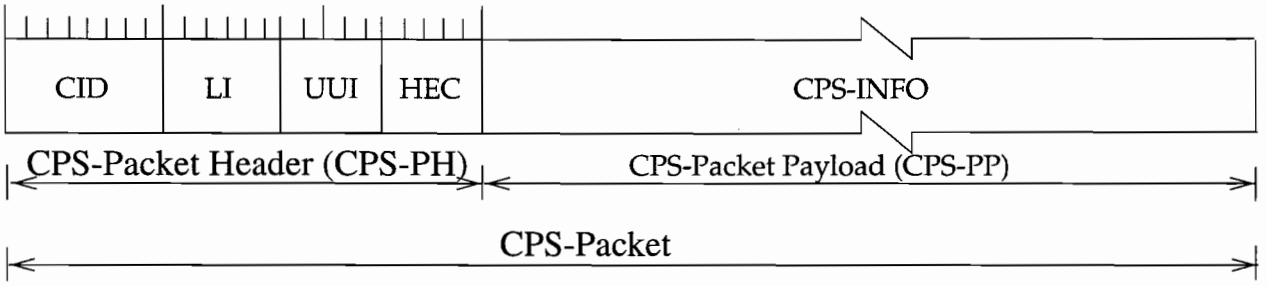


Figure 2.8: AAL2 SSCS Packet Format : Fully Protected



CID:	Channel Identifier	(8 bits)
LI:	Length Indicator	(6 bits)
UUI:	User-to-User Indication	(5 bits)
HEC:	Header Error Control	(5 bits)
CPS-INFO:	Information	(1..... 45/64 oct)

Figure 2.9: AAL2 CPS Packet Format

The AAL2 payload contains a Start Field (STF) and may contain padding in addition to the CPS-packets.

- *STF*: The STF consists of the following subfields:
  - *Offset Field (OSF)*: This field carries the binary value of the offset in octets between the end of the STF and the first start of a CPS-Packet. In the absence of a first start of a CPS packet the STF is calculated to the start of the PAD field. Values greater than 47 for the STF do not make sense and are not valid.
  - *Sequence Number (SN)*: This bit is used to number (modulo 2) the stream of CPS-PDUs.
  - *Parity*: The parity bit is used to detect error in the STF. This is set such that the parity over the 8-bit STF is odd.
- *PAD*: Padding is done to fill the partially filled CPS-PDUs. "0" to "47" bytes of padding can be done.

The multiplexing process of AAL2 CPS layer is shown in Figure 2.10. When insufficient CPS-Packets are available to fill a cell, the transmitter waits for time "Timer-CU" and then "pads" the cell. This is done to put a limit on the maximum time (the value of Timer-CU) an ATM cell waits to be sent out after the arrival of the first packet into it.

2.2 AAL2 Research at KU

Our group (AAL2 group) at The University of Kansas (KU) has been working on a project sponsored by Sprint Corporation. The aim of this project is to study various aspects of AAL2 at both design and implementation levels.

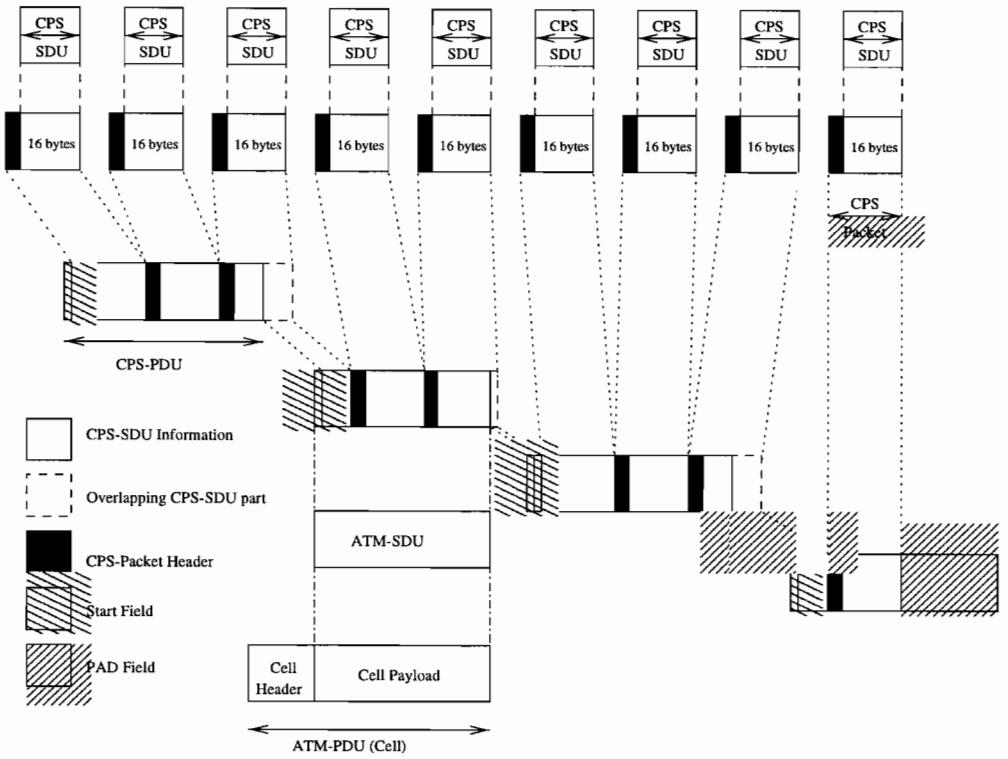


Figure 2.10: Multiplexing and Packing CPS-Packets into CPS-PDUs (ATM Cells)

Some of the initial work focussed on characterizing AAL2 performance [14], comparing the efficiency of AAL2 for voice transport [7], and implementation of AAL2 in a narrowband ATM environment [16]. This was followed by work on adding QoS support for multiple classes of traffic in AAL2 [8]. A feedback mechanism for AAL2 that dynamically controls the source coding rate was proposed [18]. Work was done on AAL2 traffic characterization [5] and estimation of voice activity statistics [6]. Implementation of AAL2 for broadband ATM was done [17]. This was followed by the design of a CAC mechanism on which this report is based. An implementation of a simplified version of the CAC scheme was also done to prove the effectiveness and implementability of the CAC scheme [15]. Work is being carried out to enhance the above implementation.

This report focuses on the CAC scheme developed for AAL2 users.

Chapter 3

Options in CAC Design for AAL2

This Chapter starts with a discussion of Congestion Control mechanisms in ATM networks. The role of CAC as a means of congestion avoidance is discussed. This is followed by a discussion of the key components of a CAC scheme are and the specific requirements for an AAL2 CAC. We pose some questions that arise when deciding on a CAC mechanism for AAL2 and then proceed to answer them.

3.1 Congestion Control in ATM networks

In packet networks there is a queue (or queues) associated with each link at every switching node in the network. As the arrival rate at this link approaches its transmission rate, the queue length dramatically increases. The buffer size puts a limit on the queue length. Packets arriving when the buffer is full are discarded. The resulting packet loss triggers retransmissions in some applications. This adds to the existing traffic. As the retransmissions increase, more nodes get congested which results in more packets getting dropped. The network might reach a state where most of the traffic in it is due to retransmissions and all the applications are jammed. Effective congestion control avoids such a state in the network. The performance of any network depends on congestion control mechanisms that would effectively control congestion at the same time maintaining high bandwidth efficiency.

Congestion control is a complex task in integrated high-speed networks. Some of the challenges faced especially in high-speed ATM networks are:

- Different sources generate traffic at significantly different rates. While a single voice application can generate traffic of only tens of kilobits per second, real-time high quality video can generate a few megabits of traffic per second.
- Different applications have different QoS requirements of absolute delay, delay variation and loss. Each application can have some or all of the above QoS requirements.
- Traffic characteristics of many applications are not well understood and accurate models for the traffic patterns do not exist today.

- Large propagation delays compared to transmission times result in long periods between the occurrence of congestion and its detection by remote network control elements.
- A reasonably high network utilization should be aimed for to reduce the cost of the network.

Figure 3.1 shows various mechanisms employed during different time frames to solve the problem of congestion. Some of the options shown could be used in conjunction with other options for effective control. An earlier work dealt with dynamic source coding [18] when using AAL2 for voice.

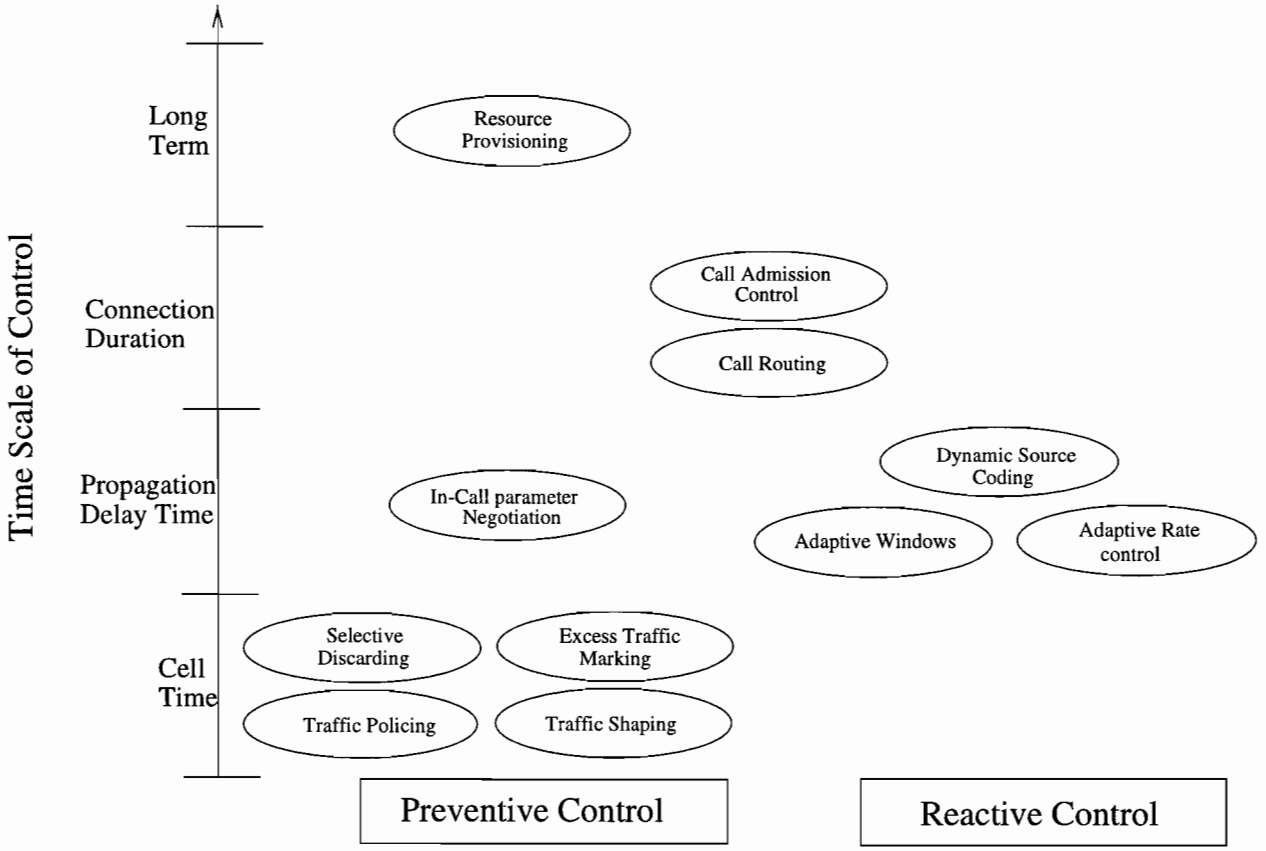


Figure 3.1: ATM congestion control options [26]

3.2 Call Admission Control

Call Admission Control (CAC) has been identified as one of the most important means of preventing congestion and providing satisfactory QoS. CAC is a means by which the admission of new calls (users of an application) into the network is controlled according to the state of the network. When a new connection request is received at the network, the call admission procedure is executed to decide whether to accept or reject the call.

The incoming call is accepted if the network has enough resources to provide the QoS requirements of the connection request without affecting the QoS provided to the existing connections. Based on the design of the system, the CAC function has to take care of calculating the bandwidth required for the incoming call based on the nature of the call. It assigns the call to a connection called Virtual Circuit (VC) if it can support the new user.

3.3 ATM Level CAC vs. AAL Level CAC

Depending on the network design, the CAC can be administered at both ATM and AAL levels. ATM level CAC takes care of allocation of VCs within a given Virtual Path (VP). AAL level CAC takes care of admitting new users (or applications) into a given VC. That is, a VC in a AAL CAC plays the same role as a VP in ATM CAC. AAL CAC is required if the AAL allows more than one user to be multiplexed onto the same VC. AAL2 is designed to carry information from more then one user in a single VC and hence requires a well defined CAC procedure at the AAL layer.

3.4 Design Issues for AAL2 CAC

Using AAL2, multiple user data can be multiplexed into the same VC. That is, each cell can contain packets coming from different users. The central idea behind AAL2 is efficient use of the ATM cell while keeping the cell formation delay small. Any CAC mechanism for AAL2 should exploit the built-in multiplexing properties. Most of the traffic carried using AAL2 is voice traffic that has very stringent QoS requirements. Also, the AAL2 CAC function has to take care of factors like high load variation inherent to telephone users, the existence of heterogeneous users (users transmitting at different coding rates) and user multiplexing within a VC, to note a few.

A CAC can be classified in a number of different ways based on its operational design. We now go through a series of steps discussing the various options available for the CAC and the choices that make more sense in the AAL2 context. A detailed CAC algorithm is postponed to Chapter 5.

3.4.1 Static Bandwidth Allocation and Dynamic Renegotiation

In a *static CAC*, the resource allocated to a particular user does not change for the duration of the call. A user presents a fixed set of traffic parameters to the CAC function and asks for a well defined QoS requirement. He promises not to change these requirements within a single connection duration. The CAC procedure in turn admits the user if it can accomodate the requirements of the user at the time of his connection request. This is a relatively simple procedure.

In a *dynamic CAC*, the resources allocated to various users can be changed during the course of a call. It is assumed that the user has a possibility of varying his transmission rate and/or his QoS requirements according to his admission contract. At admission request the user can ask for a bandwidth at which he prefers to send his traffic. The network

can in turn offer the user the maximum bandwidth closest to the requested bandwidth that it can support at that point of time. The user can accept or reject that offer. If the user accepts the offered bandwidth, he starts his transmission and limits it to the contract. At any point of time during the connection he can request for an increase or decrease in the bandwidth allocated to him. The network can also inform users of the availability of more bandwidth. If the user feels that he can't transmit at the available load, he can choose to stay idle until the network has spare capacity for the requested bandwidth. The actual procedure used for negotiation varies with the implementation and might contain some or all of the above features. This kind of resource allocation mechanism is highly efficient at the cost of higher complexity in implementation. Reliable network monitoring, effective policing and a good pricing policy are required for the implementation of this scheme. The most important aspect is the possibility for the user to change his transmission rate.

We assume that all or most of the traffic carried using AAL2 will be real time voice (telephone) traffic. The instantaneous voice transmission rate can be varied by using a different voice coder that has a different coding rate. Previous work [18] demonstrated how *dynamic source code control* can be used for limiting queuing delays in the AAL2 transmitter. For this scheme to be able to succeed, the coder should be co-located with the AAL2 transmitter and also controllable from the AAL layer. Also the change in the coder has to be communicated to the receiving end. This involves additional signaling. Though doing this is definitely possible, this aspect is not dealt with directly in this report. Instead, we try to provide suggestions on the ways to integrate the AAL2 CAC system provided in this report with the dynamic source code control in [18].

In this report, we assume that once a VC is established, its capacity is fixed. The assumption is made in view of the implementation complexity associated with a dynamically changing VC capacity. It is discussed in *section 3.3.3* how a single static VC might be inefficient for telephone traffic with a lot of load variation. Thus we employ a mixed or a *hybrid* scheme that is both simple to implement and also efficient.

3.4.2 Maintenance of VC state

The AAL2 CAC function has to keep an update on the bandwidth occupancy of the VC in use. This is required to decide on free bandwidth to support new users. This can be done in any of the following ways;

- By maintaining all allowed combinations of the number of different types of users. If there a 'n' different types of users that would require different bandwidths, the CAC mechanism should maintain an n-dimensional state space of allowable sets. This kind of implementation is easy for lesser 'n'. The state space is difficult to maintain for larger values of 'n'. The advantage of this method is lesser computation during the actual CAC procedure.

- By going through the mathematical procedure of estimating the additional bandwidth required for the new user while performing the actual CAC function. For doing this, the AAL2 CAC function must also keep track of the present VC usage. This can be kept track of by

- keeping an update of the traffic condition of the VC through on-line measurements. The CAC function decides the connection admission considering the traffic parameters of the existing VC (as calculated from the on-line measurements). This is a more accurate and thereby efficient method of doing the CAC procedure. But factors like sudden increase or decrease of traffic should be taken care of by filtering the samples.
- keeping a record of the number of already active connections. The new user is admitted if the mathematical estimate of the total bandwidth required for the existing users and the new user is less than the available bandwidth. This is the most feasible way. Though the process of calculating the bandwidth required is complex, it could be implemented in hardware for faster calculation. This method is based on expected traffic patterns and will not perform well if the actual inflow of traffic at a particular time is significantly larger or smaller than the expected value. This could happen if there is under usage or over usage by a section of users. This is most unlikely for voice traffic since the variation in traffic is typically small for telephone users. Assuming that the coding rate is fixed, the transmission rate does not vary by more than twice the expected value (corresponding to variation in speech activity [6]). We choose this scheme as the procedure for bandwidth calculation while demonstrating the effectiveness of our CAC scheme. But all the above schemes can be used in conjunction with our CAC scheme.

3.4.3 Multiple Smaller VCs vs Single big VC

Consider Figure 3.2 which shows the estimation of bandwidth required per AAL2 user for the case of homogeneous users (all users with identical voice coding rates) using a 32 kb/s coder. The estimation of required bandwidth is done using the NEC CAC scheme [10]. The graph given is for showing the trend in bandwidth requirements. Details of the method of calculation and other parameters used are discussed in Chapter 4. From the graph we see that the bandwidth required per user drastically reduces as the number of users increase for small number of users. For large number (n) of users, the value of bandwidth required per user can be calculated by multiplying the coding rate with speech activity factor and the CPS-packet and cell overheads. The formula for the bandwidth required per user is given below:

$$(\mathtt{coding\_rate})(\mathtt{speech\_activity\_factor})\frac{(\mathtt{voice\_packet\_size}+3)}{(\mathtt{voice\_packet\_size})}\frac{53}{47} \tag{3.1}$$

The above formula gives a value of 17.43 kb/s for the parameters used to generate Figure 3.2. We observe than the bandwidth required is approximated to be just above 20 kb/s because of the delay QoS constraints. These are discussed in detail in Chapter 4.

We know that the aggregate telephone traffic varies a lot during the course of a day, with more connections during the day and fewer at night. The network must be designed such that it caters to the peak demand. If the resources (bandwidth) that would support

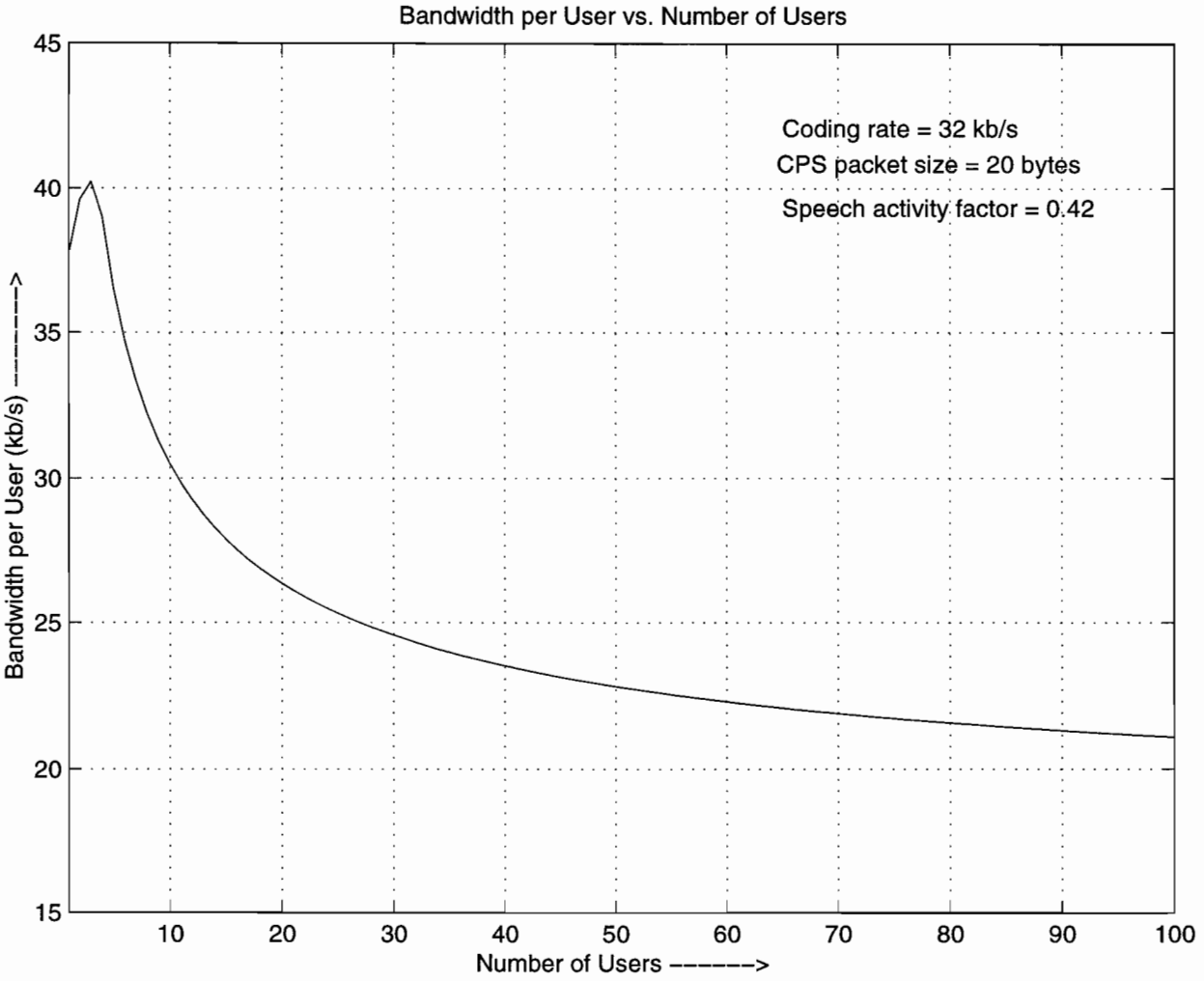


Figure 3.2: Bandwidth Required per User

peak load are allocated all through the day, they would go underutilized. This is a serious issue since at the time the bandwidth is going waste, it might actually be required for a different application.

If there is a lot of load variation, the best option is to have a VC of variable capacity such that chunks of bandwidth can be added to it and removed from it. That is, *a node should be able to negotiate an increase of bandwidth on an existing VC*. This is difficult to implement and also involves a great deal of signaling. We therefore rejected this option. Instead, we choose to have multiple VCs of smaller capacity that can be created and torn down according to the demand as opposed to a single large VC that supports peak load. There are associated trade offs for this. Now consider two extreme cases, a case where there are few users and very small load variation and a case where there is a lot of load variation and many users.

3.4.3.1 Less users and Low Load variation

As seen from Figure 3.2, the bandwidth per user is more at small number of users. For a simple illustration, a single VC carrying 15 users needs a bandwidth of 418 kb/s compared to 604 kb/s (44.5 % more) required by 5 VCs carrying 3 users each. With low load variation the gain associated with activating and releasing smaller bandwidth VC(s) will be overtaken by the loss due to lack of multiplexing gain. For such cases a single big VC of the size close to the expected peak load is recommended. The CAC procedure is quite simple in this case.

3.4.3.2 Larger Number of Users and Large Variation in Load

Consider the case of having a single large VC of 100 users having a required bandwidth of 2108 kb/s vs. 20 smaller VCs requiring a bandwidth of 2635 kb/s (25 % more). If we have a lot of load variation, we might make considerable bandwidth gains by releasing the smaller VCs when they are not needed. Thus for such a case the use of multiple VCs that are setup and torn down based on need *might* be a good choice. SVCs (Switched Virtual Circuits) in ATM networks provide temporary connection between two points in the network. They can be used in place of PVCs (Permanent Virtual Circuits) when required for relatively smaller time.

The typical setup when using a single PVC and when using multiple SVCs is shown in Figure 3.3

It is well known that Telephone traffic varies a lot over the time of the day. We therefore are interested in designing a CAC keeping a typical high end group with load variation in mind.

3.4.4 Choice of Static CBR, Static VBR or a Dynamic VBR VC

Another basic question surrounding the CAC for AAL2 is whether to use a CBR VC or a VBR VC. The answer to this question has been discussed in a previous work at KU [18]. A summary of the discussion is provided here. If a CBR VC is setup between two nodes, the sender node has to limit its traffic to a Peak Cell Rate (PCR) allowing a maximum

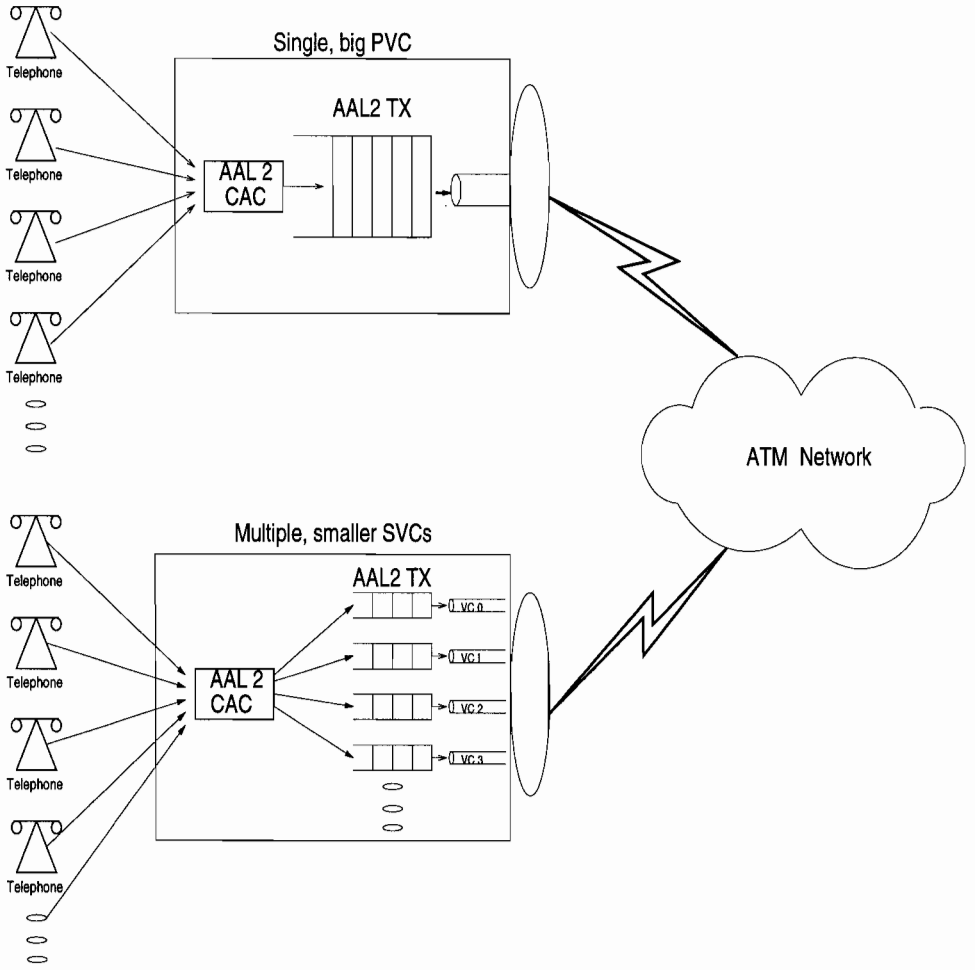


Figure 3.3: Single, Big PVC vs Multiple, Smaller SVCs

variation equal to the Cell Delay Variation Tolerance (CDVT) of the VC. If the VC is a VBR VC, the sender transmits at an average rate less than or equal to Sustained Cell Rate (SCR) allowing a Maximum Burst Size (MBS) at a rate equal to the PCR of the VC. As discussed in the previous section, a VC is called *static* if the bandwidth of a VC is fixed and is called *dynamic* if the sender node can ask the network for an increase of bandwidth of an existing VC. The sender node has to limit its traffic to the contract and is subject to policing at the network access point. The access point can discard some of the cells if the flow is found to be exceeding its traffic parameters. This process is called Usage Parameter Control (UPC). The associated traffic parameters are called UPC parameters. The estimation of these parameters is very important for an effective CAC. Chapter 4 deals with estimation of these parameters for AAL2 users.

The usage of CBR VC was found to be simpler in the previous work [18]. As stated in this work the advantages in using a CBR VC are:

- No need for long buffers at the network nodes.
- Traditional CAC schemes can be mapped for AAL2 with some modifications.
- Smaller connection setup time.

The disadvantages in using a CBR VC are

- Link utilization is not optimum.
- Costly compared to a VBR VC having the same PCR.

The reader is recommended to go through [18] for a detailed discussion on this problem.

An effort was made to define the CAC procedure keeping in view the most conservative case: having a CBR VC. However the CAC procedure to be described in Chapter 5 will be valid for the case of static or dynamic VBR VCs with some minor changes which will be discussed.

The performance measures that need to be evaluated while selecting a CAC procedure are call rejection probability and high network efficiency. The call rejection probability should be reduced while maintaining a good network efficiency. Since these two are in conflict, an effort should be made for a balance between the call rejection probability and network efficiency. The CAC should also allow for users using different coders. Having said that, we shift our focus onto the methods for estimation of bandwidth required for AAL2 users, which is an important part of the CAC procedure. After getting to know the tools for bandwidth estimation for AAL2 users, we proceed to discuss the actual CAC algorithm.

Chapter 4

Bandwidth requirements for AAL2 users

In this chapter we deal with the methods for estimating bandwidth requirements for AAL2 users for both CBR and VBR types of virtual circuits. We propose a simulation based approach called the *Virtual Buffer Measurement Mechanism* for estimating VBR traffic parameters for AAL2 users. We then discuss the mapping of *NEC Multi-class CAC* method for calculation of CBR bandwidth requirements for AAL2.

4.1 Importance of Knowing the Traffic Parameters

Traffic policing, or Usage Parameter Control (UPC) as it is known in ATM circles, is a critical component of the overall traffic management and congestion control strategy for ATM networks [2]. Associated with each Virtual Channel Connection (VCC) in an ATM network are certain traffic parameters upon which the network provider, through Connection Admission Control (CAC) procedures, can base resource allocation decisions aimed at maintaining the desired Quality of Service (QoS) for the VCC. In order to protect the QoS of each VCC from traffic misbehavior by other VCCs, each VCC's traffic stream is policed to ensure that the traffic is adhering to its stated traffic parameters. Without such policing, a VCC could consume significantly more than its allocation of network resources, thereby jeopardizing the QoS of connections that are adhering to their traffic specifications.

For Constant Bit Rate (CBR) connections, the UPC (traffic) parameters are Peak Cell Rate (PCR) and Cell Delay Variation Tolerance (CDVT). For Variable Bit Rate (VBR) connections, Sustainable Cell Rate (SCR) and Burst Tolerance (BT) are specified in addition to PCR and CDVT.

In order for the CAC and UPC mechanisms to be effective, it is fundamentally important that the specified UPC parameters accurately reflect the actual traffic stream. Parameter sets that are "too large" will tend to result in over-allocation of resources by the network CAC function, resulting in unnecessarily high costs to the network user (assuming that price is based at least in part on resource allocation [19]). Parameter sets that are "too small" will tend to cause policing violations, resulting in traffic either being discarded immediately at the policer or at least being "marked" with a low discarding priority and hence targeted as expendable if the network becomes congested [2].

4.2 Estimation of VBR UPC parameters

A fundamental question, then, is: *How can we determine or at least estimate the UPC parameters for a VCC?* The answer to this question is complicated by a number of factors. One is the uncertainty associated with the behavior of specific traffic sources. Another is the difficulty of constructing mathematical models of traffic behavior. A third is the well-known fact (see for example [20] and [21]) that there are an infinite number of UPC parameter sets that can be used to describe a given traffic stream.

Let us assume that a statistically representative sample of a particular VBR traffic stream is available. Such a sample could be obtained from observation of the traffic stream at an earlier time or from observation of a different traffic stream that is known or believed to be similar in its behavior, such as a stream originating from the same type of application in the same network environment. The existence of such a representative sample nullifies the uncertainty factor, alleviates the need for construction of a traffic model, and allows for the use of measurement and/or simulation techniques. We emphasize that, although we focus here on simulation, the technique is equally valid in a live traffic measurement context. However, even with such a representative traffic sample, determining UPC parameters via direct simulation would be an arduous task, as discussed in the next section. In that section, we also describe and analyze an efficient measurement technique, *virtual buffer measurement*, that can be used to estimate all "minimal sets" (defined later) of UPC parameters for a particular VBR traffic stream. Selection of a particular set of parameters can then be made based on factors such as QoS requirements and bandwidth usage, as discussed in [21] and later in this chapter.

In *section 4.4* we apply the virtual buffer measurement technique to an ATM Adaptation Layer, Type 2 (AAL2) traffic stream using simulation. We also validate the model by direct simulation with a UPC policer model.

4.3 Obtaining UPC Parameters via Virtual Buffer Measurement

4.3.1 Overview

Usage Parameter Control (UPC) is accomplished by using one or more traffic policers, each implemented as a Generic Cell Rate Algorithm (GCRA) [2]. For VBR traffic, a dual policer configuration is used (see Figure 4.1), in which the first policer monitors PCR and CDVT and the second monitors SCR and BT. In the Figure, $T_0 = 1/PCR$, $T_s = 1/SCR$, and the restriction of CDVT = T_0 is in accordance with standards recommendations [2].

Our goal is to find all "minimal sets" of UPC parameters PCR, CDVT (=1/PCR), SCR, and BT for a particular VBR traffic stream, where a minimal set is defined as follows.

- 1) The traffic stream will be completely conformant (all cells conforming) when policed by a GCRA(T_0, T_0) policer (see Figure 4.1) followed by a GCRA(T_s, $\tau + T_0$) policer, where PCR = $1/T_0$, CDVT = T_0, SCR = $1/T_s$, and BT = τ.

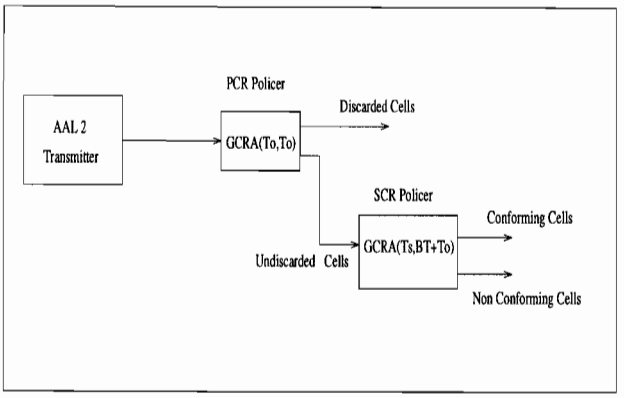


Figure 4.1: Dual Policer Configuration

- 2) Any significant reduction in any one of the values PCR, SCR, or BT will result in some nonconformance when using the above dual policer configuration on the given traffic stream.

With the restriction of CDVT=1/PCR, there is a unique minimal value of PCR for any given traffic stream. However, for *every* value of SCR between the mean rate and the peak rate of the source, there will be an associated minimal value of BT.

Even with the availability of a representative traffic stream, attempting to find even one minimal set of UPC parameters from direct simulation of this dual policer configuration would be most inefficient. This is because direct simulation would require a set of search procedures, each varying one or more parameter values until the boundary between conformance and non-conformance had been estimated to sufficient accuracy. Such a procedure is inefficient not only because the number of simulations required is relatively large, but also because the results of one simulation must be analyzed in order to set the parameter values for the next simulation. A number of such "coupled" simulations would be required to find an estimate for PCR, and then even for a given value of T_s = 1/SCR, many more such "coupled" simulations would be required to estimate the associated minimal value of BT. These difficulties are intensified if live traffic observation and measurements are attempted.

In place of this tedious direct GCRA simulation approach, we advocate here a *virtual buffer measurement* approach that is much more efficient and yet (as we will show) produces essentially equivalent UPC parameter values. The virtual buffer (VB) concept was introduced in [20], in which its equivalence with GCRA policing was assumed, but not rigorously demonstrated. The VB concept is illustrated in Figure 4.2.

The traffic flowing out of a source is fed into a virtual buffer (FIFO). The virtual buffer (VB) is served such that cells leave it at a fixed rate. The service rate (SR) of the VB is varied over a range of values; note that a number of virtual buffers with different service rates can be implemented or simulated in parallel. For each value of service rate, the maximum buffer fill (MBF) at any instant is measured (MBF excludes the cell in service). We will show that the resulting set of (SR,MBF) pairs can be used to accurately estimate both the minimal value of PCR and a set of minimal (SCR,BT) pairs, one for each service rate simulated. Thus we need perform only one simulation for each minimal (SCR,BT)

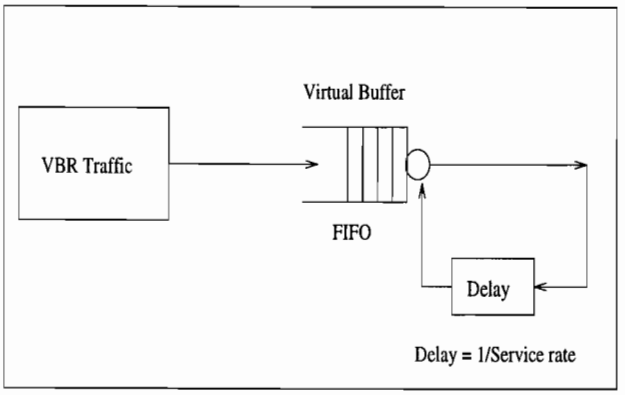


Figure 4.2: Virtual Buffer Model

pair obtained, as opposed to several coupled simulations with the direct GCRA simulation approach. Further, there is no coupling at all between any of the simulations in the virtual buffer simulation approach.

A summary of the translations between (SR,MBF) pairs and PCR, SCR, and BT values is as follows.

1) PCR and CDVT: The minimum service rate of the virtual buffer that gives an MBF of 1 is the minimal PCR for the given traffic stream, subject to CDVT=1/PCR.

2) SCR and BT: The service rate SR corresponds to the SCR = $1/T_s$, and then BT can be found from MBF, SCR and PCR = $1/T_0$ by using the relation:

$$BT = MBF * T_s - T_0 \tag{4.1}$$

Once BT is found, it can be converted to maximum burst size (MBS) according to the well-known relationship [2]:

$$MBS = \lfloor BT/(T_s - T_0) \rfloor + 1 \tag{4.2}$$

where $\lfloor\ \rfloor$ represents the integer part.

We proceed to analytically derive these relationships.

4.3.1.1 A General GCRA-VB Relationship

We first establish a general relationship between a Virtual Buffer (VB) and a GCRA policer, both acting on the same ATM traffic stream, as shown in Figure 4.3. Refer also to Figure 4.4 for details of the GCRA algorithm.

Proposition 1: A traffic stream that results in an MBF of m *when processed by a Virtual Buffer served at a rate of* $1/I$ *will be GCRA*(I, L) *conforming if* $L = mI$ *and will be nonconforming if* $L < (m-1)I$.

We begin the proof by noting that the MBF will result from one or more specific busy periods of the VB. We analyze one such busy period, letting cell n of that busy period be the one that results in the given MBF, which we designate m. The situation is shown in Figure 4.5, where the T_i values are interarrival times of cells in the traffic stream. Without loss of generality, we let t=0 correspond to the arrival time of the first cell in this busy

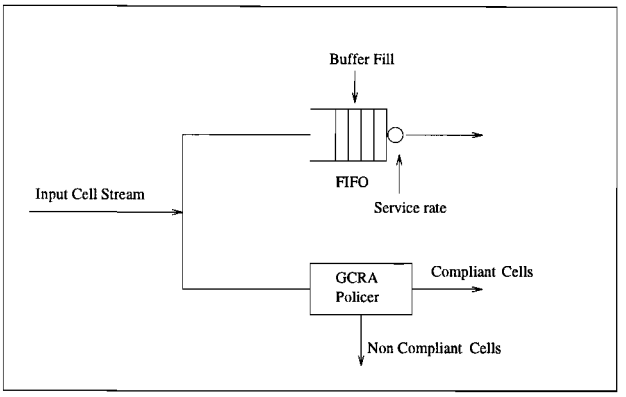


Figure 4.3: Comparison of the Two Systems

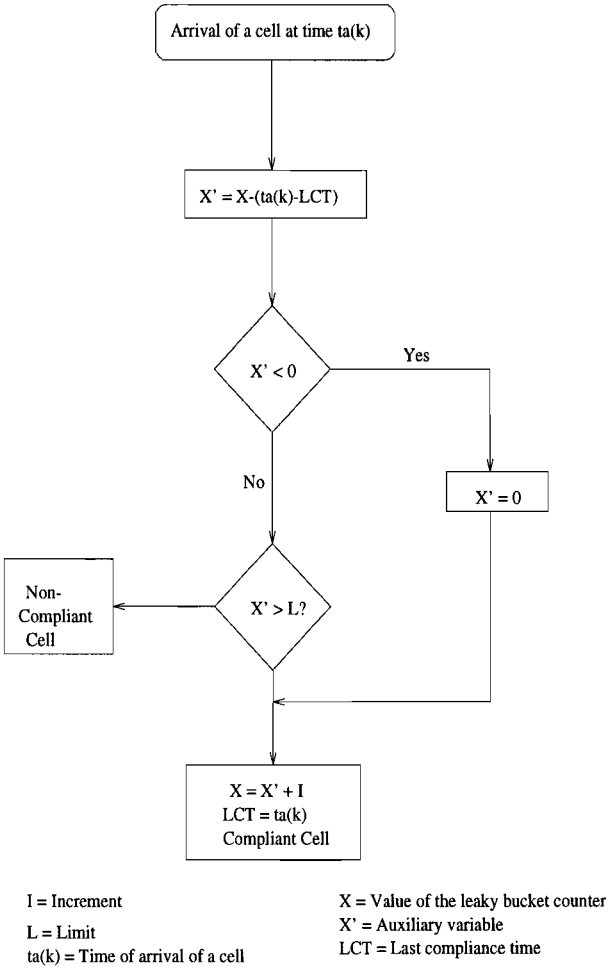


Figure 4.4: The GCRA Leaky Bucket Policer

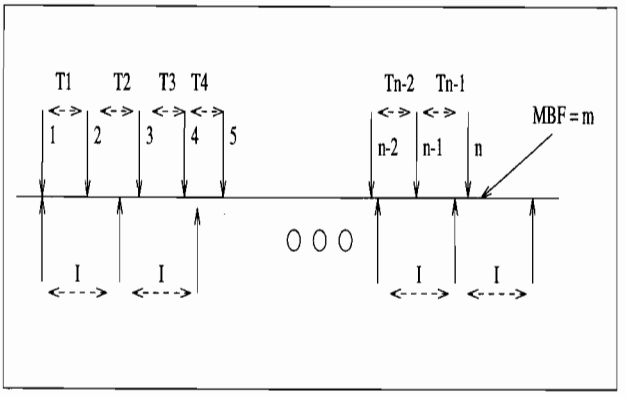


Figure 4.5: Proposition 1

period.

We first note that for cell n to result in an MBF of m, there must have been exactly $n-m-1$ service completions between time t=0 and the arrival of cell n at time $T_1 + T_2 + T_3 + ... + T_{(n-1)}$. This yields:

$$(n-m-1)I \leq T_1 + T_2 + ... + T_{(n-1)} \leq (n-m)I \tag{4.3}$$

Furthermore, since the buffer fill does not exceed m, the following must hold for every arrival $k < n$:

$$(k-m-1)I \leq T_1 + T_2 + ... + T_{(k-1)} \tag{4.4}$$

We now follow the GCRA(I, L) algorithm for each arrival and let X'_k be the GCRA variable X' after the arrival of cell k. Assuming that all cells through $k-1$ are conforming and that X' never reaches zero (which cannot happen during a virtual buffer busy period), it is easy to show that $X'_k = (k-1)I - (T_1 + T_2 + ... + T_{k-1})$ and so cell k will be conforming if $X'_k = (k-1)I - (T_1 + T_2 + ... + T_{k-1}) \leq L$ or $(k-1)I - L \leq T_1 + T_2 + ... + T_{k-1}$. Comparing this last with inequality (4.4) above, we see by induction that conformance of every cell in this (and every) busy period will be assured if $L = mI$.

Now consider cell n, which causes the maximum buffer fill of m. The arrival of this cell will result in $X'_n = (n-1)I - (T_1 + T_2 + ... + T_{n-1})$. Multiplying inequality (4.3) above by -1 and then adding $(n-1)I$, we conclude that $(m-1)I \leq X'_n \leq mI$. Since GCRA conformance requires $X' \leq L$, we conclude that a value of $L < (m-1)I$ would result in non-conformance for cell n and hence for the entire traffic stream. This completes the proof of Proposition 1.

4.3.2 Application to UPC Parameter Estimation

We first apply Proposition 1 to the problem of determining the minimum PCR for a given traffic stream from virtual buffer observations. As noted previously, PCR will be policed with a GCRA(T_0, T_0) policer, where $T_0 = 1/PCR$. Using Proposition 1 with $I = L = T_0$, we see that the only possible MBF values for GCRA(T_0, T_0) conformance are 0 and 1. Since we seek the minimum value of PCR, we conclude immediately that this will be the minimum

VB service rate that will result in an MBF of 1 in the virtual buffer. This can be determined (at least approximately) from the MBF vs. service rate data derived from VB simulation.

We next consider the problem of determining (SCR,BT) pairs from VB observations. We note that the SCR policer in Figure 4.1 has GCRA parameters $I = T_s$ and $L = BT + T_0$, where $T_s = 1/SCR$. Direct application of Proposition 1 yields the following result. Every given VB service rate is a valid SCR (within the reasonable bounds of mean_rate_of_traffic_stream $<$ SCR $<$ PCR). For a given service rate (SCR=$1/T_s$) with given MBF, Proposition 1 states that the following value of BT will result in the traffic stream being declared conforming: $BT = MBF*T_s - T_0$. (Note that we have already been able to determine T_0 from VB measurements.) Proposition 1 also states that a value of BT less than $(MBF-1)*T_s - T_0$ will result in non-conformance. Thus, $BT = MBF*T_s - T_0$ may not be the absolute minimum value of BT for the given SCR, but it is within T_s of being the minimum value, which is the best that can be done from VB measurements.

Having obtained a value for BT, the corresponding value of MBS can be obtained from equation 4.2.

4.4 Application to AAL2 Traffic

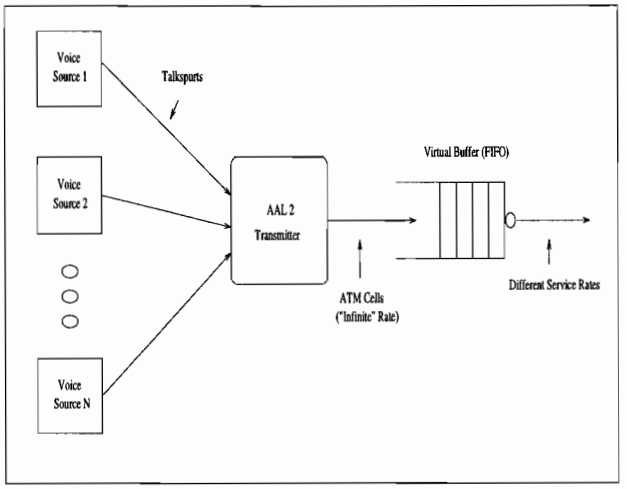


Figure 4.6: Simulation Model

In this section and the next, we use the procedure and results derived above to obtain traffic descriptors (UPC parameters) for AAL2 multiplexed voice traffic streams, then verify their accuracy. All modeling and simulation was done with the BONeS Designer simulation package [24].

The virtual buffer simulation setup is as shown in Figure 4.6. A previously designed AAL2 transmitter [14] is used as a source for generating AAL2 traffic. The AAL2 output is fed to a number of parallel virtual buffers, each served at a different rate. The output rate of the transmitter is given a practically infinite value for this traffic stream (10 Mb/s) so that there is no traffic shaping done at the transmitter.

4.4.1 Simulation Models

4.4.1.1 Sources

Each individual voice source is modeled as an On-Off source. The On and Off times are assumed to be exponentially distributed, and the sources have a constant rate when they are ON. This model for the On-Off sources is derived from [22]; it has been verified by analyzing additional recordings of telephone conversations [6]. The values of mean On and Off times estimated from these recent recordings differ somewhat from the values in [22].

4.4.1.2 AAL2 Transmitter

A previously designed AAL2 transmitter is used for multiplexing the voice traffic into a single AAL2 ATM traffic stream. The transmitter has been designed on the basis of the ITU-T draft specification [1]. It has been used in previous studies of AAL2 performance characterization [14] and finding the maximum number of users subject to a 95^{th} percentile delay constraint [7].

4.4.2 Parameters Used in Simulation

4.4.2.1 Fixed Simulation Parameters

- Mean ON time (1.230s)
- Mean OFF time (1.373s)
- CPS packet size (20 bytes).
- Voice coding rate (32 kb/s).
- CU Timer (5.1 ms).

The combination of CPS packet size and voice coding rate results in a packetization time of 5 ms. The CU timer value (time the transmitter waits before sending a cell that is not full) is selected as 5.1 ms to make sure that the cells are densely packed (two packets are packed in a cell even for a single active user). The maximum time delay for a packet is 10.1 ms (5 ms of packetization delay and 5.1 ms CU Timer). The mean On and mean Off times are taken from the recent analysis of speech files [6]. The number of CPS packets simulated for each parameter set is 60,000.

4.4.2.2 Variable Simulation Parameters

The simulations are done for 1 through 12 users in steps of 1 and from 12 through 48 users in steps of 6. The service rate is varied from a value slightly larger than effective mean rate given by,

$$effective\ \ mean\ \ rate = n(\frac{1.230}{1.230+1.373})(32)(\frac{23}{20})(\frac{53}{47})\text{kb/s} \tag{4.5}$$

to a value more than twice effective peak rate, where

$$effective\ peak\ rate = n(32)(\frac{23}{20})(\frac{53}{47})kb/s \tag{4.6}$$

Twenty rates in this range are taken to obtain the curves.

4.5 Results and Discussion

4.5.1 UPC Parameter Results

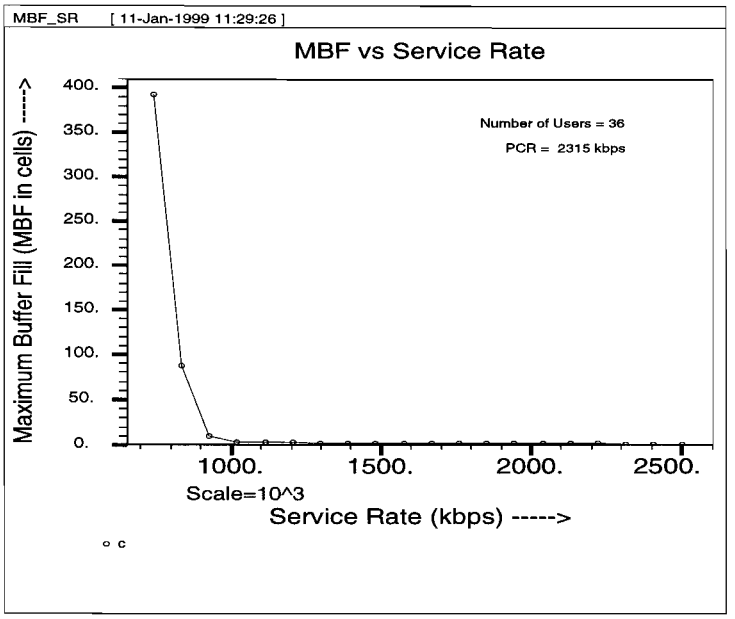


Figure 4.7: MBF vs. Service Rate for 36 users

The Virtual Buffer simulations yield a curve between MBF and the service rate (see Figure 4.7). From this curve we obtain the following.

- PCR and CDVT : As established in Section 4.3, the minimum service rate that corresponds to MBF=1 is taken as the minimum PCR when CDVT equals $1/PCR = T_0$. This choice ensures that the traffic is GCRA(T_0, T_0) conforming. Figure 4.8 shows two estimates of PCR vs. the number of users. The value "calculated PCR" corresponds to the minimum virtual buffer service rate for MBF of 1 cell. Twice the effective_peak_cell_rate (equation 4.6, expressed in cells/sec) is shown as "estimated PCR." We see that, for an AAL2 traffic stream, it may be possible to use twice the effective_peak_cell_rate to estimate the required PCR without resorting to simulation.

- Curves BT vs. SCR and MBS vs. SCR (see Figures 4.9 and 4.10): Each value of virtual buffer service rate is taken as SCR and the corresponding maximum buffer fill (MBF) is converted to the near-minimum burst tolerance (BT) using equation

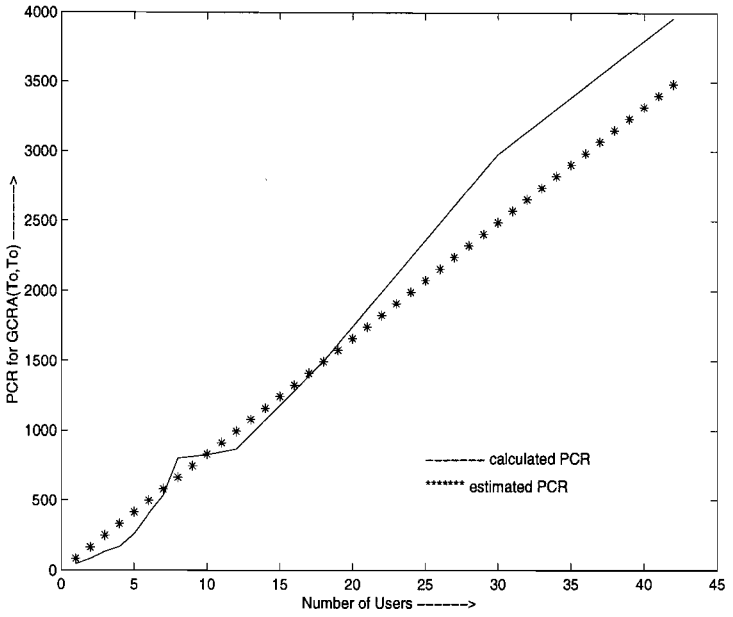


Figure 4.8: PCR Estimates vs. Number of Users

4.1, resulting in Figure 4.9. By interpolation in Figure 4.9, we can find an infinite number of near-minimal (SCR,BT) pairs that describe the given traffic stream. We can also obtain Figure 4.10 which shows the curve of near-minimal MBS vs. SCR using equation 4.2.

Any point on the curve of Figures 4.9 or 4.10 will result in GCRA conformance. The selection of a particular point on the SCR, MBS (or BT) curve can be done by different methods.

Method 1: Choose the (SCR,MBS) or (SCR,BT) pair so as to minimize the effective bandwidth requirement. The effective bandwidth might be minimized at larger values of MBS and correspondingly smaller values of SCR. This may not always be an ideal choice. The trade-offs involved in selecting a bandwidth efficient (SCR,BT) pair are discussed in [21].

Method 2: Limit MBS to a reasonable value and select the corresponding SCR value. The SCR values for MBS of 50 cells are given in Table 4.1. The incremental SCR for each additional user is also given. Table 4.1 shows that the incremental SCR required to support each additional user tends to a constant value as the number of users increases. It is clear that this incremental SCR must be lower-bounded by the effective mean rate per source of 19.6 kb/s (from equation (5) with n = 1).

4.5.2 Verification Using the Dual Policer Configuration

This section verifies that the values of PCR, CDVT, SCR and BT found above form a near-minimal UPC parameter set, using a simulated dual policer configuration shown in Figure 4.1. The conformance test has been done in 3 stages after selecting a point from the

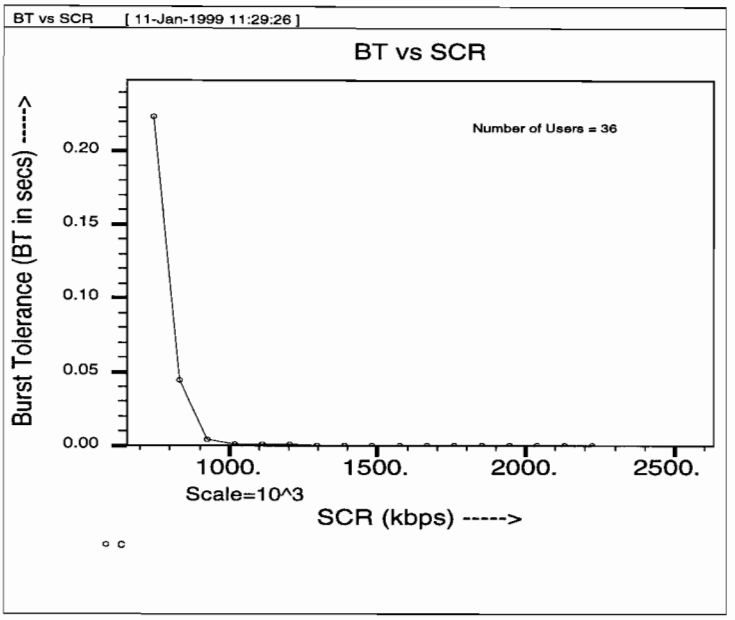


Figure 4.9: BT vs. SCR for 36 users

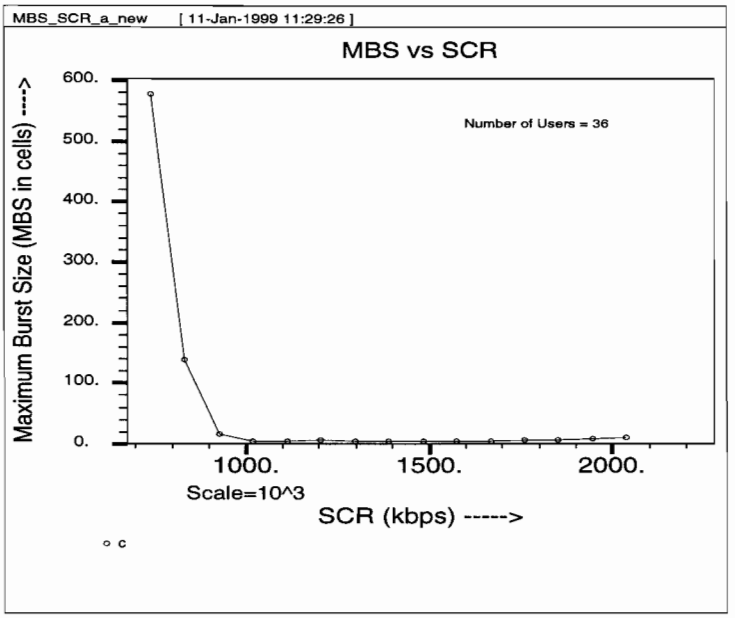


Figure 4.10: MBS vs. SCR Estimates for 36 users

Number of Users	SCR (kb/s) at MBS = 50 cells	SCR (kb/s) per User
1	40.5	40.50
2	78.5	39.25
3	113	37.67
4	150.5	37.63
5	191	38.20
6	215	35.83
12	358	29.80
18	535	29.70
24	699	29.13
30	832	27.70
36	900	25.00
42	1060	25.20
48	1158	24.13

Table 4.1: SCR Values for MBS of 50 Cells

(SCR,BT) curve for a fixed PCR (see Figure 4.9). The following point has been selected for verification:

PCR: 2315 kb/s, SCR: 835 kb/s, BT: 0.044502 s

1) Conformance test with variation in PCR: The values of SCR and BT are kept constant and the value of PCR is varied. PCR is decreased from its minimal value of 2315 kb/s in steps of the servicing rates used in the virtual buffer simulation. Table 4.2 shows that even a slight decrease in PCR gives violations in the PCR policer. As expected, SCR policer conformance is not strongly affected by PCR variations.

2) Conformance test with variation in SCR: The values of PCR and BT are kept constant and the value of SCR is varied. SCR is decreased from its minimal value of 835 kb/s in steps of 50 kb/s (smaller than the virtual buffer service rate increment). Table 4.3 shows that even a slight decrease in SCR gives violations in the SCR policer.

3) Conformance test with variation in BT: The values of SCR and PCR are kept constant and the value of BT is varied. BT is decreased from its minimal value of 0.044502 s in steps of T_s first and by larger values later. Table 4.4 shows that even slight decrease in BT gives violations with the SCR policer.

These tests verify that the traffic descriptors (PCR = 2315 kb/s, SCR = 835 kb/s, BT = 0.044502 s) found are a near-minimal set for the given traffic stream; that is, reduction in any one value results in GCRA violation (either PCR or SCR).

PCR (kb/s)	Number of violating cells with PCR policer	Number of violating cells with SCR policer
2315.0	0	0
2222.5	1	0
2130.0	1	0
2037.5	1	0
1945.0	21	0
1852.5	75	0

Table 4.2: PCR violation with SCR = 835 kb/s and BT = 0.044502 s

SCR (kb/s)	Number of violating cells with PCR policer	Number of violating cells with SCR policer
835	0	0
785	0	74
735	0	332
685	0	1056
635	0	2105

Table 4.3: SCR violation with PCR = 2315 kb/s and BT = 0.044502 s

BT seconds	Number of violating cells with PCR policer	Number of violating cells with SCR policer
0.044502	0	0
0.044085	0	1
0.043485	0	2
0.042885	0	3
0.030	0	20
0.025	0	31

Table 4.4: BT violation with SCR = 835 kb/s and PCR = 2315 kb/s

4.6 Calculation of CBR bandwidth requirements for AAL2 Users

This section focuses on the estimation of bandwidth requirements for AAL2 traffic when using a CBR VC instead of a VBR VC. In addition, the focus here is on analytic estimates in contrast with measurement based estimation. We assume that only voice applications are running on AAL2. An effort has been made in a previous work [18] to map various bandwidth calculation schemes for the AAL2 case. We select a methodology suggested by NEC [10] and use this as suggested in [18]. Here we discuss only the NEC CAC scheme and how it can be used in the AAL2 context.

There have been a number of proposals [10] [11] [12] [13] for calculating bandwidth requirements for multiplexed flows given the traffic descriptors of individual flows. The NEC Multi-class scheme has been selected for bandwidth calculations because of the following reasons;

- Suitability for heterogeneous traffic.
- Rigorous non-conservative approach that gives better approximations.
- Found to give results that match simulation results for the AAL2 case [18].

4.6.1 NEC Multi-class CAC

The NEC Multi-class CAC methodology as described in [10] takes into consideration the user's cell loss requirements, buffer size at the satistical multiplexers, and effect of individual traffic streams at the ATM multiplexers. The assumptions involved are;

- M x M input/output buffered switch.
- Buffer partitioning between traffic classes (CBR, real time VBR (rt-VBR), non-real time VBR (nrt-VBR), ABR, UBR) at each input port to reduce interaction between them.
- Frame-based scheduling method to allocate bandwidth where allocating a bandwidth of C_k to class k is equivalent to assigning n_k time slots to this class out of each frame, where

$$n_k = \left\lceil N_f \frac{C_k}{C_{link}} \right\rceil$$

  N_f represents the total number of time slots available in a frame.

The QoS requirements are defined in terms of Cell Loss Ratio (CLR) and Cell Delay Variation (CDVT). The guarantees given are;

- CLR $< \epsilon_1$

- $\text{Prob}[\text{CDV} > t_{max}] < \epsilon_2$

Here CDV is defined as the difference between the inter-arrival time and inter-departure time of consecutive cells from the same virtual connection. Although the CAC is class-based, the connections *within* each class are treated as heterogeneous sources having different UPC parameters. This feature of the NEC CAC scheme makes it ideal when dealing with heterogeneous voice sources. But it is to be noted that in this scheme all the connections share the same FIFO and hence get the same QoS (CLR and CDVT).

The following steps are followed in deciding whether a new connection with a given set of UPC parameters (PCR = λ_p^*, SCR = λ_s^*, MBS = B_s^*) can be supported on a VC of total capacity C_{cbr}^{max} with active users accounting for C_{cbr}^{old}.

- *Step I*: Construct fictitious "on/off" source model using

$$T_{on}^* = \frac{B_s^*}{\lambda_p^*}$$
$$T_{off}^* = \frac{B_s^*}{\lambda_p^*}\frac{(\lambda_p^* - \lambda_s^*)}{\lambda_s^*} \tag{4.7}$$

- *Step II.a*: Construct a Lossless Multiplexer model to find the total bandwidth required to support new+existing connections (C_{cbr}^{new})
  - Calculate C_{cbr}^{new} using eqn.

$$C_{cbr}^{new} = max\left((\lambda_p^* + \sum_{i=1}^{n}\lambda_p^i)(1 - \frac{B_{cbr}}{B_s^* + \sum_{i=1}^{n} B_s^i})^+, \lambda_s^* + \sum_{i=1}^{n}\lambda_s^i\right) \tag{4.8}$$

    where B_{cbr} is the buffer allocated for CBR traffic class and $[x]^+$ means max(0,x).
  - If CDV is not satisfied, recalculate C_{cbr}^{new} replacing B_{cbr} by $C_{cbr}^{new} * t_{max}$.
  - Additional bandwidth required to support new connection

$$\delta_1 = C_{cbr}^{new} - C_{cbr}^{old} \tag{4.9}$$

- *Step II.b*: Find C_{cbr}^{new} using a Statistical Multiplexer Model.
  - Construct a modified "on/off " source model

$$\lambda_H^* = min(1, \frac{T_{on}}{T_N})\lambda_p^* + [1 - \frac{T_{on}}{T_N}]^+\lambda_s^*$$
$$\lambda_L^* = [1 - \frac{T_{on}}{T_N}]^+\lambda_s^*$$

    where T_N is time required to empty half-filled buffer.
  - Calculate C_{cbr}^{new} using the equation

$$C_{cbr}^{new} = \bar{M}_{new} + \zeta * \sigma_{new} \tag{4.10}$$

where $\bar{M}_{new}$ and σ_{new} are the mean and variance of the aggregate arrival process due to all admitted CBR connections after the new CBR connection is admitted. ζ is dependent on $min(\epsilon_1, \epsilon_2)$

- Find additional bandwidth required to support the new connection

$$\delta_2 = C_{cbr}^{new} - C_{cbr}^{old} \tag{4.11}$$

- Admit the new connection if capacity $\Delta_{cbr} = min[\delta_1, \delta_2]$ is available in the free pool of bandwidth (i.e. $C_{cbr}^{new} \leq C_{cbr}^{max}$).

4.6.1.1 Application to AAL2 traffic

The AAL2 traffic parameters depend on the *On-Off times* of speech [6], *voice coding rate, CPS packet size, number of users* using the same VC and the value of *Timer-CU* used. The voice coder associated with each user feeds data at the coding rate to the AAL2 transmitter during the On-times of the speech. We assume no data is sent in the duration of the Off-time. The AAL2 overhead is to be considered when calculating bandwidth. Assuming that all the cells are fully filled (a reasonable assumption for large number of users), the overhead is given by:

$$\begin{aligned} \text{Overhead Factor} &= \left(\frac{\text{ATM Cell Size}}{\text{Max CPS PDU Size}}\right) * \left(\frac{\text{Packet Size} + 3}{\text{Packet Size}}\right) \\ &= \left(\frac{53}{47}\right) * \left(\frac{\text{Packet Size} + 3}{\text{Packet Size}}\right) \end{aligned} \tag{4.12}$$

$$\text{Effective Peak Rate(P)} = \text{Overhead Factor} * \text{Coding Rate} \tag{4.13}$$

This is not true for small number of users. A method to estimate the overhead for small number of users given in [18] is used and will not be discussed here.

We assume that all voice traffic (irrespective of the voice coder used) has the same priority. We put a limit on the queuing delay in the AAL2 transmitter. A 95^{th} percentile queuing delay $\leq 2ms$ is assumed. This means 95% of the cells are guaranteed to experience a queuing delay of less than or equal to 2 ms. The delay guarantee does not include the packetization delay. Users using lower bit rate coders have a higher packetization delay than users of high bit rate coders for identical packet sizes. Since we assume that all the CPS packets get the same priority, we cannot guarantee a common delay bound on $\text{Packetization Delay} + \text{Max Queuing Delay}$ in the case of heterogeneous users (at least with the original AAL2 specification). Also, a 95^{th} percentile CDV $\leq 2ms$ is assumed.

The Max Queuing Delay maps directly to the Buffer size as follows:

$$\text{Max Queuing Delay} = \left(\frac{\text{Buffer Size}}{\text{Link Rate}}\right) \tag{4.14}$$

Therefore,

$$\text{Buffer Size}\,(x) = (\text{Max Queuing Delay}) * \text{Link Rate} \tag{4.15}$$

Thus the guarantees given are:

$$\text{Cell Loss Probability(CLP)} \approx \text{Prob[Buffer Fill} > x] \leq 0.05 \tag{4.16}$$

and

$$\text{Prob[CDV} > 2\text{ms}] \leq 0.05 \tag{4.17}$$

The values of λ_p^*, B_{cbr}, B_s^*, and λ_s^* can be obtained using the following equations:

$$\lambda_p^* = \text{EffectivePeakRate(P)} \tag{4.18}$$

$$B_{cbr} = \text{BufferSize(x)} \tag{4.19}$$

$$\text{Load}\ (\rho) = \frac{\text{Mean On Time}}{(\text{Mean On Time} + \text{Mean Off Time})} \tag{4.20}$$

$$B_s^* = \text{Mean burst Size} = (\text{Mean On Time}) * (\text{Effective Peak Rate}) \tag{4.21}$$

$$\lambda_s^* = \text{Mean cell rate} = (\text{Load}) * (\text{Effective Peak Rate}) \tag{4.22}$$

Also, we have

$$\text{Mean}(m) = \rho P \tag{4.23}$$

$$\text{Variance}(\sigma^2) = \rho(P - m)^2 + (1 - \rho)m^2 \tag{4.24}$$

4.6.2 Results

The following curves obtained by applying the NEC Multi-class CAC methodology to AAL2 traffic are shown to demonstrate the use of the methodology.

1) Bandwidth per user vs. number of users (see Figures 4.11 and 4.12): The bandwidth required per user is plotted against the number of users for three different coding rates (16 kb/s, 32 kb/s, 64 kb/s). We can observe that the bandwidth required per user is high at small numbers of users. At higher numbers of users, the value gets close to but larger than:

$$(\text{coding\_rate})(\text{speech\_activity\_factor})\frac{(\text{voice\_packet\_size} + 3)}{(\text{voice\_packet\_size})}\frac{53}{47} \tag{4.25}$$

In Figure 4.11, a mean On-time of 420 ms and a mean Off-time of 580 ms is assumed. In Figure 4.12, a mean On-time of 1.232 sec and a mean Off-time of 1.373 resulting in a speech activity factor of 0.473 is assumed. These values have been derived from a previous work

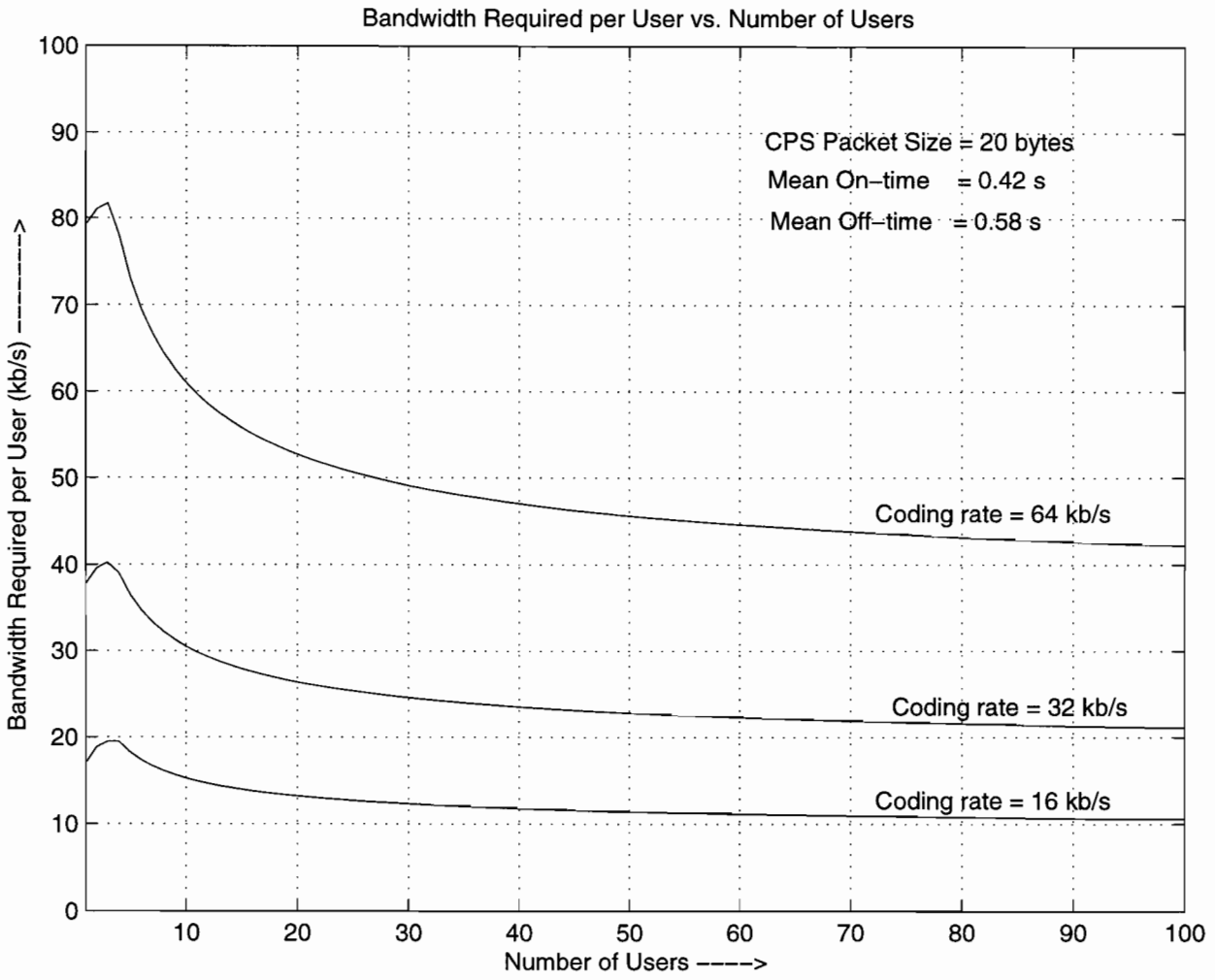


Figure 4.11: Bandwidth Required per User for Speech Activity Factor 0.420

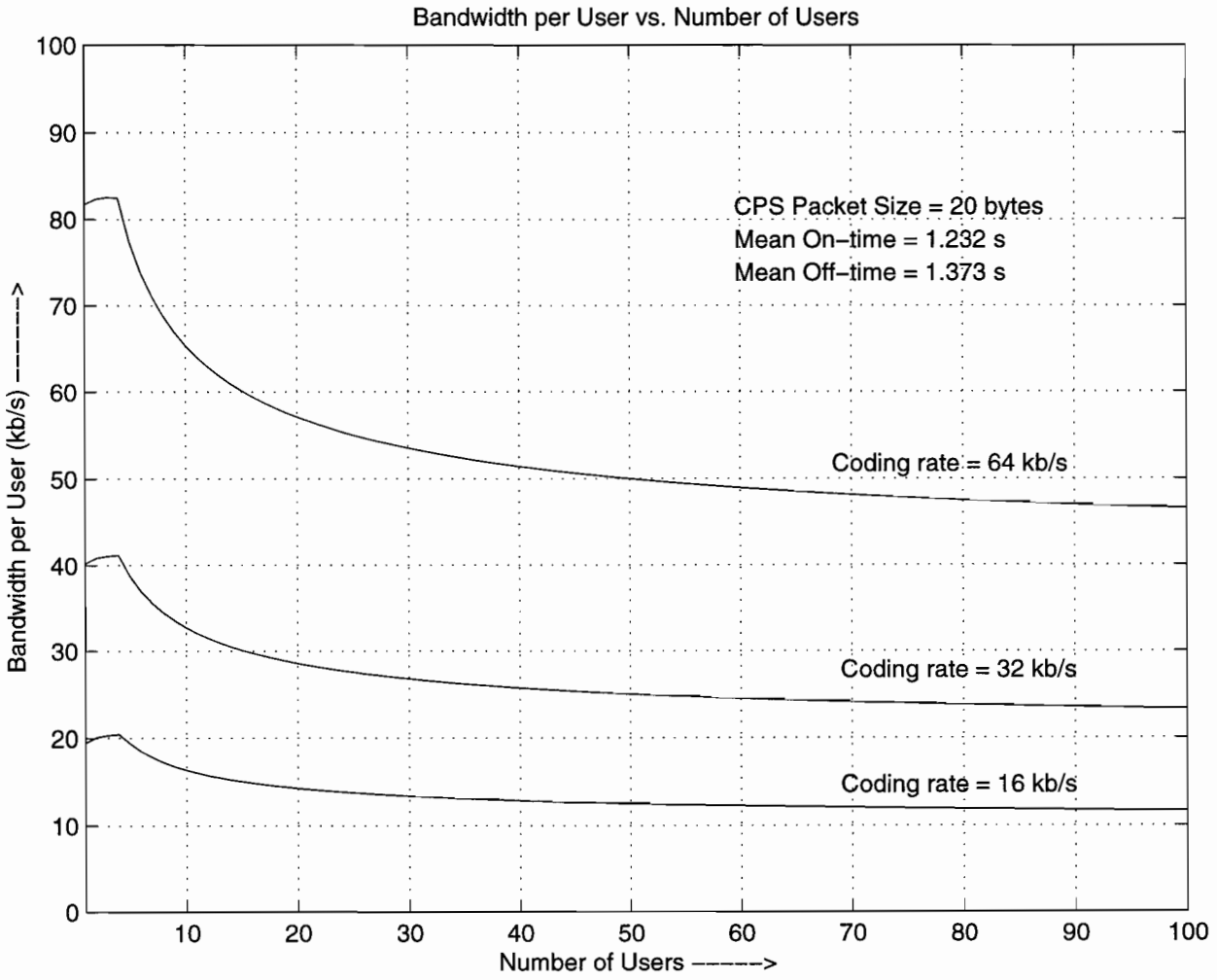


Figure 4.12: Bandwidth Required per User for Speech Activity Factor 0.473

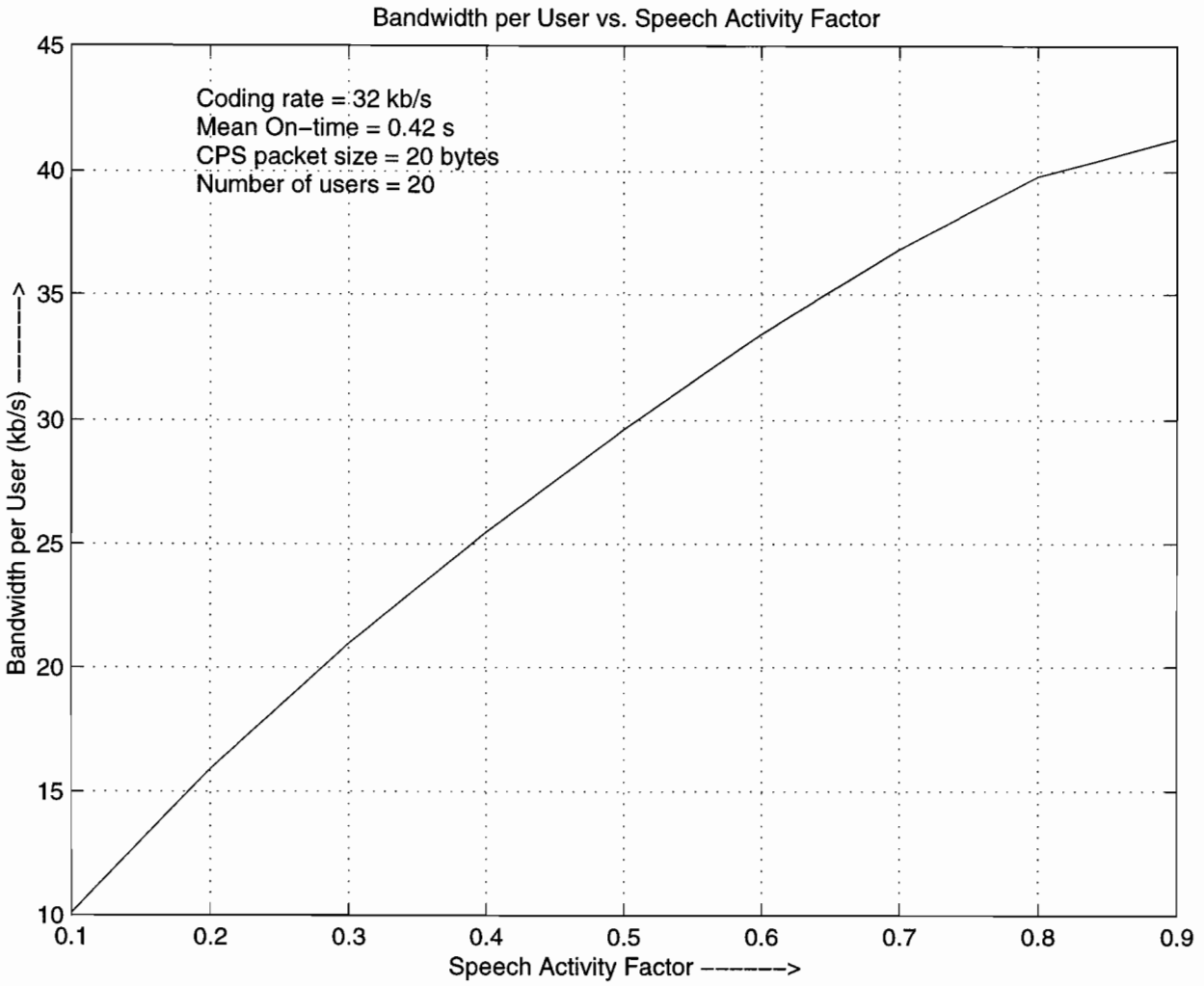


Figure 4.13: Variation in Bandwidth Required per User with Speech Activity Factor

on voice activity statistics [6]. A CPS packet size of 20 bytes is assumed for all coding rates. For the given set of parameters the above formula yields values of 39.2 kb/s, 19.63 kb/s and 9.81 kb/s respectively for 64 kb/s, 32 kb/s and 16 kb/s voice coding rates.

2) Bandwidth per user vs. speech activity factor (see Figure 4.13): The bandwidth required per user when 20 users are in the system is plotted against the speech activity factor. Speech activity values ranging from 0.10 to 1.0 are taken in steps of 0.1 for generating the plot. The mean On-time is kept constant at 420 ms.

The above plots demonstrate the use of the NEC Multi-class CAC algorithm in calculating the bandwidth requirements for AAL2 users. In the next chapter we move on to describe the AAL2 CAC algorithm.

Chapter 5

Self-Configuring CAC for AAL2 with Load Estimation (SCALE)

In this chapter a CAC procedure for AAL2 is suggested. The algorithm is described in detail. Also, some of the possible modifications and extensions to the algorithm are identified. The evaluation of the algorithm using a load distribution curve example is presented in the next chapter.

5.1 Summary of Requirements for AAL2 CAC

In chapter 3, we identified that using multiple SVCs instead of a single bigger PVC is advisable in a case where there is a high load variation. As high load variation is inherent to telephone traffic, the CAC scheme should be designed to be efficient even in such a case. We therefore incorporate demand-based setup and tear-down of SVCs into the CAC algorithm. The AAL2 CAC function should take care of requesting additional bandwidth from the network in the form of an additional SVC. There is a possibility of the network rejecting a request for a VC from the AAL2 node. The CAC function should minimize the effect of the network rejection probability on the actual call rejection probability. In doing this, high bandwidth efficiency should also be maintained. Incoming calls should be assigned to VCs in an efficient manner so as to minimize the overall bandwidth usage. Also, it is possible that the users use different coding schemes. This aspect of heterogeneous users has to be considered.

Various extensions to the existing AAL2 setup have been proposed [8] [18]. The CAC mechanism described here should be designed such that it can be easily extended to incorporate these suggestions. The CAC mechanism has been designed for the simple AAL2 case. Possible changes to the CAC to suit the actual network design are suggested.

5.2 The AAL2 CAC Algorithm

The CAC function is called each time a request for a new call is generated or an active call terminates its connection. A flowchart showing the set of actions taken by the CAC function in each of the above cases are shown in Figures 5.1 and 5.2. The algorithm is

discussed below. In the algorithm, the notation VC_X represents a VC which is ranked 'X' in decreasing order of number of users supported within the VC.

5.2.1 Admission Request

When a new user comes in with a set of traffic descriptors which (can be determined from the coding rate and voice statistics in case of voice traffic), the CAC functions goes through the following steps (see Figure 5.1):

- *Step I*: Starting from VC_0 till VC_N search for the first occurrence of a VC(VC_A) having enough spare bandwidth to support the new user. If such a VC is not found, that is, no existing VC can support the user, put admission request to pending and go to *step V*.
- *Step II*: Accept the Call and update the bandwidth usage value of VC_A.
- *Step III*: Reorder the VCs in decreasing order number of users using the VC.
- *Step IV*: If the user is assigned to VC_N and if the bandwidth usage of VC_N is more than an *upper threshold*, go to *step V*. Else go to *step VI* (exit).
- *Step V*: Ask the network for an additional VC. If the VC request is rejected, reject any pending admission request and go to *step VI* (exit). If the VC request is granted, update the record of number of VCs and go to *step II* with $VC_A = VC_N$.
- *Step VI*: Exit.

5.2.1.1 Discussion

The aim of the CAC process with multiple VCs should be to reduce the bandwidth used by reducing the number of active VCs. For doing this, a high VC utilization has to be achieved. We therefore try to use the existing VCs efficiently and withdraw VCs that are in excess of the requirement.

The VC that has the highest number of users has the highest expected holding time. At any given time, the holding time of a VC can be defined as the time taken for all the calls in the VC to terminate provided no new call enters the VC. In alternate terms, it is equal to the longest holding time of one of the existing users. Since the VC that has users cannot be discarded even it is underutilized, the VC that has the largest number of users is expected to take the longest time to be withdrawn given that no new users enter the system (under a reasonable assumption that all users have identical call holding time distributions). So we try to allocate the user to the VC that is expected to stay on for a longer time. This is the reason for ranking the VCs according to the number of users supported. Since there is a possible change in this ranking each time a new user enters the system, the VCs are reordered every time a new user enters the system or an existing user leaves the system.

If all the VCs except VC_N are fully filled and VC_N is filled beyond a level of usage (upper threshold), there is a high probability that in a reasonable time, VC_N will also be

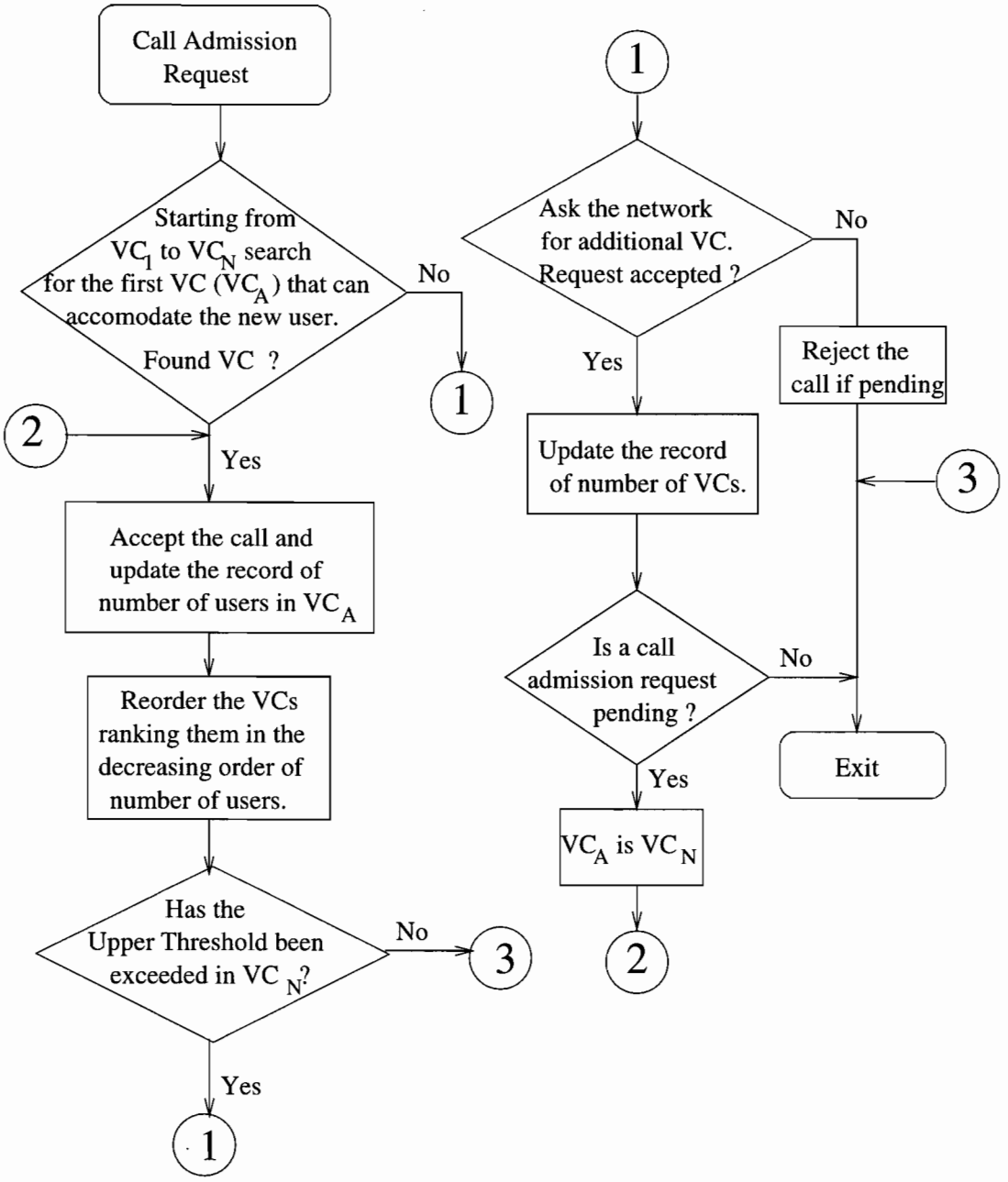


Figure 5.1: CAC Algorithm at call admission

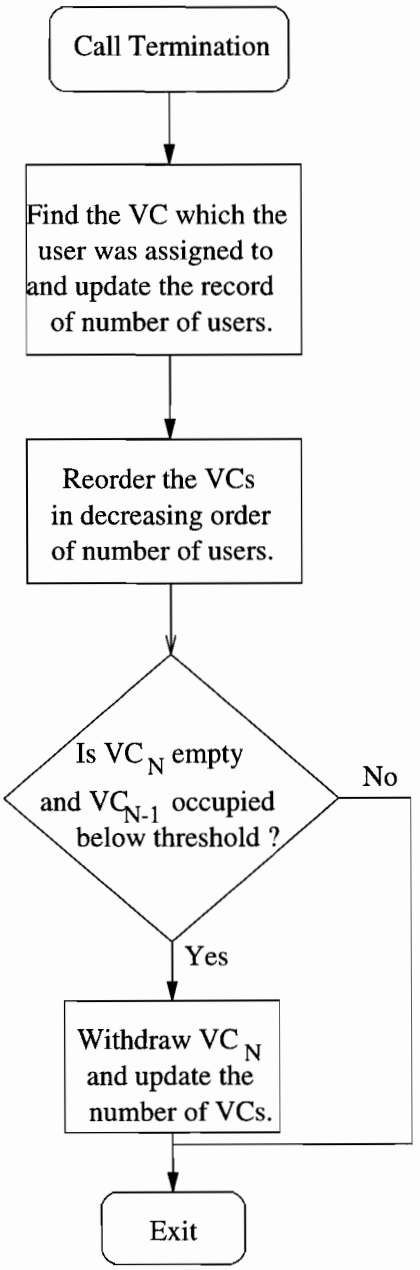


Figure 5.2: CAC Algorithm at call termination

fully filled while all other VCs remain fully filled. The call entering the system at this point will either have to be rejected or kept in waiting till a request for an additional VC is granted by the network. This will increase the waiting time and also the call rejection probability (due to the network rejecting the request for a VC) for such users. We try to reduce this by introducing the concept of an upper threshold. If VC_N reaches an *upper threshold* and all other VCs are fully filled, we request a VC from the network, anticipating more users. Setting up a VC anticipating the load will reduce the efficiency but would also reduce the call rejection probability.

There are number of ways the *upper threshold* can be set. The most simple way is to make the *upper threshold* equal to a percentage of the total bandwidth associated with the VC. A more efficient way is to set the *upper threshold* such that when the VC is just occupied to this level, there is exactly enough bandwidth for some small number of users.

The VC requests are always associated with call admission requests. Periodic VC requests are not done even if the system is loaded beyond the upper threshold. If the system is full when a call request is received, the new call is put on wait pending a VC request from the network in a case where the VC request is triggered by the entry of the users. This assumes that a longer waiting time is better than call rejection.

Another important issue that has to be resolved is the size of the VCs used. For simplicity of the CAC process we assume that all the VCs used are of identical size. In the case of homogeneous users, the VC size should be fixed such that it *exactly* supports a fixed number of users. Spare bandwidth in the VC which cannot support a user decreases the bandwidth efficiency. This might be unavoidable in the case where heterogeneous users are supported. The optimum VC size is a compromise between statistical multiplexing gain associated with larger VCs and bandwidth efficiency under load variation associated with smaller VCs. This is dependent on the load variation. The most efficient VC size for the given load variation statistics can be found by running simulations using the above algorithm.

5.2.2 Call Termination

When an active user associated with a VC in the system decides to terminate a call, the CAC function goes through the following steps (see Figure 5.2):

- *Step I*: Update the bandwidth usage value of the VC that the terminating call was assigned to.
- *Step II*: Reorder the VCs in the decreasing order of number of users.
- *Step III*: If VC_N is completely unused and the bandwidth usage of VC_{N-1} is below a lower threshold, go to *Step IV*. Else, go to *Step V* (exit).
- *Step IV*: Withdraw VC_N and update the record of number of VCs.
- *Step V*: Exit.

5.2.2.1 Discussion

The call termination process is fairly simple compared to the call admission process. The VC that the terminating user was assigned to is updated. The reordering of the VCs is done because of the change in the number of users. If VC_N is empty and VC_{N-1} is utilized below a lower threshold, the system is assumed to be under utilized and hence VC_N is withdrawn. The lower threshold values should be smaller than the upper threshold value to avoid frequent new VC requests and VC withdrawls. The lower threshold is set in a manner similar to setting the *upper threshold*.

5.2.3 Extensions to the CAC Algorithm

The CAC algorithm in its simple form is presented in the previous section. We now discuss the details that have to be taken care of when using the algorithm for the heterogeneous user case and the possible modifications to be done when the AAL2 setup has provisions for dynamic source coding rate control [18] or different QoS for users [8].

5.2.3.1 Heterogeneous Users

Depending on the network setup, there is a possibility of various coding schemes being used for voice. The coding scheme used might depend on a number of factors including the user's preference and the capability of the user end equipment. Since there are a number of bandwidth calculation methodologies like NEC CAC that are inherently designed for multi-class systems, bandwidth calculations for heterogeneous users is not difficult. The issues that have to be taken care of in the heterogeneous case are setting the threshold values and size of the VCs used. The threshold value may be set such that there is enough spare capacity to support a few users of a particular type or a few users of some or all types of users. The bandwidth required for this can be precomputed and hence the threshold is fixed. It is difficult to set the VC size *exactly* to support integral number of users because of different types of users. The optimum VC size is best decided through simulation of the above algorithm for a given load variation. This is demonstrated in chapter 6.

5.2.3.2 Further Extensions

In a case where only voice is carried on AAL2, the different users might not need different QoS requirements. But if integrated voice and data traffic is carried over AAL2, QoS differentiation [8] might be required with voice traffic getting better delay QoS. This kind of service can be given in two ways. The easiest way is to have separate AAL2 VCs (in effect separate AAL2 transmitters) for each QoS class. The CAC algorithm proposed can be directly used in such a scenario by applying the CAC algorithm separately to each QoS class. But assigning different VCs to each type of traffic is not bandwidth efficient especially when there is a small number of users in each QoS class. A modified AAL2 transmitter has been proposed in [8] that supports QoS differentiation within the same AAL2 VC. For using the SCALE for this case we need to develop a method to calculate the bandwidth requirements for users under multiple QoS constraints which is a non

trivial task. Also the threshold values might have to be set such that there is spare capacity available at least to support a fixed number of users of a particular QoS class. The optimum VC size can be found through simulation, however.

The AAL2 system might experience high VC rejection rates as the network gets closer to full utilization. High VC rejection rates increase the call rejection probability. A short term solution to this situation is to reduce the coding rates (source coding rate control [18]) of some or all of the users. The proposed system has two coding rates. The higher coding rate is preferred and is used under normal conditions. The coding rate is stepped down under congestion. For the implementation of such a system, the CAC function should be able to control the coding rate of the users. The bandwidth requirements when using the low bit rate coder should be re-calculated and the call admissions should be based on the re-calculated bandwidth values. The *upper threshold* should be set such that a fixed number of users can be supported at the higher coding rate. Alternately, a VC request can be made only when the system is fully filled. In this case the new call is admitted by reducing the coding rates of some of the users temporarily if the VC request is rejected and increasing the rate once the VC request gets accepted.

Chapter 6

Evaluation of SCALE

The CAC algorithm developed in Chapter 5 is evaluated for two hypothetical load variation curves, one for homogeneous users and another for heterogeneous users. The optimum CAC parameters are found for the given load statistics by simulating the CAC algorithm.

6.1 Parameters for SCALE

The parameters to be used in the SCALE algorithm primarily depend on the load statistics and the QoS given to the customer in terms of call rejection probability. As pointed out earlier, if the load variation is low and the the number of users is small, it actually might be better to use a single PVC instead of multiple SVCs. When using multiple SVCs, a smaller Upper Threshold value leads to a lower call rejection probability and a higher bandwidth usage. It is for the service provider to do a trade off between rejection probability and bandwidth usage. A smaller value of Lower threshold results in less VC request/withdraw activity. The service provider can simulate the CAC algorithm with his own load statistics, QoS and bandwidth constraints to get the optimum load variation statistics. We demonstrate this by simulating the CAC algorithm for two hypothetical load variation curves for cases of homogeneous and heterogeneous users.

6.2 Simulation Setup

6.2.1 Load Variation Statistics

The following load variation statistics were assumed. The assumption is purely hypothetical and is used only to demonstrate the process of finding the optimal CAC parameters for the case in hand. This is to be replicated by the service provider with his own load variation statistics.

- Homogeneous Users: Each day was divided into four 6 hour *load time zones* based on the load during the time. Each time zone experiences exponential call arrivals at a given mean rate. The mean call inter-arrival times for the four different times zones is given in Table 6.1. The call holding times are also assumed to be exponential with

Duration (hours)	Interarrival time (seconds)	Average Number of users
6	1.5	120
6	2.0	90
6	3.0	60
6	9.0	20

Table 6.1: Load Variation for Homogeneous User Case

a mean of 180 secs. Table 6.1 also gives the values of mean number in the system which is calculated as the ratio of mean call holding time and mean inter-arrival time between two calls. The average number of active users is 72.50. A coding rate of 32 kb/s is the assumed coding rate for the calls.

- Heterogeneous Users: It was assumed that three types of users using three different coding rates of 64 kb/s, 32 kb/s and 16 kb/s submit their calls to the system. Identical load time zones are assumed for each of the coding rates. The average interarrival time for each user type and the resulting values of average number of active users are given in Table 6.2. The average number of active users in the system is 78.75.

6.2.2 Simulation Parameters

- Voice Statistics

  An exponential On-Off model [22] [6] is assumed with the following means (taken from [6]):

  - Mean On-time: 1.232 sec
  - Mean Off-time: 1.373 sec

- Delay QoS Guarantees

  A 95^{th} percentile delay equal to 2 ms and a 95^{th} percentile CDV equal to 2 ms in the AAL2 transmitter are assumed as the delay QoS requirements. NEC CAC scheme described in Chapter 4 is used for estimating the bandwidth required.

Duration (hours)	Interarrival time per type of user (seconds)	Average Number of users per type of user
6	4.5	40
6	6.0	30
6	9.0	20
6	12.0	15

Table 6.2: Load Variation for Heterogeneous User Case

- Voice Coding Rates

  The following common voice coding rates are assumed:

  - Homogeneous User Case: 32 kb/s
  - Heterogeneous User Case: 64 kb/s, 32 kb/s, 16 kb/s

6.2.3 Simulation Model

The simulation model for the CAC algorithm was developed using *Extend* [25], a network simulation tool designed to work on Windows PCs or Macintosh computers. The model was fully verified and tested.

6.3 Results

The performance of the proposed CAC algorithm can be evaluated by measuring the gain in bandwidth achieved by using the algorithm. The bandwidth gain was measured in terms of the *bandwidth-time product* and with reference to the case of using a single PVC of capacity equal to support the peak load of the day. Bandwidth-time product is the integral of the bandwidth used with respect to time of usage. *Call rejection probability* is another important parameter for evaluating the CAC algorithm. The following results represent the performance of the CAC algorithm for both homogeneous and heterogeneous user cases.

6.3.1 Homogeneous User Case

Percent Bandwidth-Time Product Gain vs. VC Capacity (see Figure 6.1): The gain in bandwidth-time product obtained by using the proposed CAC algorithm is shown in Figure 6.1. The gain is calculated with reference to the case of using a single PVC of capacity equal to support the peak load of the day. In the simulation, the peak load was 149 users and hence the PVC is assumed to be of capacity equal to 3361 kb/s (needed to support 149 users, calculated using NEC CAC). The *upper threshold* is set such that there is spare capacity to accomodate 1 user in VC_N. The lower threshold is set such that VC_{N-1} has spare capacity to accomodate 2 users. Note that the VC capacity can be measured in number of users due to the homogeneity of the user population.

From Figure 6.1 we see that the optimum VC capacity for the given set of parameters is 35 users per VC. This is a compromise between larger multiplexing gain associated with VCs of higher capacity and efficient VC utilization associated with VCs of lesser capacity. Also observe that at very small number of users per VC the bandwidth gain from using multiple SVCs is overcome by the loss in bandwidth efficiency due to smaller multiplexing gain.

Percent Bandwidth-Time Product Gain vs. Upper Threshold (see Figure 6.2): A smaller upper threshold value reduces the call rejection probability, but decreases the utilization of the VC. Figure 6.2 shows how the bandwidth-time product gain increases with the

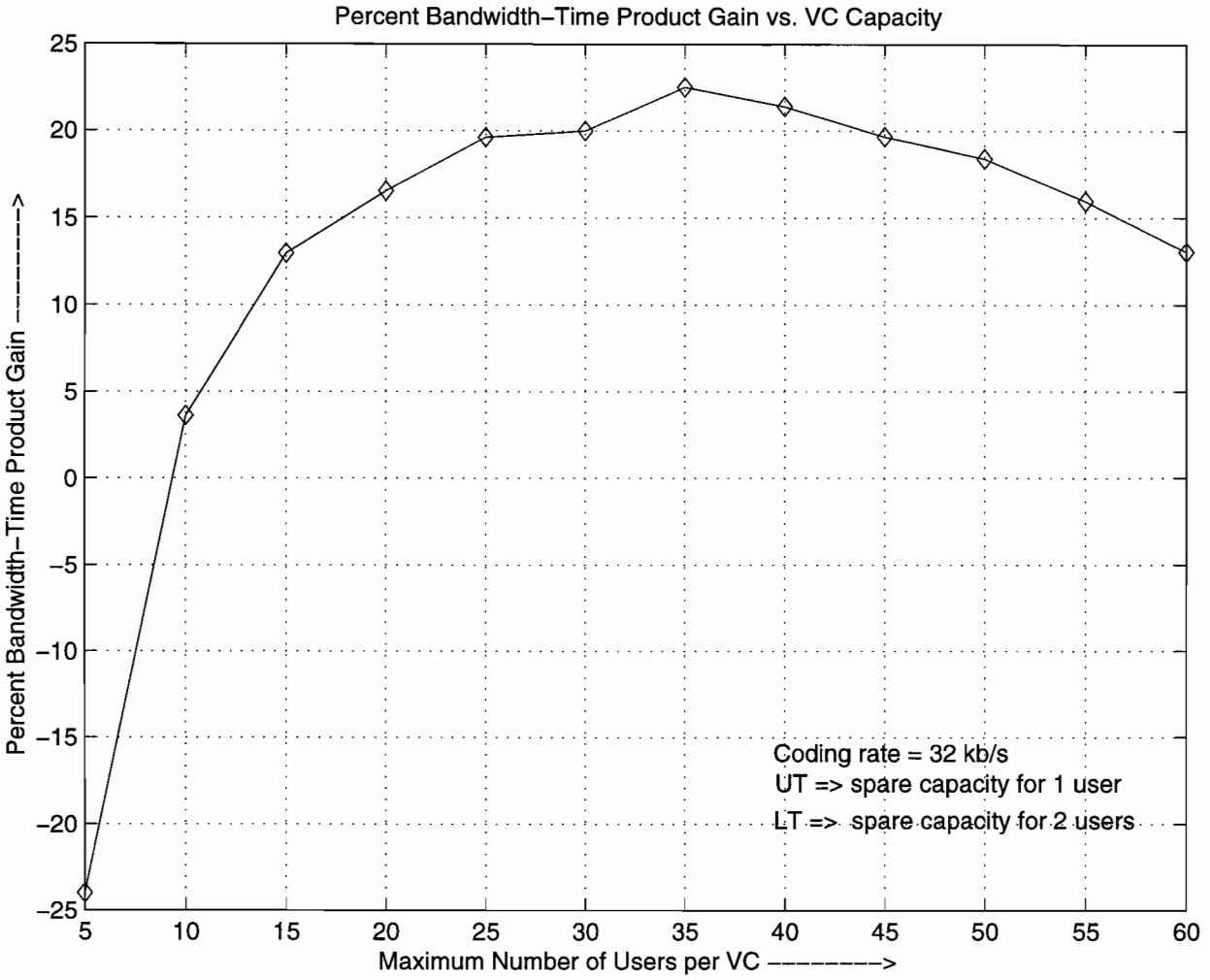


Figure 6.1: Bandwidth-Time product gain for different VC capacities for homogeneous users

upper threshold. A VC size of 30 users per VC was used to generate the plot. The lower threshold value is set such that the difference in bandwidth between upper threshold and lower threshold is sufficient to accomodate 2 users.

Call Rejection Probability vs. Upper Threshold (see Figure 6.3): Figure 6.3 shows how the call rejection probability increases with the value of upper threshold. The probability that a VC request is rejected is taken as 0.40. That is, 40% of the requests made by the AAL2 node to the network are assumed to be rejected. This is an over simplified assumption for the network rejection probability but has been used due to the lack of an established model. Each VC is assumed to support 15 users for generating the plot.

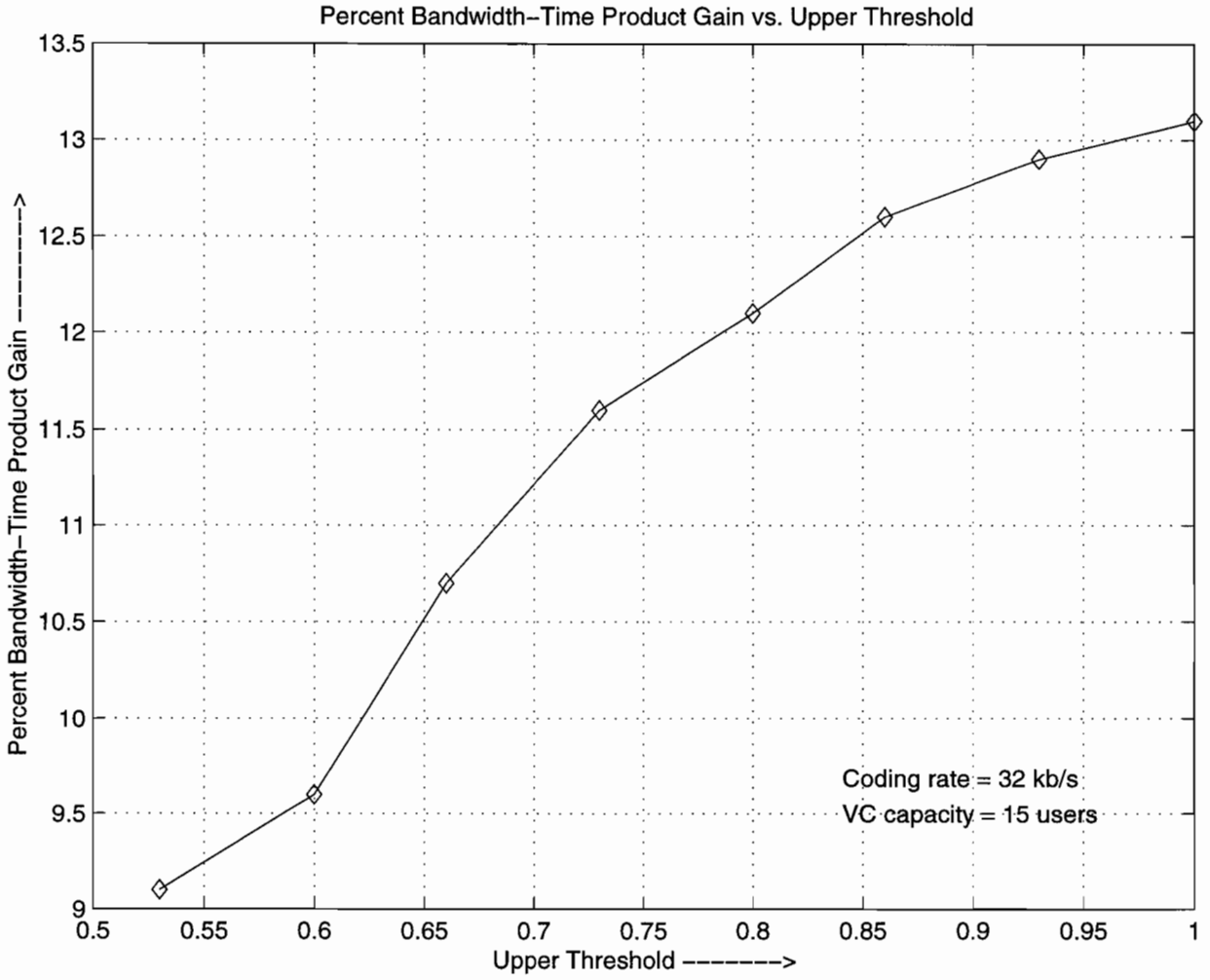


Figure 6.2: Bandwidth-Time product gain for different Upper Threshold values

VC Rejection Probability vs. Call Rejection Probability (see Figure 6.4): Figure 6.4 gives the variation of call rejection probability with the VC rejection probability. Notice that the call rejection probability is very much less than the VC rejection probability. With VC rejection probability as high as 80%, the call rejection probability is only 0.02. The VC capacity is assumed to be 15 users per VC.

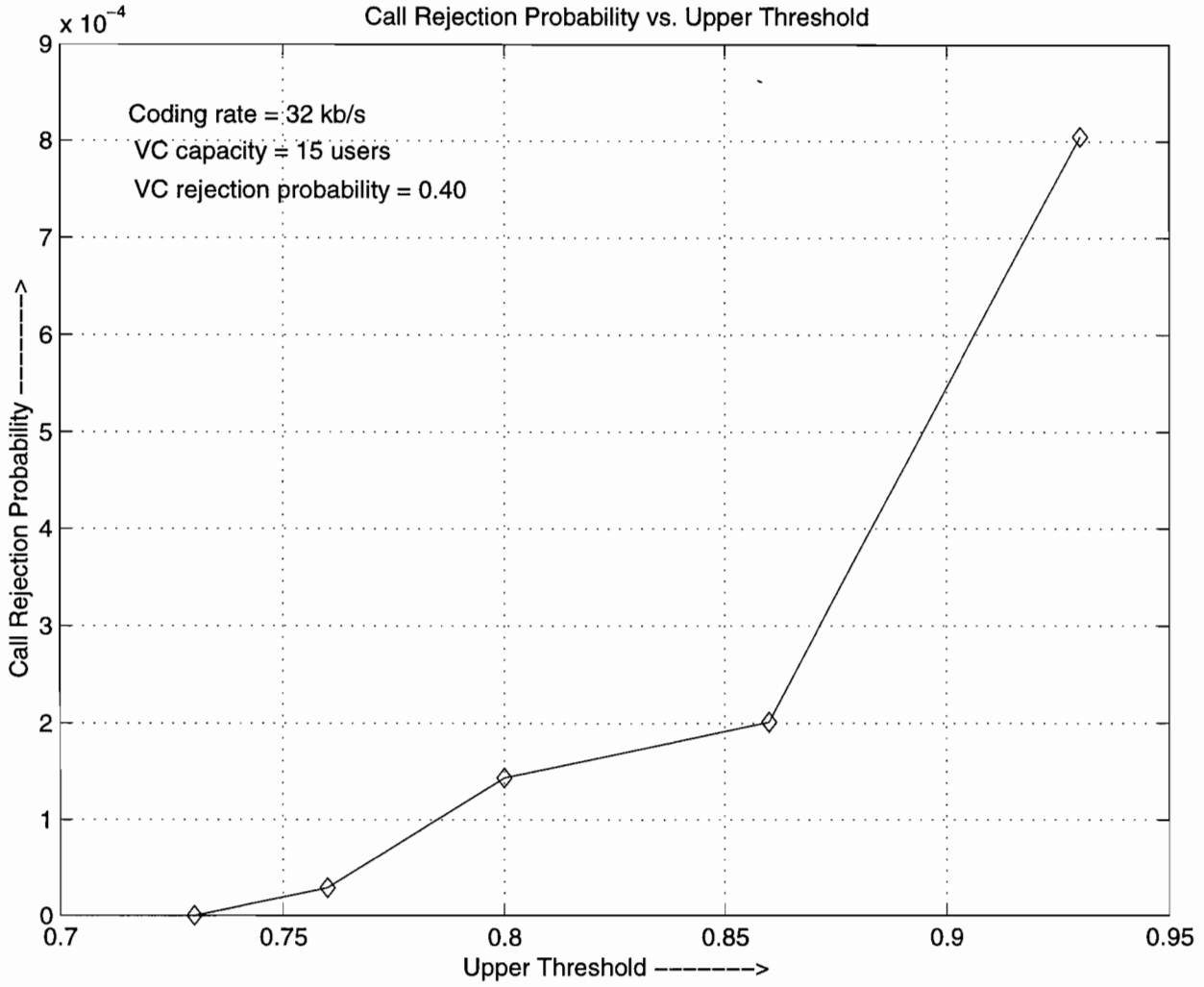


Figure 6.3: Call Rejection Probability for different Upper Threshold values

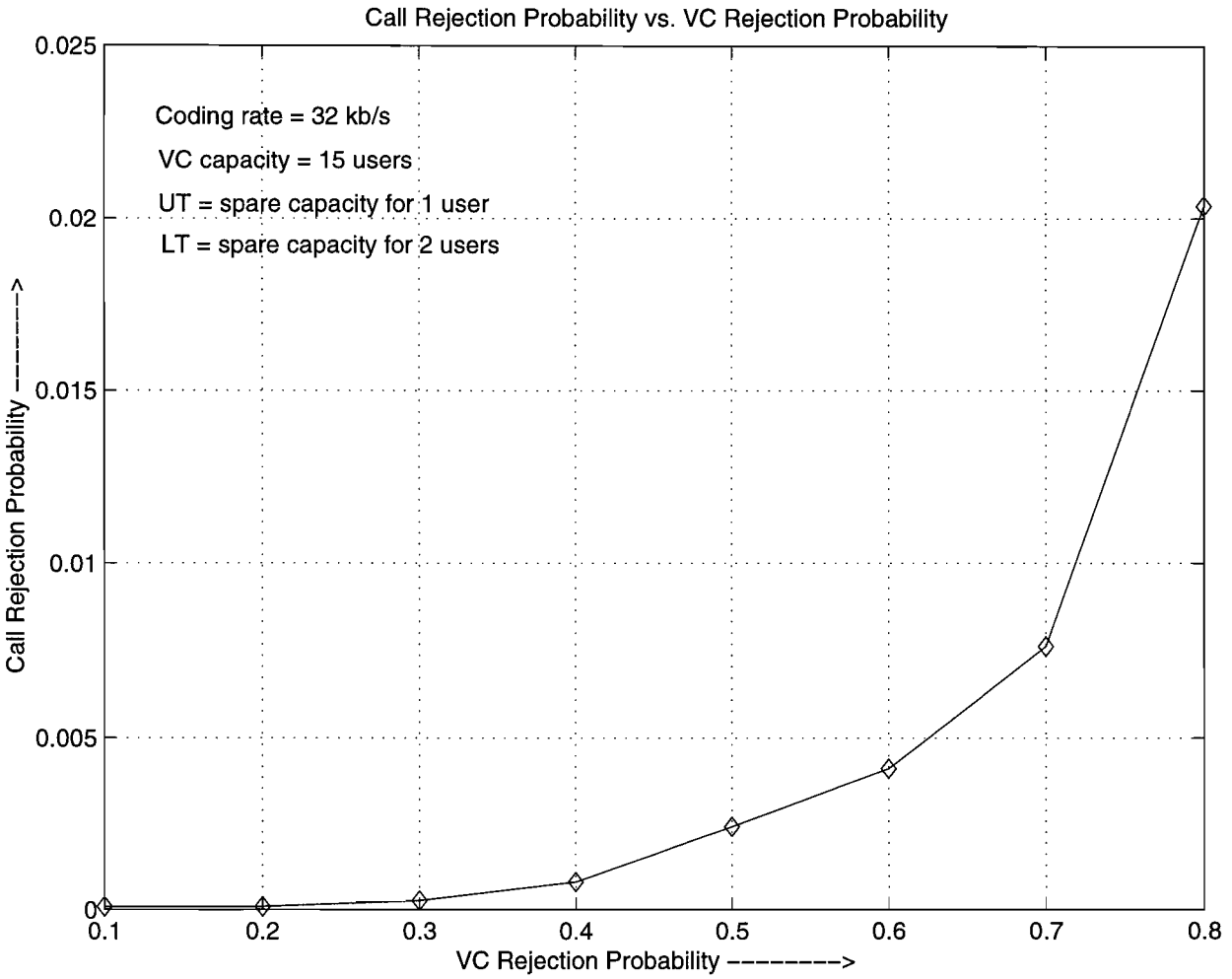


Figure 6.4: Call Rejection Probability for different VC rejection probabilities

6.3.2 Heterogeneous User Case

The simulation setup for the heterogeneous user case is similar to the homogeneous user case but the setting of threshold values and VC capacities is more complicated. In the homogeneous user case we set the capacity of the VCs in terms of the number of users. This is possible because we can precompute the bandwidth requirements for a given number of homogeneous users. The VC capacity should be set such that it just supports an integer number of users. Any extra bandwidth in the VC that is not enough to support an additional user is a waste of resource. But in the heterogeneous user case, a VC is occupied by a mixture of different types of users and hence it is difficult to decide on a VC size that would just support an integral number of users. The setting of upper threshold is also more involved for the same reason.

Bandwidth-Time Product Gain vs. VC Capacity: Figure 6.5 shows how the bandwidth-time product gain varies with the VC capacity. The VC capacity shown here is in terms of bandwidth. As pointed out earlier, setting the VC capacity to support a fixed number of users is not appropriate in a heterogeneous user case. The upper threshold value is set such that there is sufficient bandwidth for one user of the highest coding rate (64 kb/s). The lower threshold value is set such that there is sufficient bandwidth for two users of the highest coding rate (64 kb/s). The bandwidth-time product gains are shown with respect to a case where a single large PVC is used to support all the users and a case where three different PVCs are used to support the three different kinds of users. The peak load in the simulation was 198 users (62, 75 and 61 users using 16 kb/s, 32 kb/s and 64 kb/s coding rate respectively). Using NEC CAC, the single larger PVC required a bandwidth of 4991 kb/s to support the peak load. The three PVCs supporting 16 kb/s, 32 kb/s, 64 kb/s calls were assumed to be of capacities 756 kb/s, 1794 kb/s, 2982 kb/s respectively. From the curve we see that using SVCs of size 800 kb/s each will prove to be most efficient in the given scenario.

Call Rejection Probability vs. VC Rejection Probability: Figures 6.6, 6.7 and 6.8 show the relation between the call rejection probability and VC rejection probability. The model for the VC rejection probability is the same as the model used for homogeneous users. We observe that the call rejection probability increases with the coding rate. This is because of higher *effective VC capacities* (in terms of capacity for number of users) and lower *effective threshold values* for lower coding rates.

In this chapter, the gains associated with the proposed CAC algorithm are demonstrated for the given load variation statistics. The load variation curve assumed is conservative and the real life load variation statistics are expected to have more variation. The CAC algorithm proved efficient even for the conservative case studied.

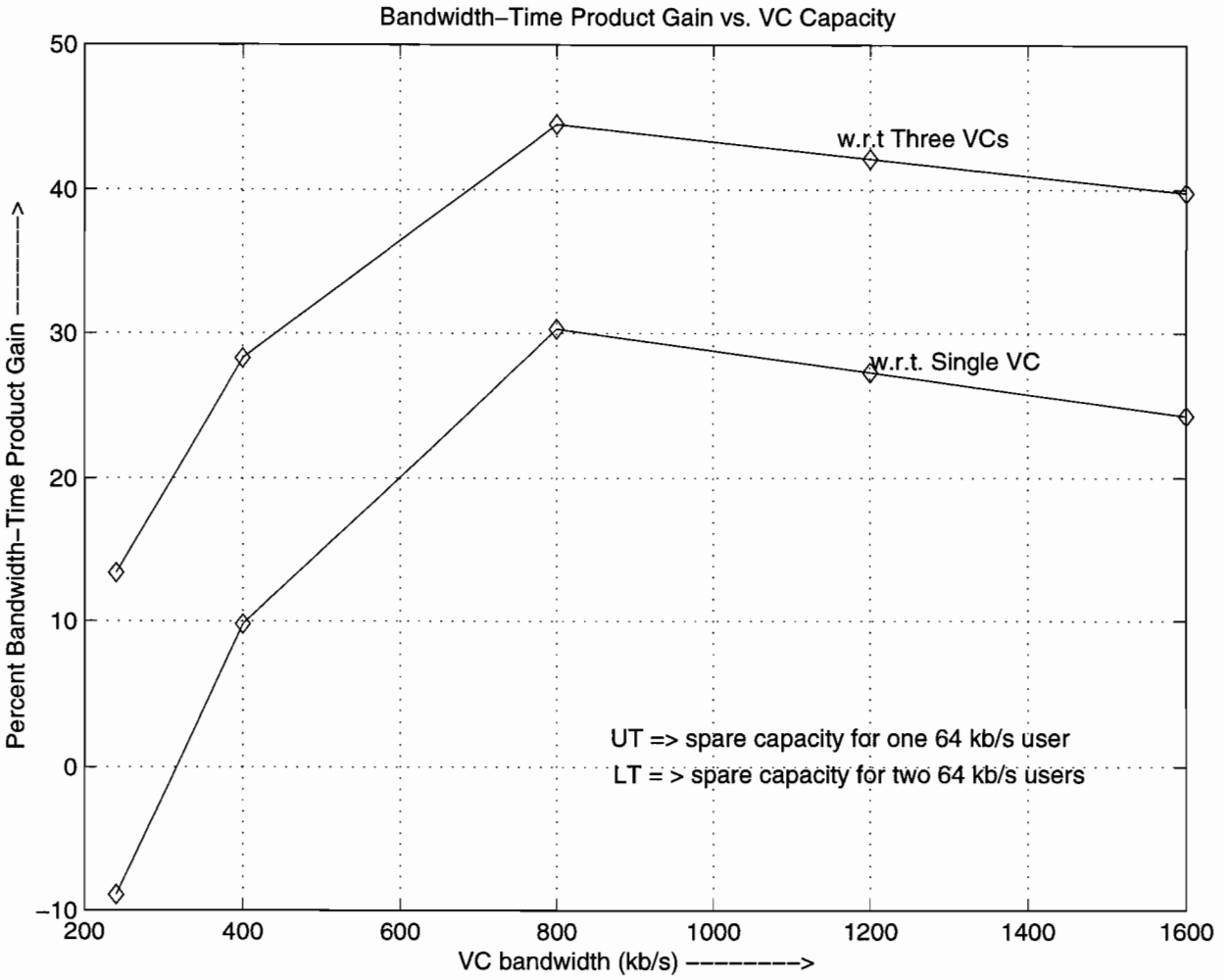


Figure 6.5: Bandwidth-Time Product gain for heterogeneous users

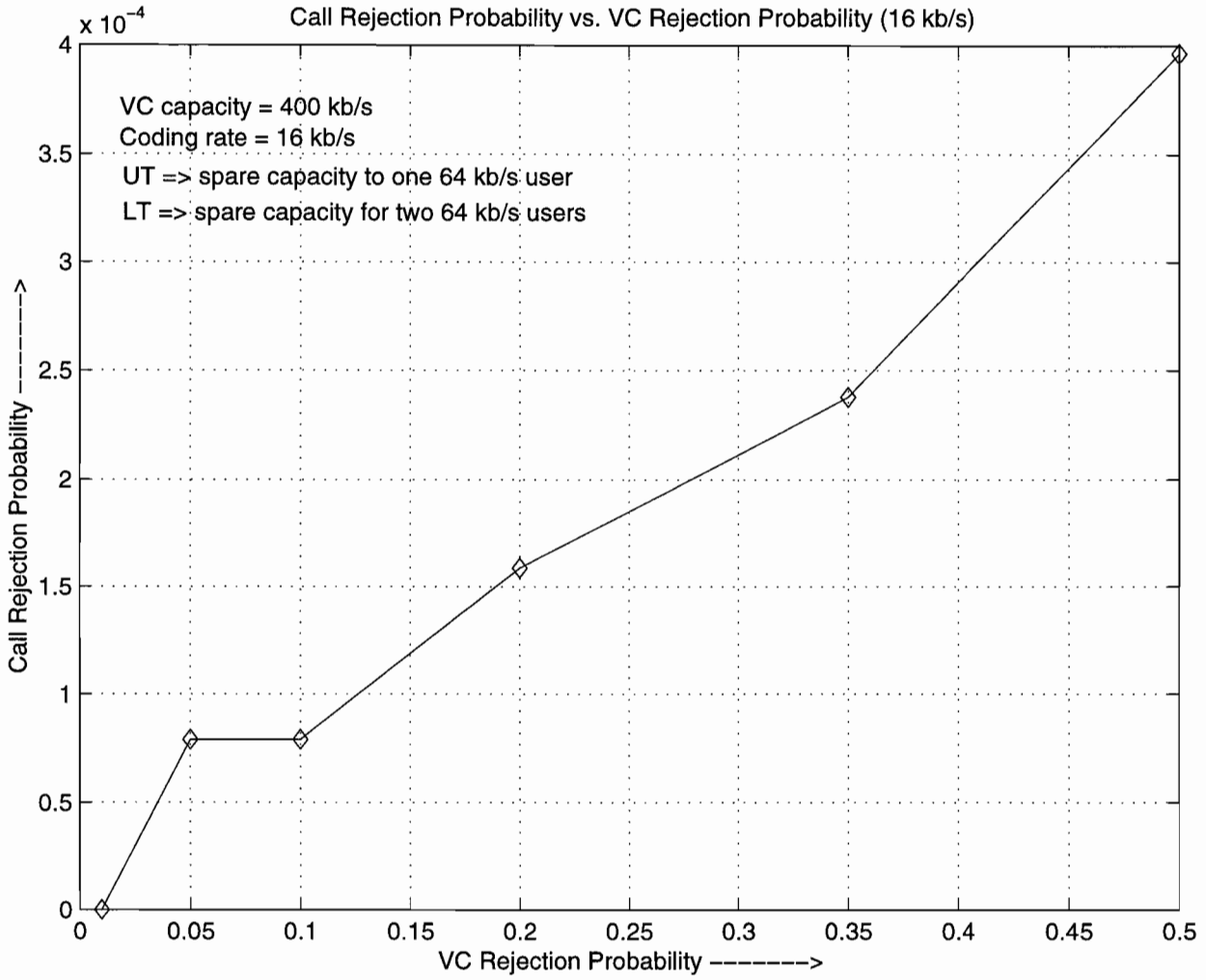


Figure 6.6: Call Rejection Probability for different VC rejection probabilities for users of 16 kb/s coder

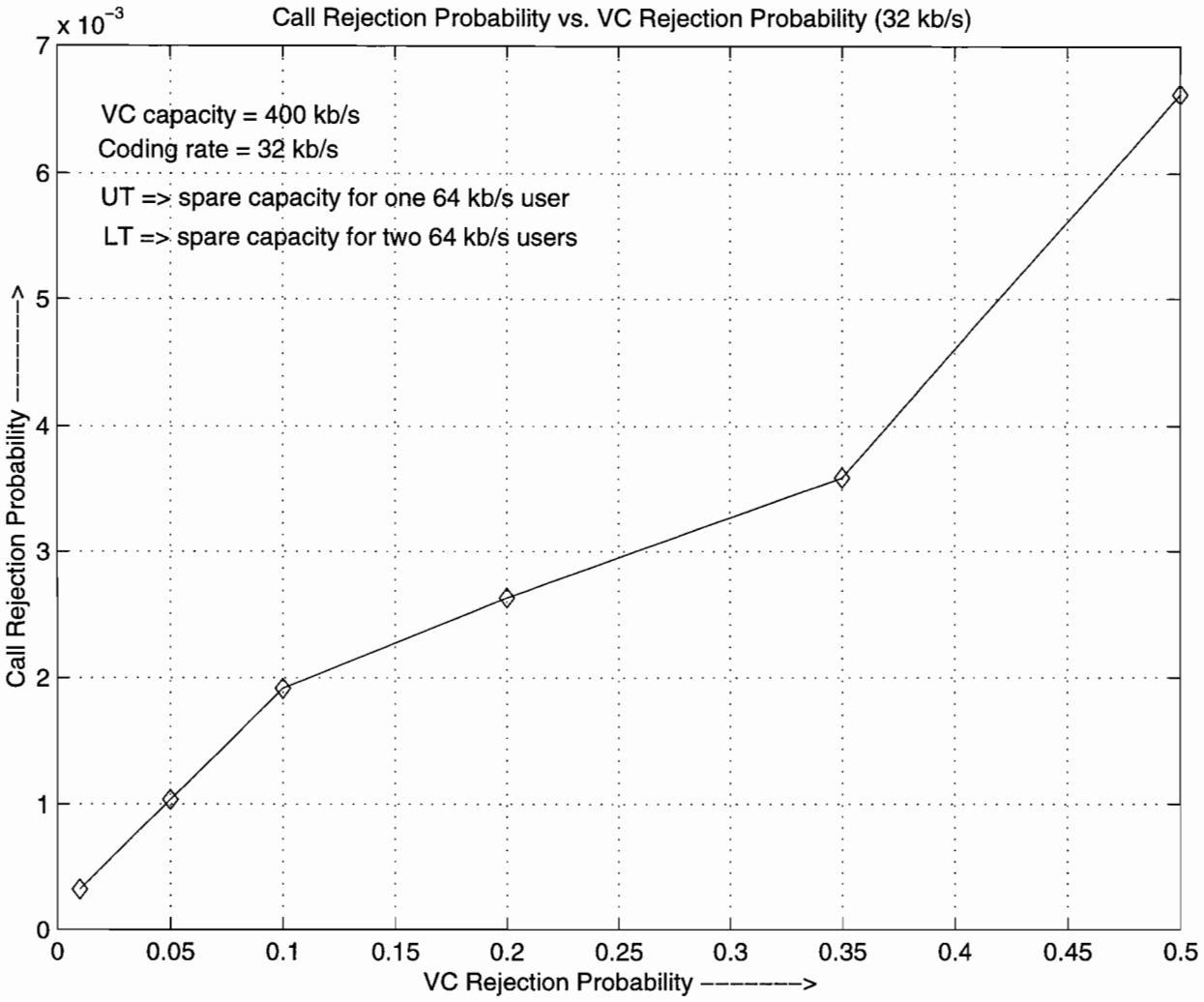


Figure 6.7: Call Rejection Probability for different VC rejection probabilities for users of 32 kb/s coder

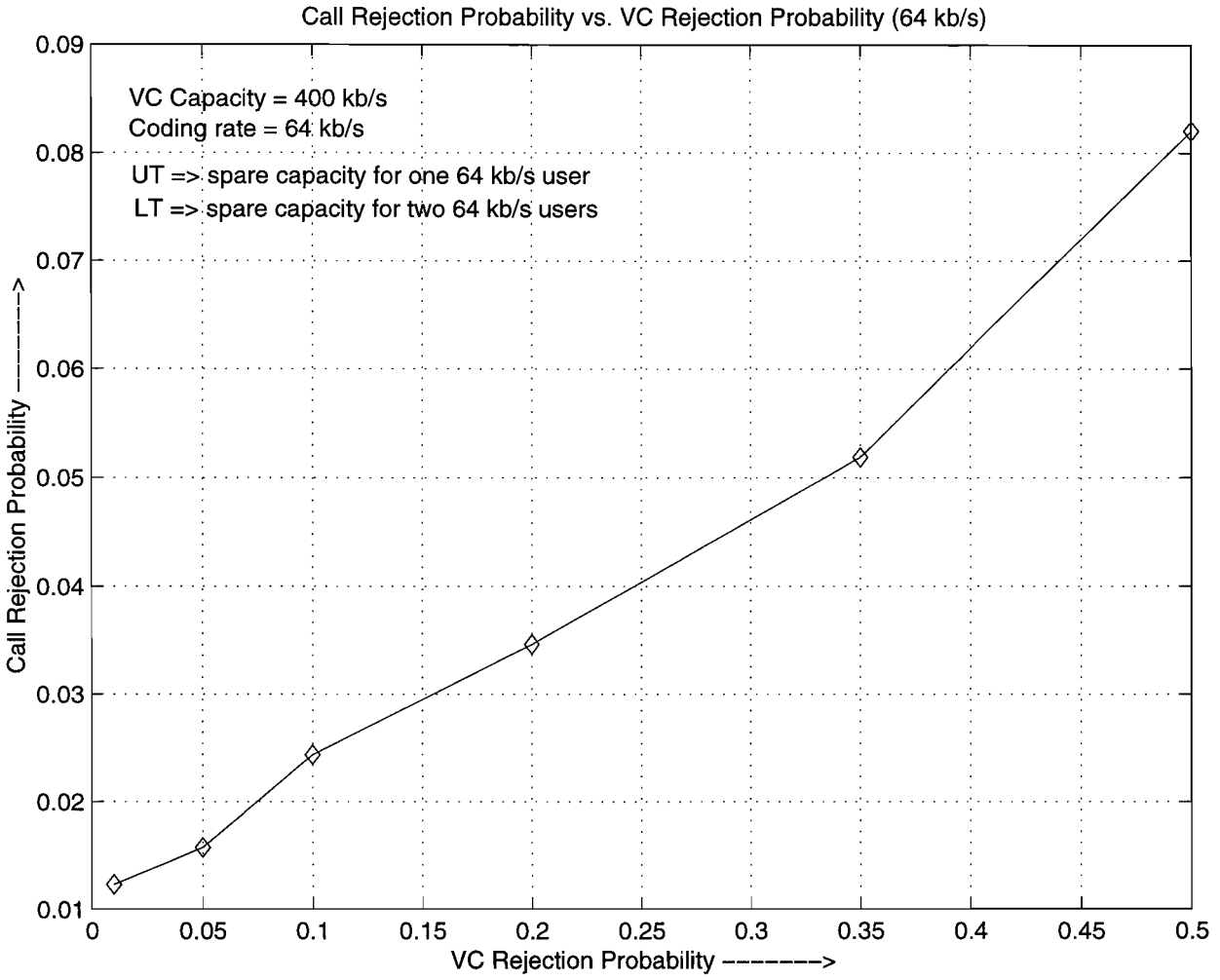


Figure 6.8: Call Rejection Probability for different VC rejection probabilities for users of 64 kb/s coder

Chapter 7

Conclusions and Future work

7.1 Summary of Contributions

In this report, a generic AAL2 CAC algorithm (SCALE) is proposed. The need for a mature CAC strategy at the AAL level when using AAL2 is identified. The concept of using multiple SVCs instead of a single PVC is developed. The proposed CAC algorithm takes care of setting up SVCs in anticipation of load. Effectiveness of the proposed algorithm is studied by simulating the CAC procedure for a hypothetical load variation curve. Optimum parameters for the CAC are found for the given load variation curve. Virtual Buffer Measurement Mechanism, a method for estimating VBR UPC parameters in simulation and live traffic environments is developed and verified. This method is used to estimate VBR UPC parameters for AAL2 multiplexed voice traffic.

7.2 Conclusions

It has been found that the proposed CAC algorithm is very effective when there is atleast a moderate load variation. The optimum VC size used is found to be a compromise between higher multiplexing gain and low utilization associated with larger VC sizes and lower multiplexing gain and higher utilization associated with smaller VC sizes. The concept of using an *upper threshold* to setup an additional VC proved to be effective in reducing the call blocking probability. Simulation results show that call rejection probabilities are very less even at high VC rejection probabilities when using the proposed algorithm. Results also show that call rejection probabilities are higher for users of higher coding rates.

7.3 Future Work

- Clearly, this report lacks analytical results on the proposed CAC algorithm. It would greatly help if results from theoretical analysis can be used instead of simulation for estimating the optimal parameters to be used for a given load variation curve. This could be more complex if the load variation does not follow a well defined statistical distribution.

- We used a constant VC size to reduce the complexity of the CAC algorithm. By using a variable VC size and deciding the optimum size through intelligent means based on factors like immediate past load and rate of increase of load etc, will prove to be more efficient.

- As stated earlier, optimal CAC parameters have been found for a hypothetical load variation statistics. The CAC algorithm has to be simulated for the service providers own load variation curve to find the relevant optimum parameters. Also, necessary modifications as suggested have to be done if the service provider's AAL2 system provides different QoS for AAL2 users and/or uses dynamic source code control.

- The model used for VC rejection probability is unrealistic and has been used because of a lack of alternate model. Repeating the simulations for finding call blocking probabilities under a better model for VC rejection probability will be useful.

- We assumed that the price for service to be proportional to the bandwidth-time product. But in reality the SVC setup costs are also to be taken care of. The pricing policy [23] should also be considered when deciding on the optimal VC sizes. It would be interesting to see the results when pricing constraints are also applied.

Bibliography

[1] *B-ISDN ATM Adaptation layer specification: Type 2 AAL*, ITU-T Recommendation I.363.2, September 1997.

[2] *Traffic Management Specification*, The ATM Forum Technical Committee, Version 4.0, af-tm-0056.000, April 1996.

[3] John H. Baldwin, Behram H. Bharucha, Bharat T. Doshi, Subrahmanyam Dravida and Sanjiv Nanda, *AAL2-A New ATM Adaptation Layer for Small Packet Encapsulation and Multiplexing*, Bell Labs Technical Journal, Spring 1997.

[4] D.W. Petr, *Lecture Notes EECS 963: Integrated Telecommunication Networks*, University of Kansas, Fall 1999.

[5] Gopi Vaddi, D.W. Petr, *AAL2 UPC Parameter Study*, ITTC-FY99-TR-15664-01, February 1999.

[6] Gopi Vaddi, D.W. Petr, *Voice Activity Statistics*, ITTC-TR-FY2000-15664-03, August 1999.

[7] Raghushankar R. Vatte, D.W. Petr, *Performance comparison between AAL1, AAL2 and AAL5*, ITTC-FY98-TR-13110-03, June 1998.

[8] Raghushankar R. Vatte, *Supporting Multiple Traffic Classes on ATM Adaptation Layer Type 2 using Importance Queuing*, Masters Thesis, University of Kansas, 1998.

[9] *Packet-based multimedia communications systems*, ITU-T Recommendation H.323, February 1998.

[10] G.Ramamurthy, Qiang Ren, *Multi-Class Connection Admission Control Policy for High Speed ATM Switches*, IEEE INFOCOM, 1997.

[11] Anick, Mitra, Sondhi, *Stochastic Theory of a Data Handling System With Multiple Sources*, Bell System Technical Jounal, Vol.61, 1982.

[12] Roch Guerin, Hamid Ahmadi, Mahmoud Naghshineh, *Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks*, IEEE JSAC, Vol.9, NO. 7, September 1991.

[13] H.G. Perros, K.M. Elsayed, *Call Admission Control Schemes*, IEEE JSAC, November 1996.

[14] Raghushankar R. Vatte, Prema Sampath, D.W. Petr, *AAL2 Transmitter Simulation Study: Revised*, ITTC-FY98-TR-13110-01, March 1998.

[15] Aarti Iyengar, *Implementation and Performance Analysis of a Preliminary AAL2 CAC Scheme*, Masters Thesis, University of Kansas, September 1999.

[16] Vishal Moondhra, *Implementation and Performance Analysis of ATM Adaptation Layer Type 2*, Masters Thesis, University of Kansas, 1998.

[17] Aarti Iyengar, Dhananjaya Rao, Joseph B. Evans, *Implementation of ATM Adaptation Layer 2*, Technical Report ITTC-FY2000-TR-15662-01, Information and Telecommunications Technology Centre, University of Kansas, July 1999.

[18] Sampath Sreepathi, *Dynamic Source-Coding Rate Control Scheme for ATM Adaptation Layer Type 2*, Master's Thesis, University of Kansas, 1997.

[19] L.A. DaSilva, D.W. Petr, N. Akar, *Equilibrium Pricing in Multiservice Priority-Based Networks*, Proceedings of the IEEE Globecom Conference, pp. S38_6.1 - S38_6.5, November 1997.

[20] Kunyan Liu, D.W. Petr, Cameron Braun, *A Measurement-Based CAC Strategy for ATM Networks*, ICC 1997, pp 1714-1718.

[21] Fabrice Guillemin, Catherine Rosenberg, Josee Mignault, *On Characterizing an ATM Source via the Sustainable Cell Rate Traffic Descriptor*, Infocom 1995, pp 1129-1136.

[22] Paul T. Brady, *A model for generating ON-OFF speech patterns in two-way conversations*, Bell System Technical Journal, Vol. 48, pp. 2445-2472, September 1969.

[23] Yuhong Liu, *The Influence of Pricing on PVC vs. SVC Service Preferences*, ITTC Technical Report ITTC-FY2000-TR-12960-03, July 1999.

[24] *BONeS Designer: Modeling Reference Guide*, Comdisco Systems Inc., April 1992.

[25] *Extend User's Manual*,Version 4, Imagine That Inc., 1997.

[26] Raif O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues*, Artech House, 1995.