UNIVERSITY OF ALBERTA

PERFORMANCE STUDY OF A CALL ADMISSON CONTROL SCHEME

FOR B-ISDN

BY

© HONG QIAN

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE**.

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

FALL 1994

ISBN 0-315-95097-8

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: **HONG QIAN**

TITLE OF THESIS: **PERFORMANCE STUDY OF A CALL ADMISSION CONTROL SCHEME FOR B-ISDN**

DEGREE: **MASTER OF SCIENCE**

YEAR THIS DEGREE GRANTED: **FALL 1994**

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

#403, 9730-106 Street
Edmonton, Alberta
T5K 1B7
Canada

Date: June 28, 1994

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommended to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **PERFORMANCE STUDY OF A CALL ADMISSION CONTROL SCHEME FOR B-ISDN** submitted by **HONG QIAN** in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** in Computing Science.

J. Harms

Dr. Janelle Harms, supervisor

Dr. Anup Basu, internal

W. Grover

Dr. Wayne Grover, external

Date: June 28, 1994

Abstract

Asynchronous transfer mode (ATM) is recommended as the network architecture of B-ISDN by CCITT. Call admission control (CAC) becomes a challenging problem because statistical multiplexing is used in ATM networks. In CAC, users must give traffic characteristics of their calls. However, users are not necessarily experts. They may not be able to provide the accurate information. Even if they can, the control may not be reliable. A new CAC scheme which requires minimum information from users is proposed in this thesis. It only requires users to choose a class (e.g. data, voice and video) for their calls. It uses the measured bit rates of the links to assist in making call admission decisions. It also tries to capture the burstiness of the traffic in ATM networks by tracking the bit rate of each call. The performance of the new CAC is compared with other two CAC schemes which require users to specify the peak rate of their calls. The simulation results show that the new CAC scheme achieves much better performance.

Table of Contents

List of Tables

Table Page

List of Figures

Figure Page

Figure Page

List of Abbreviations

AR	AutoRegressive (process)
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband ISDN
CAC	Call Admission Control
CBP	Call Blocking Probability
CBR	Constant Bit Rate
CCITT	Comité Consultatif International Télégraphique et Téléphonique (International Telegraph and Telephone Consultative Committee)
CIT	Call Interarrival Time
CRN	Common Random Numbers
DSC	Distributed Source Control
EEC	End to End Control
FCFS	First Come First Serve
IID	Independent identically distributed
IPP	Interrupted Poisson Process
ISDN	Integrated Services Digital Network
LLC	Link Level Control
MMPP	Markov Modulated Poisson Process
MPEG	Motion Picture Expert Group
N-ISDN	Narrowband ISDN
PAD	Packet Assembly and Disassembly
PCM	Pulse Code Modulation
QOS	Quality of Service
SONET	Synchronous Optical Network
STM	Synchronous Transfer Mode
STS-3	Synchronous Transport Signal level 3
TBP	Traffic Blocking Probability
TES	Transform Expand Sample
TR*Labs*	Telecommunications Research Laboratories
UNI	User Network Interface
VBR	Variable Bit Rate

VRT	Variance Reduction Technique

CHAPTER 1 Overview

1.1 Introduction

Broadband Integrated Services Digital Network (B-ISDN) has received much attention recently because of the increased demand for integrated telecommunication services (e.g. voice, data, and video). A good B-ISDN system is capable of supporting a wide variety of traffic and diverse services while maintaining Quality Of Service (QOS) of all calls in the systems. The QOS of a call may include performance requirements of loss and delay.

Asynchronous Transfer Mode (ATM) is considered to be the most promising among a number of broadband network architectures including Synchronous Transfer Mode (STM). ATM attempts to transport all services with the same format so that switches do not have to be aware of the services being transported. Information is carried in cells which are fixed size packets. According to CCITT recommendation, each cell is 53 bytes long with 5 bytes of header and 48 bytes of information.

ATM networks have advantages such as efficiency and flexibility. Because bandwidth is allocated to services according to their traffic characteristics and service requirements, ATM is able to handle the different services and traffic in B-ISDN. Bandwidth saving is gained by the statistical multiplexing of bursty traffic sources because bursty sources do not need peak bandwidth all the time. However, congestion can result if many bursty sources become active simultaneously. Therefore, congestion control is needed.

There are two types of congestion control: reactive control and preventive control. Reactive control which takes control action upon receiving feedback from the congested node may not be effective in ATM networks due to the high transmission speed. Consider the following scenario [9]. Suppose two nodes, *A* and *B* are linked by a 100-km bi-directional cable. For simplicity, assume the cell length is 500 bits (the CCITT recommended cell length is 424 bits). Also assume a propagation delay of 5 μs per 1 km of a cable. We can obtain the following results.

- For the channel speed of 1Mbits/s, the transmission time for one cell is (500 bits)/(1 Mbits/s) = 0.5 ms. Since the propagation delay is 500μs, node *A* can transmit one cell during this delay.
- For the optical fiber cable of 1 Gbits/s speed, the transmission time for one cell is (500 bits)/(1 Gbits) = 0.5 μs. Node *A* can transmit 1000 cells during the propagation delay.

With reactive control, node *B* must send feedback to node *A* when congestion occurs at node *B*. Then node *A* will reduce its transmission rate. If the low speed channel is used, only one cell has been transmitted into the channel by node *A* when it receives the feedback. Node *A* will reduce its transmission rate immediately and this one cell will not cause congestion at node *B*. However, if the optical fiber is used, 1000 cells have been transmitted into the channel by node *A* when it receives the feedback. These cells in the channel will cause further congestion at node *B*. Therefore, it is too late for node *A* to reduce its transmission rate because cell loss may have already occurred.

Preventive control is a good choice. It tries to prevent the network from becoming congested. Preventive control usually controls the traffic flow at the network entry point. Call Admission Control (CAC) and bandwidth enforcement (or policing) are two types of preventive control. The CAC determines whether to accept or reject a new call. The policing mechanism controls the bandwidth usage of the accepted call and punishes the misbehaved user by dropping or delaying cells.

The topic of this thesis is CAC. An optimal CAC must maximize the utilization of network resources and accept the maximum number of calls while maintaining the QOS of all accepted calls. The QOS could be cell loss and delay requirements. The network resources include link bandwidth, buffer size and processor power. The other requirement of a good CAC is that it should require minimum information from users since they may not be able to provide accurate technical information on their calls [43]. Even if they can, the descriptors may not be sufficient to estimate a call's resource requirements [3]. This is a challenging problem. A good CAC scheme has not been found yet. Please see Chapter 3 for details of CAC.

Since various protocols and applications must be studied before the network is built, researchers have used models of traffic sources and networks for studying network performance. The network model must describe the topology and protocol of a network. The source model is a mathematical model which represents the characteristics (e.g. mean cell rate and burstiness) of the source traffic. There are many source models because ATM networks must provide integrated telecommunications services producing video, voice and data traffic. With a network model and using source models, the network performance can be evaluated by simulation or queueing analysis. Thus accurate source models are required for studying the ATM networks. We discuss source models in Chapter 2.

1.2 Objective of the Thesis

The objective of this thesis is to design a new CAC scheme called Refinement CAC and study its performance. Refinement CAC requires minimum information from the users. Users are only required to choose a class (e.g. data, voice and video) for their calls. The scheme uses the measured bit rates of the links to assist in making CAC decisions. It also further classifies calls by tracking the bit rate of each call.

Refinement CAC proposed in this thesis is compared with two other CAC schemes which require peak rate from the users. The first peak rate allocation scheme requires the user to specify the peak rate requirement of his/her call. Then, it makes call admission decisions based on whether it has enough free bandwidth to allocate the peak rate to the call. The second scheme uses measurement to adjust the free bandwidth for call admission. These CAC schemes will be described in more details later.

Simulation is used to study the performance of the CAC schemes. The C programming language is used to build the simulation systems. A network model and several source models are built. The workstations at TR*Labs* and University of Alberta were used to run the simulations.

1.3 Organization of the Thesis

First of all, abbreviations are listed after Table Of Contents. The organization of the chapters is as follows.

In Chapter 2, the source models proposed in literature are revieweJ. The new CAC proposed in this thesis is partly motivated by the study of source models.

In Chapter 3, the CAC schemes proposed recently in literature are presented. The study of these CAC proposals helps us to design the new CAC.

In Chapter 4, the architecture and desigp of the new CAC are presented. This chapter explains the details of the new CAC design.

In Chapter 5, the simulation details and validation are described.

In Chapter 6, the values of the parameters for the simulations are described.

In Chapter 7, the simulation results and analysis are presented.

In Chapter 8, a summary of the entire research project is presented. The performance of the new CAC is concluded and suggestions of future research are made.

CHAPTER 2 Survey of Source Models in B-ISDN

2.1 Introduction

Since various protocols and applications must be studied before the network is built, researchers have used models of network and traffic sources for studying network performance. The network model must describe the topology and protocol of a network. The source model is a mathematical model which represents the characteristics (e.g. mean cell rate and burstiness) of the source traffic. With network model and source models, the network performance can be evaluated by simulation or queueing analysis.

ATM is capable of handling the different types of traffic in B-ISDN since bandwidth is allocated to calls according to their traffic characteristics and service requirements. Therefore, the ATM network is very flexible. Traffic can be divided into two categories: constant bit rate (CBR) and variable bit rate (VBR). CBR sources transmit cells at a constant bit rate. For example, a voice source with common PCM coding generates CBR traffic of 64 kbps. VBR sources have periods during which the bit rate is lower than peak rate or equal to zero. The typical VBR sources are bursty sources which transmit cells in bursts and have silent durations between bursts. We will see in the following sections that many traffic sources are VBR sources. For example, a voice source with silent period suppression generates VBR traffic. With VBR sources, there is the possibility of using statistical multiplexing in the ATM networks which results in increasing network utilization [41]. Therefore, the ATM network has the potential of being efficient. We concentrate on VBR sources in this study because most sources in ATM networks are VBR sources.

As introduced in Section 1.1, source models are used to study the network performance by simulation or queueing analysis. Source modelling in ATM networks is a particularly challenging problem because the networks must carry many different types of traffic. These traffic sources are usually statistically multiplexed during transmission. Accurate source models are required for studying the performance of the ATM networks.

This chapter surveys a number of source models proposed in literature for data, voice and video sources. In some cases, the authors of the papers have validated their models by comparing them to real sources or simulation results. We will note that where the validation is done.

2.2 Data Source Models

This section presents the data source models proposed in literature. The data applications include file transfer, email, etc. and are usually VBR sources. Both single and aggregate data sources are discussed. An aggregate source is the traffic source with several single sources multiplexed together. Some data models which are also suitable for voice or video may be introduced in subsequent sections.

In modeling single data sources of existing packet networks, the Poisson arrival process (for the continuous time case) or Geometric interarrival process (for the discrete time case) has long been used to model the data sources [9]. However, the cell size is fixed in ATM networks, while existing packet networks allow packets with either variable or constant length. This means that multiple cells may be required for one data packet and the source model should represent this.

One general method of modelling aggregate data sources is the Markov modulated fluid model [46, 47]. The bit rate of the source is controlled by an m-state Markov chain where the state represents the current bit rate. The bit rate of the source changes when a transition is made to a new state and cells are transmitted with a constant bit rate within a state. For a two-state fluid model ($m = 2$), the source alternates between two bit rates. Four parameters are needed: the bit rate and mean duration of each state. The duration of each state is generated by an exponential distribution. If the bit rate in one of the two states is zero, this model becomes an On-Off Fluid model [8, 30]. The On-Off Fluid model is also suitable for voice and video sources. Other models which are suitable for data and other types of traffic will be discussed in subsequent sections.

2.3 Packet Voice Source Models

Models proposed for single and aggregate voice sources are surveyed in this section. A voice source with common PCM coding generates CBR (constant bit rate) traffic with rate of 64 kbits/s. The models described in this section are for packetized voice sources which are coded and compressed, and therefore generate VBR (variable bit rate) traffic. The cell arrival process is highly correlated, making it complicated to model accurately.

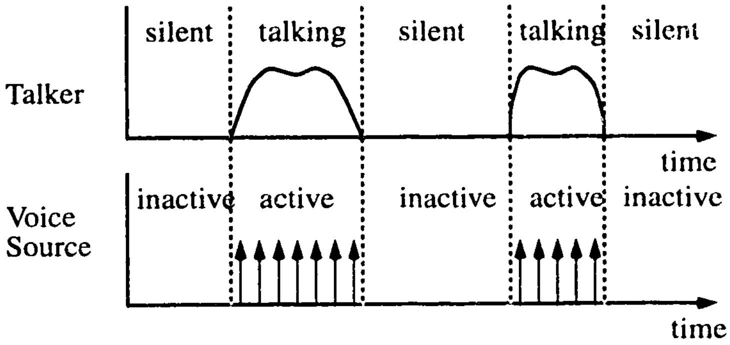


Figure 2.1: Typical voice source behavior

Figure 2.1 illustrates the typical behavior of a voice source [38]. The packet voice source in a telephone call could be modeled as active periods when the caller speaks interrupted by silent periods. So, the voice source generates cells during active periods, and generates no cells during silent periods. With silence detection techniques, it has been shown that the active talkspurt duration for voice is only in the range of 35-40 percent [10]. The mean durations of active and idle states have been measured on real voice traffic and shown to be 352 ms and 650 ms, respectively [31, 8, 11]. ATM networks can support such bursty sources with reduced bandwidth by using statistical multiplexing because the sources do not need bandwidth during silent periods. The VBR voice sources are becoming popular because of this bandwidth saving.

The Interrupted Poisson Process (IPP) can be used to model a single voice source [38]. The IPP traffic model is a Markov chain consisting of two states: an active state and an idle state. Cells are generated according to a Poisson process during the active state and no cells are generated during the idle state. Figure 2.2 [37] shows the transitions between the two states. The transition rate from active to idle is β; while the transition rate from idle to active is α. The durations of both the active and idle states are exponentially distributed.

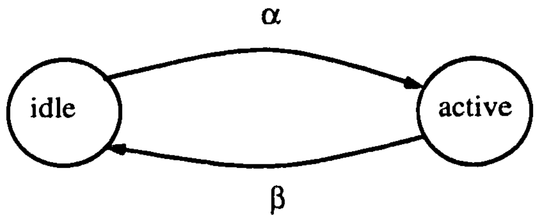


Figure 2.2: IPP Model

One variation of the IPP model is the On-Off fluid model. It is introduced in this section because of its popularity [10, 11, 38, 28, 31]. It is used for single voice, single VBR coded video [33] and data sources. In this model, the cells are generated by a two-state Markov chain. The first state is the OFF state during which no cells are generated and the second state is ON state during which cells are generated with constant rate. The duration of each state follows an exponential distribution. This model is simpler than the IPP model and therefore more analytically tractable because constant bit rate is used during the ON state. Another variation of the IPP model uses Bernoulli process instead of Poisson process for cell arrivals during active periods [37].

Rathgeb [35] describes a 2-phase burst/silent model which is also similar to the IPP model. He believes this model is a fairly realistic model for packetized voice, still picture and interactive data sources. It has three parameters: the number of cells per burst (geometric distribution), the cell interarrival time in burst (constant), and the silence duration (exponential distribution). Figure 2.3 is a diagram of this model. The three parameters of the model can be determined by the following traffic descriptors: maximum cell rate, mean cell rate and mean burst duration. This model has been used for analytical study as well as simulation to compare several policing mechanisms.

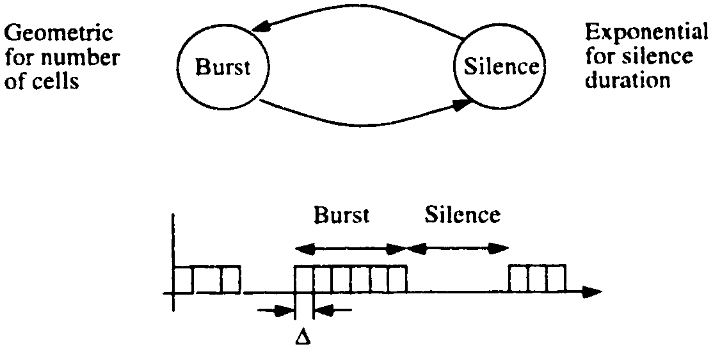


Figure 2.3: Two-phase burst/silence source model

The Markov Modulated Poisson Process (MMPP) is used to model aggregate voice and data sources [11]. It is a Poisson process whose rate is determined by a Markov chain. An aggregate of N independent identically distributed (IID) IPP sources produces an $(N+1)$-state MMPP pro:ess. Thus the IPP model is a special case of the MMPP model. Assuming each of the N IPP sources has parameters α and β as described in Figure 2.2, the resulting MMPP process is given in Figure 2.4 [38].

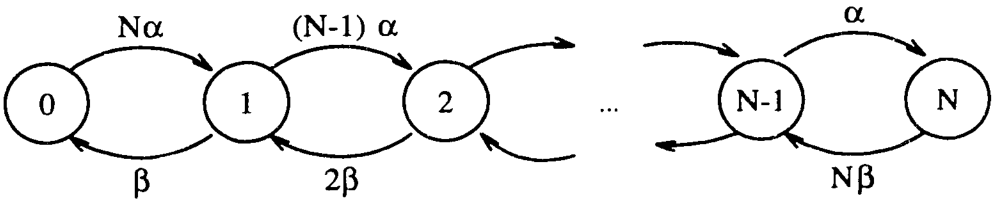
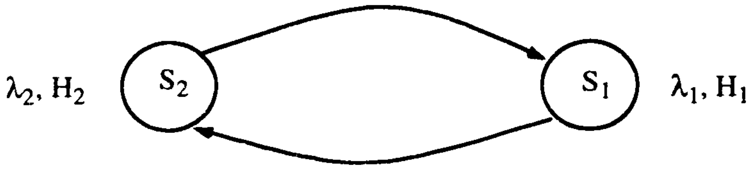


Figure 2.4: MMPP Model or Birth-death Model for N IPP Sources

The two-state MMPP is usually used for modeling aggregate voice and data sources [11]. The state transition diagram is shown in Figure 2.5. In state S_1, the mean holding time is H_1 with cell arrival rate λ_1. In S_2, the mean holding time is H_2 with arrival rate λ_2. The duration of each state is exponentially distributed and in each state, cells are generated according to a Poisson process. Note that this model is less bursty than the IPP model since it does not have a silent period. This is reasonable for aggregate traffic where one call may be in an active state when another call is in its silent state.



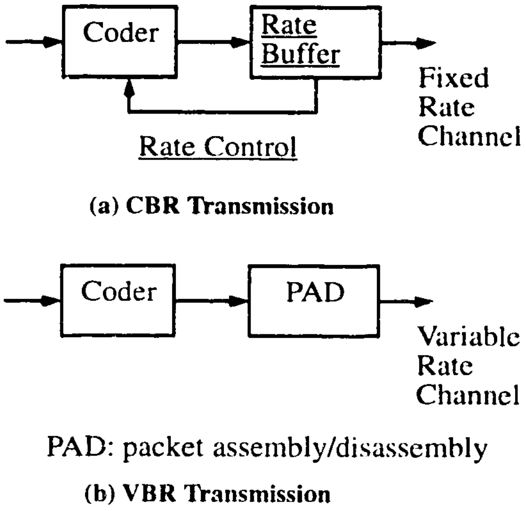
Holding time distribution: Exponential
Arrival rate distribution: Poisson

Figure 2.5: Two-state MMPP Model

The validity of the MMPP model depends on the values of the four parameters. Several studies have been made to set these values. In [11], the four parameters are calculated by matching a set of characteristics of the multiplexed traffic. These characteristics can be obtained from simulation of the aggregate source. The comparisons of the MMPP model and simulation results show that the model is accurate. In [13] two other sets of multiplexed traffic characteristics were proposed to determine the values of the four parameters. The authors claim that the two-state MMPP model defined by their statistics is more accurate than [11] in predicting packet loss because they consider the effects of correlation in choosing appropriate statistics.

2.4 VBR Video Source Models

With the fast development of multimedia communications, the video service becomes increasingly important. There are two types of coding methods for video: constant bit rate (CBR) and variable bit rate (VBR). Figure 2.6 shows the basic concepts of CBR and VBR coding [16]. In the conventional CBR coding, a large rate buffer is needed to maintain the constant bit rate and optimize video coding parameters [27]. This introduces memory cost and delay due to the large buffer memories. Furthermore, constant quality cannot be maintained for all scenes with CBR coding unless peak rate bandwidth is allocated. This wastes bandwidth. The alternative is VBR coding. VBR coding does not need.the rate buffer. It has a PAD (packet assembly/disassembly) unit to process the bits generated by the coder.



(a) CBR Transmission

PAD: packet assembly/disassembly

(b) VBR Transmission

Figure 2.6: CBR and VBR Transmission Concepts

There are usually two types of VBR coding methods: interframe and intraframe. The interframe coding algorithms compress the video by encoding significant differences between successive frames. Therefore, it is efficient when successive frames have little difference as is the case of video with low motion activities. When the video has high motion activities or scene changes, an intraframe coding algorithm is more efficient. The intraframe schemes focus on the interesting area within each frame and send more accurate information for that area. Some VBR coding schemes use only interframe coding because scene changes and high motion do not occur frequently in many video applications such as a video conference. Some other VBR coding schemes such as MPEG combine both interframe and intraframe coding methods.

The source characteristics of VBR video are very complex because they are affected by different coding algorithms and video contents. This section will survey the models proposed for video sources using different VBR coding algorithms. Some of the models are based on frames. That is, the models try to capture the source behavior frame by frame. Some other models are given at the cell level. Generally, the frame-based sources are more accurate in presenting the frame nature of a single video source; while the non-frame-based sources are more suitable for aggregate sources. In Section 2.4.1, video models without scene changes or high motion are surveyed. In Section 2.4.2, video models with scene changes and high motion are surveyed. In some cases, models are used for both.

2.4.1 Video Models without Scene Changes or High Motion

Models proposed for video sources without scene changes or high motion are suitable for applications such as video teleconference or video phone. Normally they are derived from experimental data of a coded video sequence showing a talking person. Usually no scene changes, panning or zooming are used in the experiments.

2.4.1.1 AR Source Model

The first-order autoregressive (AR) process can be used to model a single VBR video source [16, 17]. Within a frame n, cells are generated at a constant bit rate $\lambda(n)$ The AR process produces the bit rate for each frame and can be expressed recursively as follows:

$$\lambda(n) = a\lambda(n-1) + bw(n) \qquad \text{(EQ 2.1)}$$

where $w(n)$ is a Gaussian random variable; a and b are coefficients. Assume that $w(n)$ has mean η and variance 1. Thus, the bit rate of the current frame is calculated from the bit rate of the last frame adjusted by a weight and a Gaussian random variable. It is noted in [17] that the AR process will reach a steady average bit rate with large n. The values of a and b are obtained by matching the average bit rate $E(\lambda)$ and the autocovariance of the bit rate $C(n)$ which can be calculated from the measured data of the video source. The relations of these parameters are given in the following equation.

$$E(\lambda) = \frac{b}{1-a}\eta, \quad C(n) = \frac{b^2}{1-a^2}a^n \qquad \text{(EQ 2.2)}$$

The first-order AR model is validated to be accurate by comparing its results to the measured data of a video without scene changes [17].

2.4.1.2 A Discrete-State, Continuous-time Markov Process

The discrete-state, continuous-time Markov model is not based on frames [17]. In this model, a discrete-state, continuous-time process $\bar{\lambda}(t)$ is used to approximate the continuous-state, continuous-time process $\lambda(t)$ which describes the bit rate of a single video source at time t.

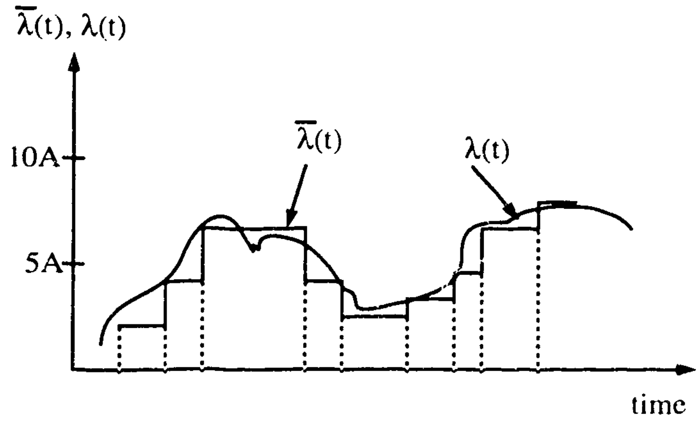


Figure 2.7: Poisson sampling and quantization of the source rate

The process $\lambda(t)$ is obtained from the measured rate of the source. Then, the process $\bar{\lambda}(t)$ is built by sampling $\lambda(t)$ at random Poisson distributed time instances. The process $\bar{\lambda}(t)$ has a finite number of states that correspond to finite discrete levels which are multiples of the quantization step A as shown in Figure 2.7. To make a more accurate model, one can decrease the quantization step A and increase the Poisson sampling rate. A birth-death Markov model represents the behavior of a source. The discrete levels are represented by the different states in the birth-death model. The parameters of this model are the quantization step, the number of states, and the state transition rates. These parameters are obtained by matching with the measured data of video.

This model can be used for single video sources as well as aggregate video sources. The analytical results of this model are close to the simulation results of the AR model for a video coded by an interframe coder [17].

2.4.1.3 A Source Model for Subband Coder

A video source model for a subband coder which uses several subbands to code the video signals is used for simulation in [31]. It models a single video call with fixed frame duration of $F = 62.5\,\mathrm{ms}$. Within a frame, the source has an active period A during which it generates cells at a constant bit rate of 10 Mbits/s. The active period A is a uniform random variable between 10 ms and 40 ms. After the active period, the source enters an idle period of $F-A$ during which no cells are generated. The source repeats starting a new frame after the end of the last frame. Therefore, the average rate of the source is 4 Mbits/s. An aggregate source model can also be defined based on this model. A source consisting of K video calls may be modeled by uniformly distributing the frames starting at intervals of (F/K) ms. This model was originally proposed in [32] and validated by the real-time measurements of traffic from a simulated subband video coder.

2.4.1.4 A Model for Video Conference

A source model for video teleconference is proposed in [34]. The fixed frame duration of 40 ms is used for a single video source. The number of cells per frame is computed using a Markov model. The Markov chain needs three parameters: the mean and variance of the number of cells per frame, and correlation between the number of cells in successive frames. Let k be the number of cells per frame and f_k be the negative binomial probability.

$$f_k = \binom{k+r-1}{k} p^r q^k \qquad \text{(EQ 2.3)}$$

where $p = (mean)/(variance)$ and $r = mean \times p/(1-p)$. The Markov transition matrix is then given by the following formula.

$$M = \rho I + (1-\rho) Q \qquad \text{(EQ 2.4)}$$

where ρ is the correlation factor between successive frame sizes (number of cells per frame), I is the identity matrix, and each row of Q consists of the negative binomial probabilities $(f_0, \ldots, f_K, F_K)$ where $F_k = \Sigma_{k>K} f_k$ and K is the maximum number of cells per frame.

The values of the three parameters are obtained by the measured data of real video traffic. The sequences of video teleconference showing a head-and-shoulders scene with moderate motion and with little camera zoom or pan are used for the measurement.

2.4.2 Video Sources with Scene Changes and High Motion

The behavior of a VBR video source varies with different motion states (e.g. low, medium and high motion) and scene changes. Different coding methods may be used for the different motion states. Thus the models should capture the characteristics of different motion states and scene changes. Models for video sources with scene changes and high motion are surveyed in this section.

2.4.2.1 A Video Model of Three Motion States

This model for single video source is motivated by the fact that the VBR video within a scene can stay in one of the three motion states: low, medium and high motion. The bit rate of the coder is higher during the higher motion state. The peak bit rate occurs during scene changes. An adaptive inter/intra frame coding scheme is used in [18]. It first determines the motion class of the next frame by some detection technique, then applies the appropriate coding scheme to the frame. That is, the interframe coding method is used for low and medium motion frames, and the intraframe coding method is used for high motion frames.

The video source during the three motion states are modeled by three first order AR processes with corresponding parameters to each state. The AR process generates the number of bits for each successive frame in a motion state. The number of frames in each motion state follows a geometric distribution. It can be used to indicate the duration of the motion state because the frame duration is fixed.

The number of bits $\lambda_i(n)$ generated during the *n*th frame of state *i* (*i*=1: low motion, *i*=2: medium motion, *i*=3: high motion) are given by (EQ 2.5).

$$\lambda_i(n) = a_i\lambda_i(n-1) + G_i(n) \qquad \text{(EQ 2.5)}$$

where a_i is the coefficient of the AR process for state *i*. $G_i(n)$ is a Gaussian random variable for state *i*.

The values of the parameters of this model are estimated and obtained from measured data of a VBR full-motion color video sequence with 500 frames [18]. This model is validated by comparing its simulated bit rates with actual measured bit rates.

2.4.2.2 TES-Based Video Source Model

The Transform-Expand-Sample (TES) model is used for modeling a single video source coded by a scheme which combines intraframe and interframe coding methods [26]. Its main feature is to produce a distribution or approximation which closely matches the empirical histogram of a data set.

In [26], each frame of the video is divided into 12 groups of blocks (GOB) for coding purpose. The histogram of the bit rate of GOB's is obtained by measuring real video traffic. Then TES is applied to generate the appropriate distribution according to the histogram. The advantage of TES is that it captures the autocorrelation of the random variables.

This model is used for simulation study. It is verified with a single video source of video phone. The simulation results of the model match the empirical data very well. It is also demonstrated that the TES model is more accurate than the AR model because the ordinary first order autoregressive processes do not consider autocorrelation.

2.4.2.3 Source Models for MPEG Video

The Motion Picture Expert Group (MPEG) [20] coding algorithm is used for storing compressed video on digital storage media. Then the compressed video can be randomly accessed and decoded. The MPEG algorithm combines interframe and intraframe coding techniques. It is applied to a wide range of video applications such as high definition television and multimedia workstations.

In [39], a method to predict the number of cells generated in the next frame according to the information of the current frame and experimental data is proposed. It is used to allocate bandwidth for the next frame. Let c_{n-1} be the actual number of cells generated in frame *n-1*. Then the prediction of number of cells in the next frame $\bar{c}_n$ is calculated as follows.

$$\bar{c}_n = max(\mu, c_{n-1} + \Delta) \qquad \text{(EQ 2.6)}$$

where Δ is the standard deviation and μ is the mean number of cells in a frame. These values are obtained by experimental data. The minimum of the prediction is the mean.

Some other models for MPEG video are also proposed. In [20], the Gamma distribution is found to be suitable for modeling the number of cells generated per frame of a low bit rate MPEG video. However, no distribution is found to be suitable for high bit rate sources. In [40], the TES model which is described earlier is used to model an MPEG video source. The values of its parameters are determined by experimental data.

2.4.2.4 A Histogram-Based Model

A histogram-based model for single and aggregate video sources is presented in [24]. In this model, the cell rate λ in a frame is characterized by a random distribution during the video transmission. A Poisson process is assumed for the cell arrivals in each frame. Since each ATM cell has the same size, its service time is fixed. Therefore, the system is an M/D/1/N queue for each frame. This distribution can be approximated by the bit rate histogram of the video source.

It is shown in [24] that the multiplexer performance for VBR video sources is affected by the way the data is transmitted in each frame. The worst case is that all the cells generated in a frame are transmitted at peak rate from the beginning of the frame. The best case is that all the cells generated in a frame are transmitted uniformly over a frame period. It is possible to implement the best case if the cells are delayed for one frame and the cell arrival process in a frame can be approximated by Poisson process.

The buffer occupancy distribution at a single network access node can be found by solving the M/D/1/N problem. Thus cell loss probability can be obtained. This model is validated by comparing its buffer occupancy with simulation results. It does not depend on specific coding algorithms. However. the histogram of video source must be obtained before transmission.

2.4.2.5 A Multiple Level Markov Model

This model is a multiple state Markov process where each state has a fixed data rate [15]. It has two basic data rate levels: the high rate levels and the low rate levels. The high rate levels are used to characterize scene changes. The low rate levels are used to characterize bit rate changes within a scene. Within each high or low rate level, several states are defined by bit rate and the state transitions are controlled by a Markov chain. The state transitions between the rate levels are also controlled by Markov chains.

This model can be used for both single video sources and aggregate video sources. It is also analytically tractable. Its parameters can be determined by matching the main statistics of the video data such as the ratio of the average rate in the high rate level to that in the low rate level and the overall mean bit rate [15].

2.5 Summary

We have introduced models of data, voice and VBR video sources for ATM networks. Many of the models described here are based on a two state model. The states correspond to a source transmitting at different bit rates. Models vary on how cells are generated within the states and how the duration of the states are found. These sorts of models have been suggested for all three types of traffic: data, voice and video. More complex models have been suggested for video since video sources are complicated by the possibility of different coding schemes.

The congestion control strategy of ATM network must take into account source characteristics. We develop a call admission scheme in this thesis which tries to capture the burstiness of the sources. In order to generate diverse traffic, we use several source models for the simulation study. We use the AR model and the model for subband coder for video sources in the simulation study to give some variety. We also use MMPP model for aggregate data sources. The On-Off fluid model and IPP model are used for single voice sources because of their popularity.

CHAPTER 3 Survey of Call Admission Schemes

3.1 Introduction

Call admission control (CAC) determines whether to accept or reject a new call. The optimal CAC must maximize the utilization of network resources and accept the maximum number of calls while maintaining the QOS of all accepted calls. The QOS requirements include cell loss and cell delay. The network resources include link capacity, buffer size and processor power at each node. The processor power and desire for fast CAC decisions limit the complexity of the CAC. The current usage of the link bandwidth and the bandwidth requirement of new calls are important considerations in making CAC decision.

CAC in ATM networks must deal with different types of traffic and services. And therefore it needs knowledge of the bandwidth requirement of the new call. Normally it requires that a user specifies traffic characteristics and the QOS of his call so that it can calculate how much bandwidth the call needs and make admission or rejection decisions. The bandwidth requirements range from a few kilobits per second to several hundred megabits per second. And the traffic may be continuous or highly bursty. Some traffic is loss sensitive while some is delay and/or delay variance sensitive. Therefore, a set of traffic descriptors is needed to describes the characteristics of traffic. Some possible traffic descriptors are peak rate, average rate, the rate in burst, the burst period distribution, the silent period distribution, and the maximum burst size. Some desirable characteristics of traffic descriptors are as follows.

1. Be few in number.
2. Be understandable and easy to specify by the user.
3. Allow for effective resource allocation by CAC.

CAC must also calculate the free bandwidth (or bandwidth usage) of the links. The performance of a CAC scheme depends on how accurate the bandwidth calculation is.

Therefore, the method of calculation and the selection of traffic descriptors are very important. An effective CAC scheme should:

1. Provide QOS to each accepted call.
2. Define a set of good traffic descriptors.
3. Avoid over-allocating bandwidth to a call and achieve great bandwidth saving through statistical multiplexing (high utilization).
4. Require minimal computation and provide reliable control.
5. Tolerate inaccurate specification of traffic characteristics because users may not be able to specify them accurately.

There is a trade-off between QOS and bandwidth saving. If bandwidth is over-allocated to the calls, the QOS can be satisfied but some bandwidth is wasted. If bandwidth is under-allocated, congestion may occur and QOS will not be met. The CAC scheme tries to provide QOS to each accepted call while saving bandwidth.

In this chapter we survey some recent proposals for CAC schemes. These schemes can be separated into two types. The first type can be called static or fixed CAC. This type of CAC calculates the bandwidth needed by a new call according to the user-specified values of traffic descriptors. If the network has more free bandwidth than what is needed, the call is accepted. Otherwise, it is rejected. When a call is admitted, the network updates its free bandwidth by deducting the needed bandwidth. The static CAC may not be reliable because users may not be able to provide accurate information about their calls and therefore calculation of the needed bandwidth may not be correct. The second type can be called dynamic CAC. This type of CAC admits a call in a similar manner to the static CAC. However, it adjusts the bandwidth allocation with measured information of the network. The measured information could be the current link utilization, the number of calls, and the current bit rate of a call. The free bandwidth can be derived from current link utilization. We believe that the dynamic CAC is more appropriate because it uses the current information of the network.

The CAC schemes are also separated into three groups according to the number of traffic descriptors. The number o^f ɜscriptors in a CAC scheme should be as few as possible so that users can specıfy them easily. The CAC schemes with one traffic descriptor are surveyed in Section 3.2. Those with two traffic descriptors are surveyed in Section 3.3. Those with three or more traffic descriptors are surveyed in Section 3.4. The summary is in Section 3.5.

3.2 CAC with One Traffic Descriptor

The CAC schemes with one traffic descriptor are surveyed in this section. Users may prefer them because they only need to provide a small amount of information for their calls. However, a scheme may not perform well because there is little information aɓout a call's request. Some schemes try to solve this problem by dynamic measurement.

3.2.1 Peak Rate Allocation

The peak rate call admission scheme allocates the peak rate to each call so that the sum of bandwidth of all virtual circuits in a link is less than the link's capacity. Bandwidth enforcement is applied before the cells from each call enter the network. This scheme can provide QOS of accepted calls if the users give the correct peak rates. Bandwidth saving through statistical multiplexing cannot be fully achieved since it over-allocates bandwidth to calls.

Gersht and Lee [4] propose a congestion control framework based on this approach. They suggest *express* service and *first-class* service for real-time (e.g. voice and video) and nonreal-time applications (e.g. data), respectively. For express calls, users request the peak rate (denoted by P_k). For first-class calls, users request both the peak rate and a guaranteed rate (denoted by G_k). Cells are transported on each route according to the total reserved bandwidth of the calls using the route (similar to virtual path). For a call k, the bandwıdth reservation at call setup is as follows:

$$M_k^{(s)} = P_k \tag{EQ 3.1}$$

$$M_k^{(b)} = \begin{cases} (1+\eta)P_k & \text{for express calls} \\ G_k + \gamma(P_k - G_k) & \text{for first class calls} \end{cases} \quad \text{(EQ 3.2)}$$

$$M_k^{(d)} = P_k \quad \text{(EQ 3.3)}$$

$M_k^{(s)}$ is the bandwidth reserved on the source user network interface (UNI). $M_k^{(b)}$ is the bandwidth reserved on the backbone links. $M_k^{(d)}$ is the bandwidth reserved on the destination UNI. The parameter η $(\eta > 0)$ is used to control the express cell loss probability at the router (or switch) and the parameter γ $(0 < \gamma < 1)$ is used to control the average delay of first-class VC's. These parameters are adjusted so as to satisfy the performance requirements. Let $W^{(s)}$, $W^{(b)}$, $W^{(d)}$ denote the transmission capacities of source UNI, backbone links and destination UNI, respectively. The call admission scheme limits the maximum permitted bandwidth reservation by rejecting calls. A call is rejected if any of the following conditions are violated:

$$\frac{\sum M_k^{(s)}}{W^{(s)}} \le \hat{\rho}^{(s)} \quad \text{(EQ 3.4)}$$

$$\frac{\sum M_k^{(b)}}{W^{(b)}} \le \hat{\rho}^{(b)} \quad \text{(EQ 3.5)}$$

$$\frac{\sum M_k^{(d)}}{W^{(d)}} \le \hat{\rho}^{(d)} \quad \text{(EQ 3.6)}$$

where the maximum allowed utilization levels ($\hat{\rho}^{(s)}$, $\hat{\rho}^{(b)}$, $\hat{\rho}^{(d)}$) are determined by the network provider to control the cell loss probability. Usually the maximum allowed utilization level is close to 1. That is, the total reserved load or utilization should be no more than the allowed utilization level at the source UNI, backbone links and destination UNI. A reactive congestion control at cell-level is used for first-class VC traffic. There is no congestion for express-class VC traffic.

This approach does not require much computation and is easy to implement. The traffic descriptor is peak rate which users can understand easily, though it may be difficult to specify accurately.

3.2.2 A Dynamic CAC by Boyer

Boyer [43] assumes that peak bit rate is the only traffic parameter the users are able to provide. Furthermore, he suspects that the actual activity of a user's call will be different from his traffic contract either by ignorance or malignancy. He proposes a congestion control framework with three levels: call, burst and cell levels. The burst level is to control bursty sources. The cell level control is for policing or peak rate enforcement. We are interested in the call level control.

The user must declare the peak rate Λ of the new call. Each node in the network updates a measurement ρ_p of the load at its outgoing links. Let μ be the total link bandwidth, and ρ_N be the load nominal value[1]or the fraction of the link capacity. The call can be admitted according to the maximum load contribution of the call and the total load assessment. That is, the following equation must be satisfied in order to accept a call.

$$\rho_p + \frac{\Lambda}{\mu} \leq \rho_N \qquad \text{(EQ 3.7)}$$

This CAC takes into account the measured link load. It requires the user to specify the peak rate, but uses measurement subsequently to handle incorrect specification of peak rate.

3.2.3 Schedulable Region

Hyman, Lazar and Pacifici [31] define the *schedulable region* as a region of load within which the QOS of the calls in the system is guaranteed. They define three classes of calls: class 1 for video calls, class 2 for voice calls and class 3 for data traffic. The calls within each class are assumed to be identical.

[1] In the paper, the load nominal value is used for dimensioning purpose.

They propose one source model for each of the three classes. Then, the schedulable region is found by simulation. The region specifies the threshold of the number of calls in each class that can be admitted if the number of calls in the other two classes are provided. Users do not worry about traffic descriptors. They are simply required to indicate the class of their calls. For making a call admission decision, the loads of the calls from different classes should be within the schedulable region.

For the sources used in this paper, this CAC provides QOS to all the accepted calls. Its computation is simple once the schedule region is constructed. However, whenever a new class of application is added, the schedulable region has to be reconstructed by simulation. Also, the calls in each class are assumed to be identical which may not be accurate.

3.3 CAC with two traffic descriptors

This section describes CAC schemes with two traffic descriptors. It may be less attractive to the users because more information is requested. These schemes may exhibit good performance if users are patient, honest and informed enough to provide accurate information.

3.3.1 A dynamic CAC by Saito

Saito and Shiomoto [5] proposed a dynamic CAC scheme. The traffic descriptors are peak bit rate and average bit rate. The CAC admits a new call if the upperbound of cell loss probability is less than the maximum admissible cell loss probability assuming the call is accepted. Otherwise, the call is rejected. The upperbound is calculated by the current measurement of cell arrival rate and the traffic descriptors of the new call.

The delay requirement is satisfied by the buffer size dimensioning method in [6]. Assume FCFS order for the cells at the transmitter, then the buffer size K is given by the following equation to satisfy the maximum delay T.

$$K = (TC)/L \qquad \text{(EQ 3.8)}$$

Here C is the link capacity and L is the length of the cell. The cell measurement is applied in several renewal periods. Figure 3.1 shows the measurement scheme. The renewal period consists of N measurement periods of length s. The measured distribution of number of cells in these N measurement periods at each renewal period is used to calculate the upperbound of cell loss probability.

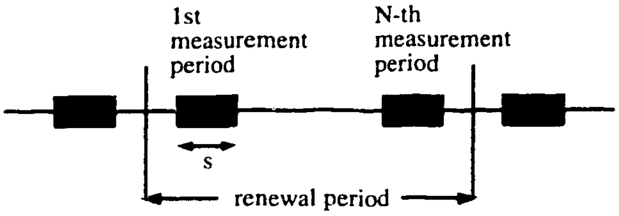


Figure 3.1: Measurement Scheme

This CAC scheme is compared with a fixed CAC scheme that only uses traffic parameters specified by users. Simulation results show that this dynamic CAC scheme is more effective than the fixed CAC and the cell loss probability requirement is satisfied. As a side note, the mean call duration is 10 ms and the upperbound of cell loss probability is set to 10^{-3} in the simulation.

This scheme uses measured information of the network to make CAC decisions. Therefore it can tolerate errors in the user specifications upon call setup. Computation is needed after each measurement. The computation should be fast so that the CAC can update bandwidth allocation quickly. The measurement interval is also an important parameter for the network performance. Further study is needed to decide the length of measurement interval and whether it should be constant or variable.

3.3.2 Distributed Source Control

A CAC scheme which uses a window based control called DSC (Distributed Source Control) is proposed in [7]. Two control parameters, the number of cells in a window W_S and window interval T_S, are important in DSC. The access node admits no more than W_S cells from the source into the network every nonoverlapping interval of T_S seconds. These two parameters are negotiated between the source and the network access node. The average throughput $\bar{\lambda}$ of the source during its active period is equal to W_S/T_S.

A new call must specify its required average bandwidth or throughput $\bar{\lambda}$ when active. The new call can also specify the minimum throughput $\bar{\lambda}_{min}$ that it will accept. Every node along the path of a virtual circuit must check two aspects for the new call so as to admit it.

- First, the sum of the average bandwidth requirements of all active VC's, using the link of the node, must be less than the link bandwidth. This ensures that the link is not overloaded.
- Second, the sum of the DSC windows of all active VC's must be less than the buffer size at each node. This ensures that no cells are lost due to buffer overflow.

If the bandwidth requested by the call cannot be met by a node, the node can offer a reduced value of bandwidth. If the throughput with the reduced bandwidth is less than the minimum throughput $\bar{\lambda}_{min}$, the call is rejected.

The new call is assigned to virtual circuit i if it is accepted and the DSC parameters W_{Si} and T_{Si} are chosen so that the throughput is equal to the negotiated throughput of the call. It is demonstrated that the delay and cell loss performance dramatically improve if the smoothing interval T_S decreases. Therefore, T_S is chosen to be small.

Two other controls are applied in DSC to regulate the cell rate into the network at the network access node.

1. A link-level control (LLC) between the data source and network access node. It is a reactive control to protect the buffer at the network access node from overflow. The reactive control is effective since the distance between the source and the access node is short and the propagation delay is not large.
2. An adaptive end-to-end control (EEC) between the network access and egress nodes. This is a window control which is used as a backup to LLC. Let T_R be the round trip delay. The end-to-end window size W_E must satisfy the following equation,

$$W_E \leq \bar{\lambda} T_R \qquad \text{(EQ 3.9)}$$

This CAC method avoids over allocating bandwidth to a call. The computation time for this contro! is not large. There are two drawbacks. First, since window mechanism is used, the source traffic cannot exceed the predefined throughput $\bar{\lambda}$. This does not permit slight violation even if there is enough free bandwidth. Second, since idle sources are not considered when a new call is considered for admission, an idle source may find no sp.:⁻ᵉ bandwidth to use when it becomes busy again and will have to wait.

3.4 CAC with three or more traffic descriptors

The CAC schemes in this section request more information from users. Their performance should be very good if users can provide accurate values of traffic descriptors. However, it may not be realistic to request so much information from users.

3.4.1 Equivalent Capacity Allocation

Equivalent capacity is a metric to represent the bandwidth requirement of single or multiplexed connections. The equivalent capacity of a set of connections multiplexed on a link is defined as the amount of bandwidth required to achieve a desired QOS. The buffer overflow probability when a source is active was selected as the QOS measurement in [8]. The equivalent capacities are obtained by negotiating between the results of the two approaches: fluid-flow approximation and stationary bit rate approximation.

The On-Off fluid model is used for the fluid-flow approximation. As described in Section 2.3, the source consists of two states: idle and active. It transmits at zero bit rate in the idle state and it transmits at peak bit rate in the active state. The idle and active periods are exponentially distributed. With a given buffer size x and the upperbound of buffer overflow probability ε, the equivalent capacity of a single connection can be found by queueing analysis. For the N multiplexed sources, the total equivalent capacity $\hat{C}_{(F)}$ is the sum of the equivalent capacities of the single sources. The fluid-flow approximation is easy to compute, however it ignores the effect of statistical multiplexing and thus may overestimate the amount of bandwidth required.

The stationary bit rate approximation is used when the statistical multiplexing is dominant. The value $\hat{C}_{(S)}$ of the equivalent capacity is determined so that the aggregate stationary bit rate exceeds $\hat{C}_{(S)}$ only with a probability less than ε, the desired buffer overflow probability. This overestimates the actual bandwidth requirement because it ignores the "smoothing" effect of buffering. The equivalent capacity is the smallest value of $\hat{C}_{(S)}$ which satisfies:

$$Pr(B > \hat{C}_{(S)}) \le \varepsilon \qquad \text{(EQ 3.10)}$$

where B is the aggregate bit rate. The distribution of the aggregate bit rate B is assumed to be a Gaussian distribution. Therefore, $\hat{C}_{(S)}$ can be obtained by approximation of an inverse Gaussian distribution.

Since both approximations overestimate the actual equivalent capacity for the two different multiplexing behaviors, the equivalent capacity $\hat{C}$ is the minimum of $\hat{C}_{(F)}$ and $\hat{C}_{(S)}$. The call admission decision is based on the value of equivalent capacity $\hat{C}$ of all existing calls and the new call at each link along the route of the new call. The equivalent capacity $\hat{C}$ of the calls in a link should not exceed the link capacity. The authors claim that the computation of this method is simple.

This approach bases its results on a specific source model. Other approaches for obtaining equivalent capacity are also proposed in the literature. For example, Elwalid and Mitra use the Markov Modulated Fluid model and MMPP model to calculate the effective bandwidth (or equivalent capacity) of multiplexed sources[46].

3.4.2 A Dynamic CAC by Bolla

Bolla, Danovaro and Marchese proposed in [28] a hierarchical CAC scheme that decomposes the traffic into several classes. Each class is controlled by a call admission controller which operates within the class. A bandwidth allocation controller periodically reallocates bandwidth to each class based on the call blocking information in each class from previous periods. The source model used in the study is the On-Off fluid model. The connections within each class have the same traffic characteristics and are independent of each other.

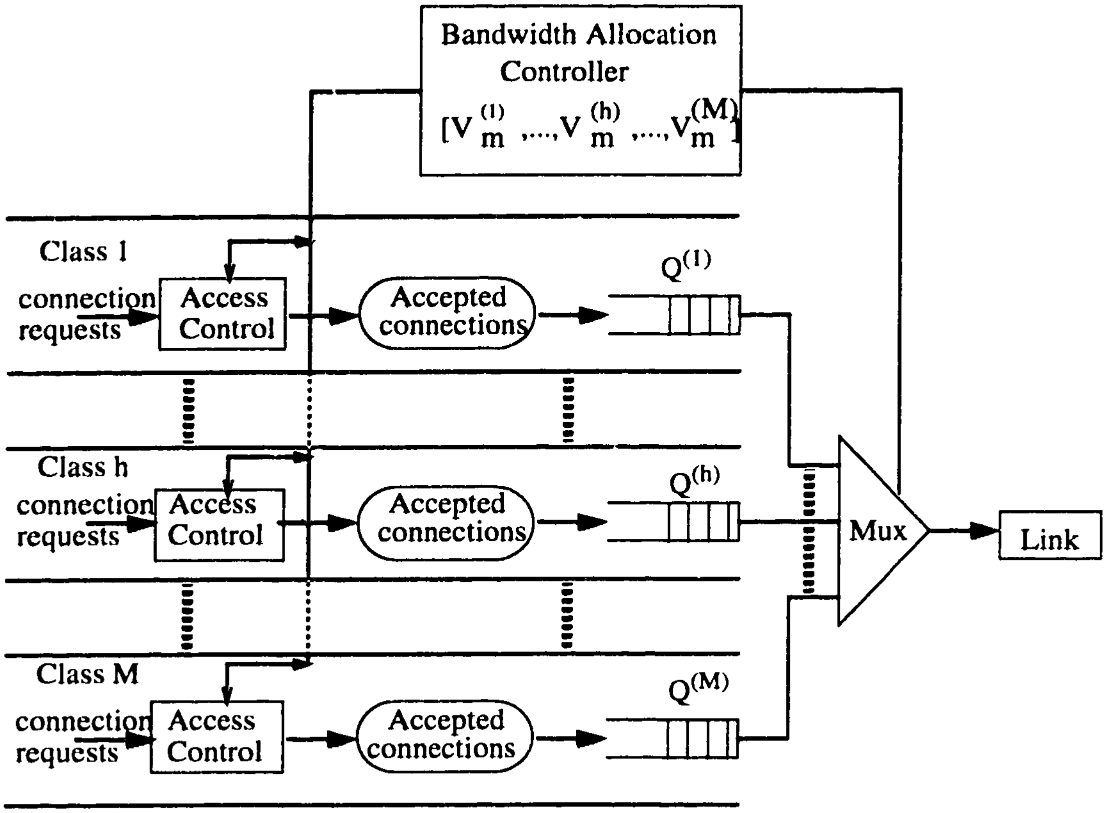


Figure 3.2: Structure of the overall control system

Figure 3.2 shows the structure of the scheme. There are M admission controllers (one for each traffic class) and a bandwidth allocation controller. Each admission controller makes admission decisions for the specific class according to the current "virtual capacity" which is dynamically assigned to the class by the bandwidth allocation controller. A time slot is defined to be equal to the duration for transmitting a cell in the system. The allocation controller divides the link capacity C_T into virtual capacities $V_m^{(h)}$, class $h = 1, \ldots, M$, where $m = 0, K, 2K, \ldots$ are the instants of bandwidth reallocation. Here, the length of reallocation period is K slots. Each admission controller calculates the number of acceptable connections based on the loss and delay requirements for the current period with the source model. The total bandwidth requirement of the calls within a class cannot exceed the virtual capacity of the class.

The bandwidth allocation controller reassigns the virtual capacities $V_m^{(h)}$ at every K slots by means of minimizing a cost function. The cost function takes into account the blocked calls in each class during the previous period and attempts to allocate more bandwidth to classes with blocked calls.

The paper [28] does not discuss the traffic descriptors. However, it suggests that the parameters of the source model can be used. The CAC may require peak bit rate, mean active period and mean idle period of the call from users since On-Off fluid model is used.

It is clear that this approach can provide QOS to all the accepted calls if each connection fits well into its predefined class. That is, the user specified parameters should be accurate. The computation required in this method is simple. However, it may become a bottleneck if the reallocation period is short.

3.4.3 A Dynamic CAC by Tedijanto

Tedijanto and Gün [30] proposed a control scheme that operates at two levels, the connection level and the cell level. Connection-level controls include path selection, admission control and bandwidth allocation. Cell-level controls are access control at the UNI and buffer management at intermediate nodes. The required traffic descriptors are peak rate R, the mean rate m and the mean burst length b (number of bits in a burst). The source model is an On-Off fluid model. A simple approximation method is used to map the user traffic into the model characterized by (R, m, b).

The network model is a network node multiplexing N connections. Each connection is monitored by a leaky bucket. It is assumed that the peak rate R does not change for the duration of a connection. The bandwidth needed for a connection is called the equivalent capacity. It is calculated given the desired cell loss probability and the buffer size. The call admission decision is based on the equivalent capacity of a new call and the available bandwidth of the nodes along the route.

Cell-level controls are implemented by leaky buckets. There are two types of leaky buckets, namely queueing leaky bucket and tagging leaky bucket. In a queueing leaky bucket, cells are queued if no token is available. In a tagging leaky bucket, cells are tagged and sent into the network if no token is available. The tagged cells will be discarded first in case of congestion. The leaky buckets are used to monitor traffic and measure its key parameters. Then the connection-level controls adjust the bandwidth allocated to the connections according to the measurement.

The reasons for dynamic leaky buckets are as follows. First, it is necessary to monitor the sources to detect the misbehaving users. Second, many users cannot accurately specify the traffic characteristics. Third, the source characteristics may change during a long time connection. Therefore, the control should be able to adjust the allocated bandwidth so as to utilize bandwidth efficiently.

A peak rate controller is placed in front of each leaky bucket at the access node. This is to smooth the traffic. The cell-level control generates bandwidth update requests according to the measurement. If there is not enough bandwidth for updating the bandwidth assignment, the peak rate controller of the connection which needs more bandwidth should operate with a reduced peak rate. The connection-level may also disconnect or reroute the call.

The mean bit rate and the fraction of cells without token are measured. The measured values and previous values are then used to predict the values of the next interval. These two parameters can be used to compute the equivalent capacity. Therefore, the bandwidth update request can be made upon the prediction.

The peak rate controllers and leaky buckets regulate the source traffic before it enters the network. Congestion control is done at the edge of the network. This CAC can tolerate inaccurate specifications of traffic characteristics.

3.5 Summary

From the discussion above, we find that no call admission scheme meets all the criteria of Section 3.1. There is a trade-off between providing QOS and saving bandwidth in call admission control. The traffic descriptors of a CAC scheme must be simple because users may not be able to specify the technical parameters accurately. Class approach is a good choice because it can be easily specified by users. Dynamic CAC schemes which use measured information of the network may be more reliable than static CAC schemes because more accurate information can be obtained by measurement than traffic descriptors. Therefore, a dynamic CAC scheme which uses a class as its traffic descriptor is designed in this thesis.

CHAPTER 4 The Design of Refinement CAC

4.1 Introduction

We have designed a new CAC scheme called Refinement CAC. The motivation of the new design comes from the discussion of source models and the survey of CAC schemes. There are three main ideas for the new design.

1. Users are not experts. They may not be able to specify the characteristics of their calls such as mean rate, peak rate and burstiness [43]. Therefore, traffic descriptors must be simple. Peak rate is easy to understand and may be taken from equipment specifications. But this generally over-estimates the peak rate [51]. Other traffic descriptors such as mean rate and burstiness are even harder to specify. A simple method is to provide several classes and let users choose which class their calls belong to. For example, the classes can be video, voice or data. They are easy to understand. More specific classes such as slow motion video and high motion video can be defined.
2. Even if users can provide accurate information of their calls, the traffic descriptors may not be sufficient to estimate a call's resource requirements [3]. Dynamic CAC can calculate the free bandwidth (or bandwidth usage) of the links using actual measured data and does not totally rely on the traffic descriptors. Since the class traffic descriptor is so simple and general, a dynamic CAC is preferred.
3. The traffic sources in ATM networks are bursty. They have periods during which cells are transmitted at a high bit rate and periods during which cells are transmitted at low bit rate. The new CAC tries to capture the burstiness by refinement. By keeping track of the behavior of each call in terms of its bandwidth requirements, the new CAC scheme tries to predict the behavior of the calls in the future and use this information to make CAC decisions. For example, if several calls are currently transmitting cells at very low rate, the CAC may predict that the cell rates will become high in the next interval because of burstiness.

We separate calls into several classes. Calls with similar behavior or source characteristics are grouped into the same class and these classes must be easily understood by the users. In this thesis, three basic classes are defined to simplify the simulation design. They are data, voice and video. It is obvious that users can declare their classes without pain. However, more classes may be designed. For example, the video class can be separated into high motion, medium motion and slow motion classes. This would give the system more information on expected behavior than the three classes and results in better control due to the more specific grouping of similar types of traffic.

Each class is further divided into three refined subclasses: high rate, medium rate and low rate. These refined subclasses are invisible to users. Periodically, each call is moved among the subclasses based on a comparison of its current measured bit rate and threshold values for its class. The system uses information of the number of calls in each subclass and measured rates of the calls to predict future behavior and calculate the total adjusted rate of the traffic. Call admission decisions are based on the total adjusted rate and the class of the new call.

4.2 Refinement

Refinement is used to place a call into a particular refined subclass of the chosen class and only occurs at access nodes. Note that the refinement only happens in the same class. That is, the call cannot be put into a refined subclass of a different class. A high bit rate threshold T_h and a low bit rate threshold T_l are used to separate the refined subclasses within each class. The refinement process is described as follows.

1. The medium refined subclass is the default refined subclass. That is, when a call is accepted, it is put into the medium refined subclass. Subsequently, the call will be refined to a suitable refined subclass.
2. The system measures bit rate of each accepted call and makes refinement in a constant interval which is called the refinement interval. The measured bit rate is a moving average which is usually taken over a few refinement intervals. Thus, the bit rate is measured over a period which is longer than a refinement interval. By using a moving average, a longer term measured bit rate is found which is used to make long term call admission decisions. Each class is assigned its own refinement interval because the calls in the same class have similar characteristics.

3. After each measurement, refinement takes place. The measured bit rate of the call R_c is compared to two thresholds T_h and T_l.

   If $R_c < T_l$, the call moves into the low rate refined subclass.

   If $R_c > T_h$, the call moves into the high rate refined subclass.

   Otherwise, the call moves into the medium rate refined subclass.

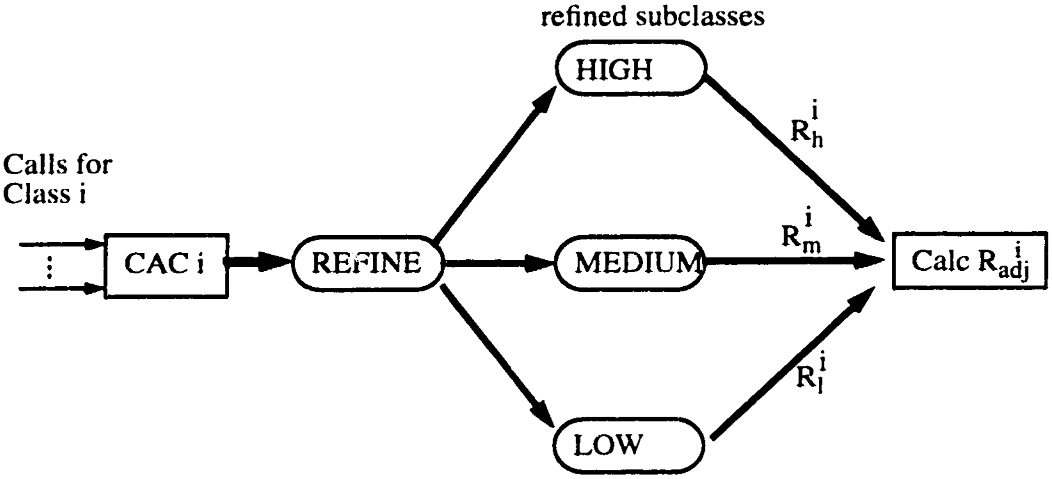
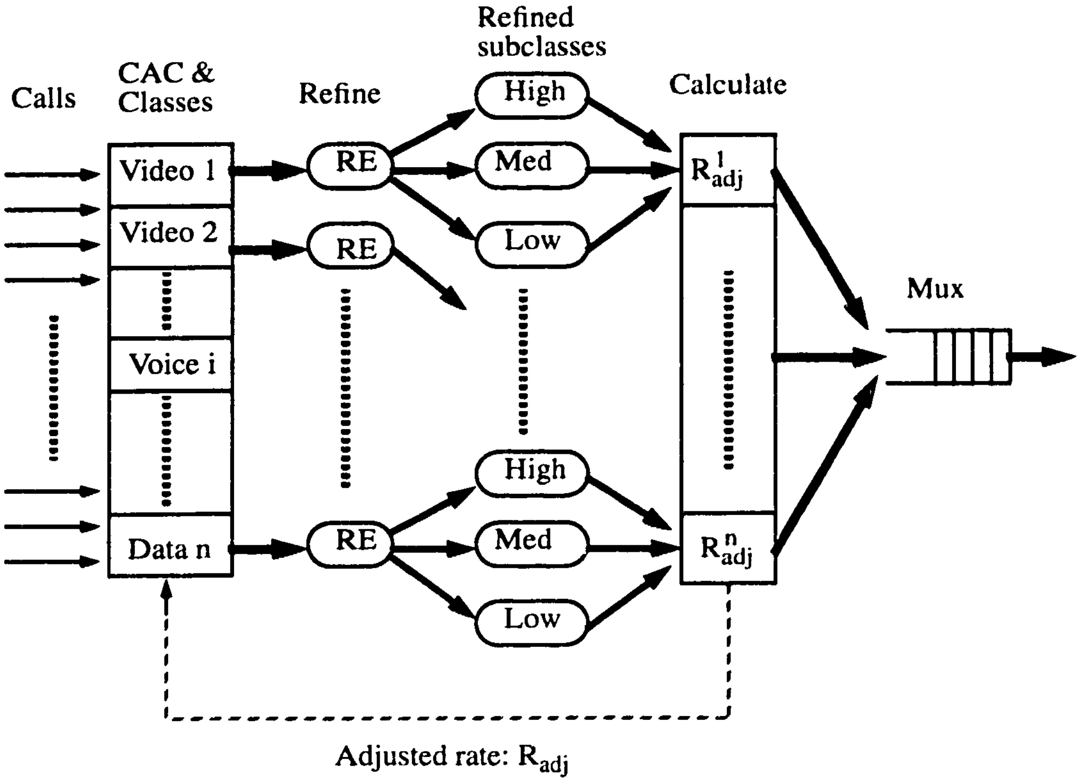


FIGURE 4.1: Class *i* in Refinement CAC

Figure 4.1 shows how refinement works an any class (say class *i*). The system keeps track of the number of calls in each refined subclass for each class. Because of the burstiness of some traffic sources, the Refinement CAC expects that the calls in lower refined subclass will transfer to the higher refined subclass with some probability and vice versa. Therefore, the number of calls in each refined subclass can be used to predict the future behavior of the calls. This prediction is used to adjust the total bandwidth requirement of the network. More details are discussed in the next section.

4.3 The General Architecture of Refinement CAC

In Refinement CAC, several classes are defined and each class has three refined subclasses. CAC takes control at source nodes (or access nodes). Figure 4.2 is the model of the Refinement CAC at a source node. Each class has a default bit rate for its calls, the expected mean bit rate of the class. Each node in the network keeps track of the free bandwidth R_{free} on its outgoing links. In this section, we first describe the call set up procedure. Then we describe the calculation of free bandwidth.



Video 1 could be full motion video
Video 2 could be still motion video

Figure 4.2: Architecture of Refinement CAC

The user declares the class and destination of his/her new call. The bandwidth requirement of the new call R_{new} is set to be the default bit rate of the class. The source node sends a reserve cell with R_{new} along the route to the destination. Before the source node sends the reserve cell, or when a node along the route receives the reserve cell, it compares its free bandwidth R_{free} with R_{new}. If R_{free} is larger than R_{new}, the node reserves the default rate and updates its free bandwidth by deducting the default rate R_{new}. Otherwise, it puts a rejection note in the reserve cell. The reserve cell travels back after it reaches the destination node. When a node (including the source node) receives the returning reserve cell with a rejection note, it releases R_{new} by adding it to the free bandwidth R_{free} if R_{new} was reserved before. If the source node receives the reserve cell without rejection note, the call is accepted. Otherwise, it is rejected.

The free bandwidth R_{free} is updated in different ways at source nodes and internal nodes. R_{free} at internal nodes is updated solely by the default rates of calls to keep the CAC scheme simple. We believe that this control at internal nodes is enough. R_{free} at source nodes is also updated by measurement as described below. The CAC measures and updates the critical parameters such as the bit rates of links at source nodes in a constant interval which is called the measurement interval. The measurement interval is usually set to be long enough to capture the long term average bit rate and include several call arrivals. The following parameters are measured:

- The mean bit rate in the measurement interval of all the calls in each of the three refined subclasses. Let R_l^i, R_m^i and R_h^i denote the measured bit rate in low, medium and high refined subclasses as shown in Figure 4.1. Therefore, the total measured bit rate R_t^i of class i is: $R_t^i = R_h^i + R_m^i + R_l^i$.
- The number of calls in each of the three refined subclasses. The number of calls in low, medium and high refined subclasses are denoted by N_l^i, N_m^i and N_h^i, respectively. The average number of calls over a fraction of the measurement interval is used in this thesis because the number of calls in each refined subclass may change over a measurement interval.

These measured parameters are used to predict the bandwidth requirement of all the accepted calls of class i in the next period. This predicted bandwidth is called the adjusted rate of class i. The adjusted rate of class i is denoted by R^{i}_{adj} and calculated with following equation.

$$R^{i}_{adj} = \alpha R^{i}_{h} + (1-\alpha) N^{i}_{h} B^{i}_{m} + R^{i}_{m} + \beta R^{i}_{l} + (1-\beta) N^{i}_{l} B^{i}_{m} \qquad \text{(EQ 4.1)}$$

B^{i}_{m} is the default rate of a call in class i. The parameter α is the probability of a call in the high refined subclass staying in the same refined subclass for the next interval. The parameter β is the probability of a call in the low refined subclass staying in the same refined subclass for the next interval. All classes have the same values of α and β.

The total adjusted rate R_{adj} of the calls at a source node is the sum of the adjusted rates of all the classes.

$$R_{adj} = \sum_{i=1}^{n} R^{i}_{adj} \qquad \text{(EQ 4.2)}$$

As described earlier in this section, the free bandwidth at a source node is usually updated when a new call comes. The free bandwidth R_{free} is calculated with the following equation whenever the adjusted rate is updated,

$$R_{free} = R_{target} - R_{adj} \qquad \text{(EQ 4.3)}$$

where R_{target} is the target rate of the link which is a fraction of the link capacity. That is,

$$R_{target} = \gamma \cdot C_{link} \qquad \text{(EQ 4.4)}$$

where C_{link} is the link capacity and γ is the fraction of the link capacity. The target rate is proposed because it may be unrealistic to consider the full link bandwidth at call admission time. The CAC uses the target rate of the outgoing link for bandwidth allocation. When a call is released, the number of calls in its refined subclass decreases by one.

At the intermediate nodes, the target bandwidth R_{target} is also used for each outgoing link to prevent internal congestion by traffic from other access nodes. However, they do not calculate the adjusted rate. The free bandwidth R_{free} at an intermediate node is initialized to be R_{target}. When a call is released at this node, R_{free} is increased by its default rate.

The parameters α and β in (EQ 4.1) are used to estimate the transfers of calls among the refined subclasses. They affect the free bandwidth R_{free} as shown in the (EQ 4.1) to (EQ 4.4). Basically, R_{free} increases as α decreases, and R_{free} also increases as β increases. It is obvious that more calls can be accepted if R_{free} increases. By setting the values of α and β, we can define a number of Refinement CAC's. The adjustment of α and β explains what the CAC expects the calls to do in the system. Here are three Refinement CAC's among which R_CAC_Cons and R_CAC_Aggr will be used in the simulation experiments.

- R_CAC_Cons: $\alpha = 1$ and $\beta = 0$.

  This CAC scheme assumes that calls will either stay in or move toward the high refined subclass. Therefore, the adjusted rate is higher than the measured rate, allowing fewer calls to be accepted. This is the most conservative case.

- R_CAC_Meas: $\alpha = 1$ and $\beta = 1$.

  This CAC scheme assumes that calls will stay in the current refined subclass. In fact, refinement is not used since the adjusted rate is the same as the measured rate. Only measurement is used. This CAC accepts more calls than R_CAC_Cons.

- R_CAC_Aggr: $\alpha = 0$ and $\beta = 1$.

  This CAC scheme assumes that calls will either stay in or move toward the low refined subclass. Therefore, the adjusted rate is lower than the measured rate, allowing more calls to be accepted. This is the most aggressive case.

R_CAC_Cons and R_CAC_Aggr represent the extreme cases of this scheme and will be used for comparison with other CAC schemes. The behavior of R_CAC_Meas will always fall between them.

The length of measurement interval is another important consideration because it is the interval for updating the adjusted rate and free bandwidth of the link. If the measurement interval is short, the free bandwidth is updated more frequently and the computation of the CAC becomes large. Furthermore, the measured bandwidth may have less relation to the refined calls in (EQ 4.1). If it is long, the measurement may not be up-to-date.

The refinement interval of a call is fixed for each class. Different classes have different refinement intervals depending on the variability of the calls. The refinement interval should allow the measured bit rate of the call to change so that the call can transfer among the refined subclasses. Also the call should stay in a refined subclass for a while. If the refinement interval is too short, the measured bit rate of a call will be too bursty to represent the real bit rate and the call will jump among the refined subclasses. If it is too long, the measured bit rate of a call may be too close to the mean bit rate and cannot represent the burstiness. Note that the actual measurement period of the bit rate is longer than the refinement interval since the moving average bit rate is used. The value of the refinement interval is obtained by observing the behavior of the source models with simulation and is discussed in Chapter 6.

4.4 Summary

This chapter describes the architecture of the Refinement CAC. This CAC scheme uses refinement and measurement to calculate the free bandwidth at the source nodes. The bandwidth at intermediate nodes is controlled by reserving and releasing the default rate of the calls. We will use simulation to evaluate the performance of the Refinement CAC (R_CAC_Cons and R_CAC_Aggr) by comparing with peak rate allocation CAC and Boyer CAC. The simulation details are described in the next chapter.

CHAPTER 5 Simulation Details and Validation

5.1 Introduction

Refinement CAC will be compared with peak rate allocation CAC (Peak CAC) and Boyer CAC in this thesis. This chapter describes the simulation details and validation. We use event driven simulation to compare their performance. The simulations are written in C. The machines used for the simulations are: SPARC, SGI stations at University of Alberta, SPARC, DEC, and α stations at TR*Labs*.

5.2 Peak CAC and Boyer CAC

We describe the Peak CAC and Boyer CAC for the simulation in this section. Both are CAC schemes with one traffic descriptor: peak rate. As discussed in Chapter 3, the number of traffic descriptors must be as few as possible. Furthermore, peak rate is easy to understand and may be taken from equipment specifications. Thus both of the schemes may be reasonable choices for the CAC schemes in ATM networks. In fact peak rate is used in real ATM switches today. Therefore, it is important to compare with these CAC's. Boyer CAC is a dynamic CAC and Peak CAC is a static one. A similar Peak CAC is described in Section 3.2.1 and the principles of Boyer CAC are described in Section 3.2.2.

In Peak CAC simulation, each node in the network maintains the free bandwidth R_{free} of its outgoing links. R_{free} is initialized to be the link capacity. Calls arrive randomly at the network access nodes (or source nodes). The user specifies the peak rate R_{peak} of his/her new call. The source node then sends a reserve cell with R_{peak} along the route. Before the source node sends the reserve cell, or when a node along the route receives the reserve cell, it compares its free bandwidth R_{free} with R_{peak}. If R_{free} is larger than R_{peak}, it reserves the peak rate and updates its free bandwidth by deducting the peak rate R_{peak}. Otherwise, it puts a rejection note in the reserve cell.

The reserve cell travels back after it reaches destination node. When a node (including the source node) receives the returning reserve cell with a rejection note, it updates its free bandwidth by adding the peak rate R_{peak}. If the source node receives the reserve cell without rejection note, the call is accepted. Otherwise, it is rejected. When a call is released, each node along the route updates its free bandwidth R_{free} by adding the peak rate of the call. Thus, in this scheme the free bandwidth at each node is updated at call admission by subtracting the peak rate of the call and at call release by adding the peak rate of the call.

Boyer CAC does not specify what the CAC should control at the internal nodes [43]. If peak rate of a call is reserved at the internal nodes, the internal nodes will drive the CAC behavior and the results will be similar to Peak CAC. The alternative is to perform no reservation and simply use a fraction of the link capacity to prevent internal congestion. This is the way we choose to implement Boyer CAC.

In Boyer CAC, a reserve cell is also sent out along the route upon call arrival to establish a connection. However, it only reserves peak rate at the source node and does not make bandwidth reservation at internal nodes. If the source node has enough bandwidth for the peak rate, the call is accepted. Otherwise, it is rejected. Like Peak CAC, the free bandwidth at the source node is updated at call admission by subtracting the peak rate of the call. However, Boyer CAC also measures the total bit rates R_t of the outgoing links at the source node. Then, the free bandwidth R_{free} at the source node is updated periodically by the following equation:

$$R_{free} = C_{link} \cdot \gamma - R_t \qquad \text{(EQ 5.1)}$$

where C_{link} is the link capacity and γ is a fraction of the link capacity or the link nominal load. Overall, Boyer CAC is a dynamic CAC with peak rate as traffic descriptor.

5.3 Common Elements of the CAC Simulation Model

In order to fairly compare the three CACs, all CACs must be tested under the same conditions. This section describes the common elements of the CAC's in simulation. The common elements of the simulation include the network model, the traffic source models, the call arrival process, etc.

5.3.1 Network Model

Figure 5.1 is an illustration of the network model that is used in the simulation. It consists of five nodes and four links. Two of the nodes (*A* and *B*) are network access nodes (or source nodes). One of the nodes (*C*) is an intermediate node which relays the cells to the two destination nodes (*D* and *E*).

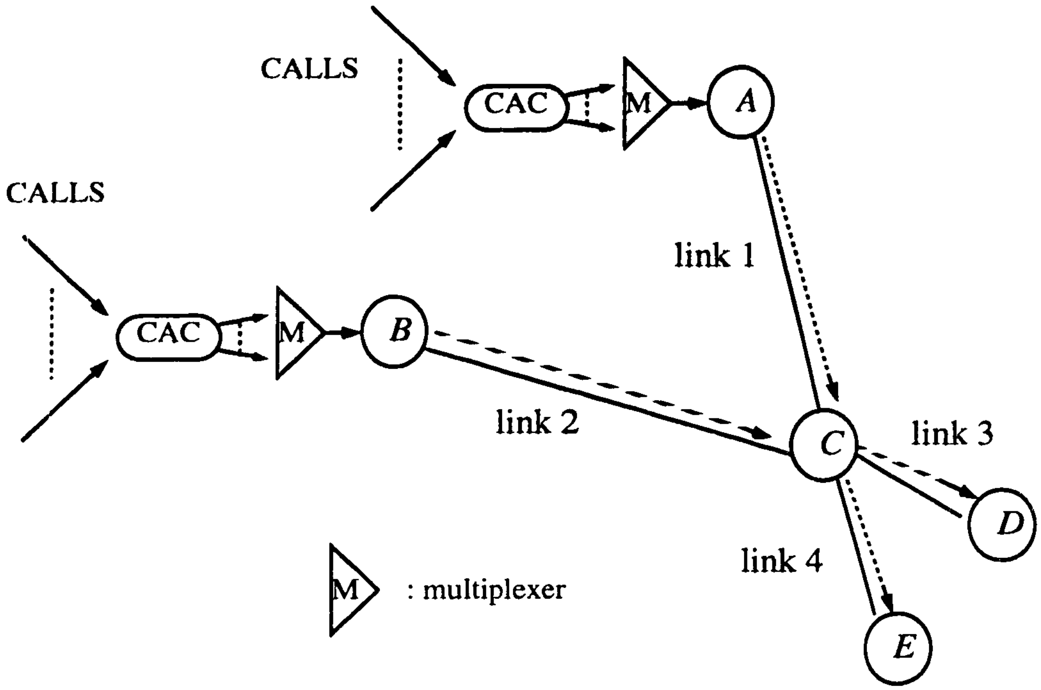


Figure 5.1: Network model

The destination of the call must be indicated by the call request. Since there is only one intermediate node, the route of the call is fixed when the destination is indicated. The traffic from source nodes *A* and *B* can reach destination nodes *D* or *E* via the intermediate node *C*. Therefore, we can study the interaction between traffic flows from source nodes *A* and *B*. The simulation models are usually simplified due to the limit of computer processing power. Many CAC studies only use a network model with one access node and one link for simulation [5, 8, 28, 31]. Therefore, the interaction of traffic flows cannot be simulated.

This network model is a complete network model, though it is small compared with reality. The performance of Peak CAC, Boyer CAC and Refinement CAC is examined using this network model.

5.3.2 Traffic Source Models

Six source models are implemented for the simulation. These source models can be used for data, voice and video traffic. For an initial study of the Refinement CAC scheme, three classes are defined. They are data, voice and video.

Classes	Models	Models
Data	2-state MMPP	2-state MMPP
Voice	IPP	On-Off fluid model
Video	AR model	Subband coder

Table 5.1: Implemented source models

The source models and their corresponding classes are listed in Table 5.1. We use five different source models to simulate the diverse traffic in ATM networks. These source models are very popular in the literature. Please see Chapter 2 for the detailed description of these source models. Peak CAC and Boyer CAC do not need classes. However, the traffic in Peak CAC and Boyer CAC are the same as Refinement CAC for the purpose of comparison.

5.3.3 Call Information

New calls arrive randomly in the simulations, and are admitted or rejected by CAC. In the simulation, several parameters of a new call are generated upon call arrival, although the user only declares peak rate (in Peak CAC and Boyer CAC) or class (in Refinement CAC) of his/her call. The generated parameters of a call are listed as follows:

- Destination. From the topology of the network model in Figure 5.1, there are two destinations which are node *D* and *E*. It is obvious that the route is determined by the destination. The destination is generated by Bernoulli trial.
- Call type. Each call type contains the static information of a call: class, peak bit rate (in Peak CAC and Boyer CAC) or default rate of the class (in Refinement CAC), source model, parameters of the source. Six call types are used in the simulation to represent the six sources (two sources in each of the three classes). Each call type is chosen with a probability.
- Call interarrival time. This is generated according to an exponential distribution.
- Call duration. This is used to schedule the call release event and generated according to an exponential distribution.

If the call is admitted, it starts to generate and transmit cells randomly according to its source model which is defined by its call type.

5.3.4 Assumptions on Call Setup and Release

We must simulate several types of traffic and a network with five nodes. Due to the limitation of computer power and complexity of the problem, we need to make some reasonable assumptions of the system to simplify the simulation. This section discusses the assumptions made in the simulations.

The call admission control, connection establishment and release are performed on the control plane in B-ISDN. Its control signalling protocol is an extension to the D-channel signalling protocol in N-ISDN [42]. In N-ISDN, the common channel signalling system #7 is used to establish connections. B-ISDN also uses dedicated signalling channels for call and connection control, though the signalling protocols have not been fully defined by CCITT [42]. Therefore, the control packets are transported only in the signalling channel. They do not consume the bandwidth of the data network.

Since our goal is to study the CAC scheme, we must simulate the signalling mechanism of the call setup and release. The call setup and release procedures are described in Section 5.2 (Peak CAC and Boyer CAC) and Section 4.3 (Refinement CAC).

Call Setup

We assume that the propagation delay of links between the same nodes of the data network and signalling network is the same. This is reasonable since the links have approximately the same length. We also assume that the transmission rate of the signalling network is the same as the data network and the processing time of the reserve cell is negligible.

The next assumption is that the call arrival rate is very low compared to the transmission rate and processing speed for the reserve cell. Therefore, we do not need waiting queues for the signalling network. The reserve cell is sent to the signalling channel immediately after the call request arrives. We have some reasons for that.

- The transmission rate of optical fiber channel is very high and processors are becoming more powerful.
- The signalling network is not the main consideration. We focused on the performance of the data network under the CAC scheme.
- We have propagation delay for the reserve cell though we do not have queueing delay for it. This allows the interaction between the two source nodes at the intermediate node.
- With the assumption, the reserve cells at the source nodes are transmitted and processed in the same order as the corresponding call requests. This still maintains fairness to the call requests.

In the case of simulation implementation, the CAC decisions are made at the intermediate node *C* upon arrival of the reserve cell. If the call is accepted, it waits a propagation delay for the reserve cell to travel back to the source node, then starts to transmit cells. Please note that call admission decisions cannot be made immediately upon the call arrival at the source node because the free bandwidth at the intermediate node may be updated by other events such as call release after the call arrival.

Call Release

When the call is accepted, the duration of the call is generated according to an exponential distribution and the call release time is set. However, the call release time is not fixed. It is used only to tag the last cell of the call. When the destination node receives the tagged cell, the call of the tagged cell is released immediately. If the tagged cell is lost somewhere due to buffer overflow, the second last cell of the call which has not reached destination is tagged. If it cannot be found, the call of the tagged cell is terminated immediately.

The actual signalling procedure of call release is not simulated. The call release procedure should have the release request from one end and the acknowledgment from the other end. This results in a round trip delay. So the assumption ignores the delay, but this is reasonable since the call duration is much longer than the round trip delay.

5.3.5 Performance Metrics

In order to compare the two CAC's, we need to define a set of performance metrics. The link utilization, call blocking probability, and QOS of calls are the important metrics since the objective of a CAC is to make the full use of the network resources or accept the maximum number of calls while maintaining QOS of all the accepted calls. The QOS of calls includes mainly cell loss probability, cell delay and delay variance. Next is a discussion of the metrics.

- Link utilization. Link bandwidth is the most important resource in ATM networks. In our network model, there are four links. We use the average of the four link utilizations to represent the link utilization.
- CBP (Call blocking probability). Calls are separated into three classes (see Section 5.3.2). So there are three CBPs for them. However, this is not enough to represent the total blocked traffic since the bit rates for the classes are different. Therefore, we define TBP.

- TBP (Traffic blocking probability). This is a combined metric of the CBP's. It considers the rates of different calls because the load offered by a call really depends on its rate. Suppose the mean bit rate of a call in class i is λ_i and the CBP of class i is CBP_i. Let $\lambda_{all} = \sum_i \lambda_i$. Then, the TBP is calculated with following equation,

$$TBP = \sum_i \frac{\lambda_i}{\lambda_{all}} \cdot CBP_i \qquad \text{(EQ 5.2)}$$

where the value of $\frac{\lambda_i}{\lambda_{all}}$ is the traffic weight of class i in terms of the mean bit rate.

- Mean queueing delay. This is the mean delay for a cell waiting in the buffers.
- Maximum cell delay and delay variance. This is the maximum delay endured by a cell to reach its destination from its generation. The end-to-end delay variance is also calculated.
- Buffer. Infinite buffers are used. However, the maximum buffer occupancy is measured so that we have an idea how large the buffers should be.

We do not measure the cell loss probability though it is also a QOS requirement like cell delay. This is because the cell loss probability requirement is 10^{-9} and cannot be simulated with current computer processing power unless some special statistical techniques are used to capture rare events. Therefore, we set the buffer size to infinity and make the cell loss equal to zero. However, this does not mean that it is completely ignored because we also measure the maximum buffer size. If the buffer size is very small, we can safely claim that no cell loss occurs. If it is very large, then large memory is needed and it may not be realistic. Furthermore, large buffer size often results in large cell delay. The buffer size will be discussed in Chapter 7.

5.4 Description of Events

Discrete event simulation is used in this thesis. Events occur at separate points in time. An event queue which queues the events by increasing time order is used to manage the occurrence of events. Future events are inserted into the event queue according to their time of occurrence. After an event is finished, the first event in the event queue is taken out and executed. The time of simulation is updated to be the current event time. The events of the simulation are listed in Table 5.2. Usually some future events will be arranged in the execution of one event.

Event	Function
CaR (A)	Call request at node A
CaR (B)	Call request at node B
CaG (A)	Call admission or rejection for node A call requests
CaG (B)	Call admission or rejection for node B call requests
CeA (A)	Cell generation at node A
CeA (B)	Cell generation at node B
CeA (D)	Cell arrival at node C for destination of node D
CeA (E)	Cell arrival at node C for destination of node E
CeD (D)	Cell departure from node C for destination of node D. If it is the last cell, release the call.
CeD (E)	Cell departure from node C for destination of node E. If it is the last cell, release the call.
Refine	Refinement of current call (for Refinement CAC)
Mea (A)	Measurement at node A (for Refinement & Boyer CAC's)
Mea (B)	Measurement at node B (for Refinement & Boyer CAC's)

Table 5.2: The list of events

The relation of the events for Refinement CAC is exhibited in Figure 5.2. Each round corner rectangular represents an event. The figure shows how the simulation starts from source node A. The event order from source node B is the same if we replace node A with B. Each call starts from a call request event CaR (A). Then it arranges the call admission event CaG (A) and the next call arrival event CaR (A). If the call is admitted, it starts to generate cells by arranging the next cell arrival event CeA (A). If a cell arrives at node A, the next cell arrival event is scheduled after a cell interarrival time according to the call's source type. The cell travels through the network until it reaches the destination node. The call is released when its last cell arrives at the destination. The simulation stops and no event will be executed when the simulation time moves to end.

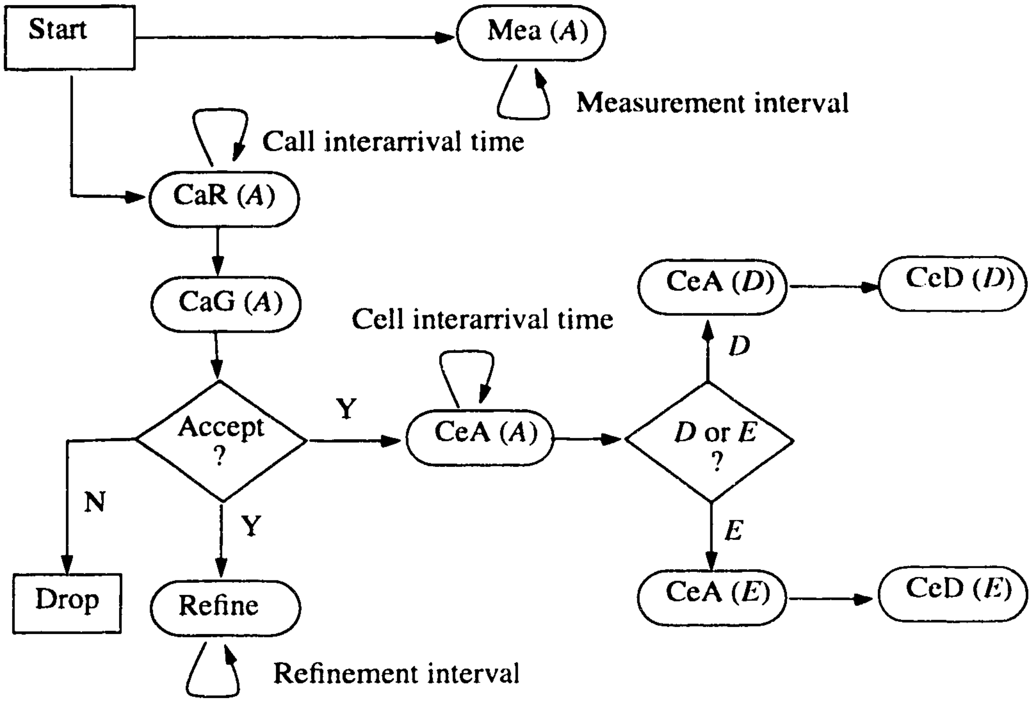


Figure 5.2: Relation of events

We can obtain the relation of events for Peak CAC if we delete the events Mea (*A*) and Refine in Figure 5.2. The relation of events for Boyer CAC can be obtained by deleting the event Refine in the figure.

5.5 Random Number Generator and Variance Reduction Technique

The random number generator used in the simulations is the function "random()" in the C library. It generates uniform U(0,1) random numbers. All the other random variates such as exponential numbers are generated from U(0,1). Its description can be found by typing "man 3 random" in UNIX operating system on a workstation. Its period is greater than 2^{69} or 5.9×10^{20} which is sufficient for the simulations in this thesis.

A VRT (variance reduction technique) called CRN (common random numbers) is used to raise the efficiency of the simulations. With VRT, we can obtain smaller confidence intervals for the same simulation cost. CRN is considered to be the most useful VRT for comparing two or more alternative system configurations [44]. CRN uses the same random variates for the different system configurations so that the experimental conditions are similar for alternative configurations (in the thesis, Peak CAC and Refinement CAC).

In order to implement CRN, the random numbers of the Peak CAC and Refinement CAC should be matched or synchronized. That is, each random number used by Peak CAC must be used for exactly the same purpose by Refinement CAC. For instance, the random number used for the 100th call duration by Peak CAC must be used for the 100th call duration by Refinement CAC.

We separate the random numbers in the CAC simulations into two groups: one for call random numbers, the other for cell random numbers from source models. It is obvious that the cell random numbers cannot achieve synchronization because the two CACs will accept and reject different calls. We know that different calls use different source models. Therefore, the cell random numbers cannot be matched for the two CACs.

The call random numbers can be synchronized. As described in Section 5.3.3, each call has four parameters which are randomly generated. They are generated in call request arrival event and in the same order for both CACs. Furthermore, we use two streams. One stream is used exclusively for call random numbers, the other for cell random numbers. In this way, the call random numbers are synchronized and CRN is achieved for call variates.

5.6 Simulation Control

The simulation control parameters include the simulation stop time (*TStop*) and transient time (*Trans_time*). The simulation starts from time 0. *TStop* is set to 200 seconds. It is enough to obtain good confidence intervals of the results. Transient time, the time when the system enters steady state, is also found. The simulation measures the performance metrics after the transient time. Transient time can be obtained by observing the results of some important metrics with increasing time. If the values of the metric become steady after some time, this time is set to be the transient time.

Several observations are made with simulations of different configurations to set the transient time. Figure 5.3 shows some of the observations of average server utilization. It is obvious that *Trans_time* can be set to 30 seconds. Each simulation is repeated eight times to obtain the confidence interval.

5.7 Confidence Interval

Confidence intervals are used in the results analysis of this thesis. We give an introduction in this section. It is referenced from [50].

Suppose we have performed an experiment of m runs and collected a sample of m observations. Let these observations be $X_1, X_2, \ldots, X_m$. Here, X_i is the mean waiting time collected in the ith run. Then the sample mean is:

$$\bar{X} = \sum_{i=1}^{m} X_i / m \qquad \text{(EQ 5.3)}$$

$\bar{X}$ is an estimate of the population mean μ. The sample variance is:

$$S^2 = \left[\sum_{i=1}^{m} X_i^2 - m\bar{X}^2 \right] / (m-1) \qquad \text{(EQ 5.4)}$$

S^2 is an estimate of the population variance δ^2. In discrete event simulation, μ and δ^2 are not known. If the length of run is long enough, we can argue that X_i has a symmetrical distribution. The distribution of $\bar{X} - \frac{\mu}{S/\sqrt{m}}$ can be approximated by a t-distribution with $m-1$ degrees of freedom.

The 95% confidence interval is given by $\bar{X} \pm t_{0.975, m-1} S/\sqrt{m}$. It means that the population mean is within the confidence interval with 0.95 probability.

5.8 Simulation Validation

The simulation is verified by tracing the events and comparing with theoretical results. Tracing is used to validate the event execution in the simulation. A short simulation is run to print out the timestamps of the events. Each timestamp may include the event name, time, node, call ID and cell generation time, etc. Then we check the event sequence to see if they are following the correct order and also check performance metrics.

The source models are verified by comparing the simulation results with theoretical results. The mean cell interarrival time is often used for comparison. The simulation results and theoretical results match very well for the source models.

The validity of Peak CAC is also tested by comparing its results to theoretical results. The lower bound of call blocking probability (CBP) can be obtained by queueing analysis for Peak CAC if identical constant bit rate sources are used. The system can be treated as an M/M/m queue without buffer if we only consider one source node. Let the bandwidth requirement of each call be R and the link capacity be C. Then the number of server $m = C/R$. Let the mean call duration be D. Then the service rate of a call is $\mu = 1/D$. Let $\rho = \lambda/\mu$ where λ is the call arrival rate and the CBP is given by the following equation according to the M/M/m formula [52].

$$CBP = \frac{\rho^m/m!}{\sum_{i=0}^{m} \rho^i/i!} \qquad \text{(EQ 5.5)}$$

However, there are two source nodes in the network and the traffic flows from the two soruce nodes interact at the intermediate node. The CBP from simulation is higher than the theorectical one because the intermediate node may block some more calls. Therefore, the CBP obtained by (EQ 5.5) is the lower bound. The link capacity is set to be 50 Mbps. Mean call duration is 4 seconds and the source bit rate is 1 Mbps. The call interarrival time ranges from 60 ms to 120 ms. Figure 5.4 shows the simulation results and the lower bounds of CBP. The CBP's from simulation are higher than their lower bounds. This demonstrates that the simulation of Peak CAC is credible for identical sources.

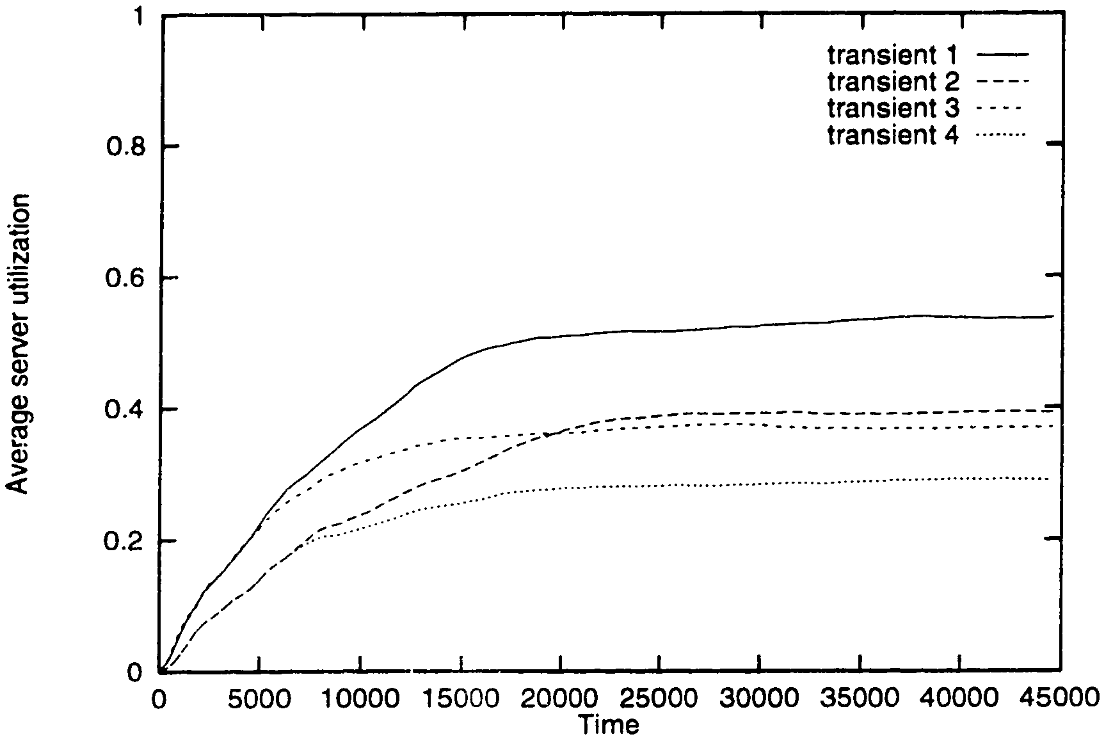


Figure 5.3: Average server utilization for transient time

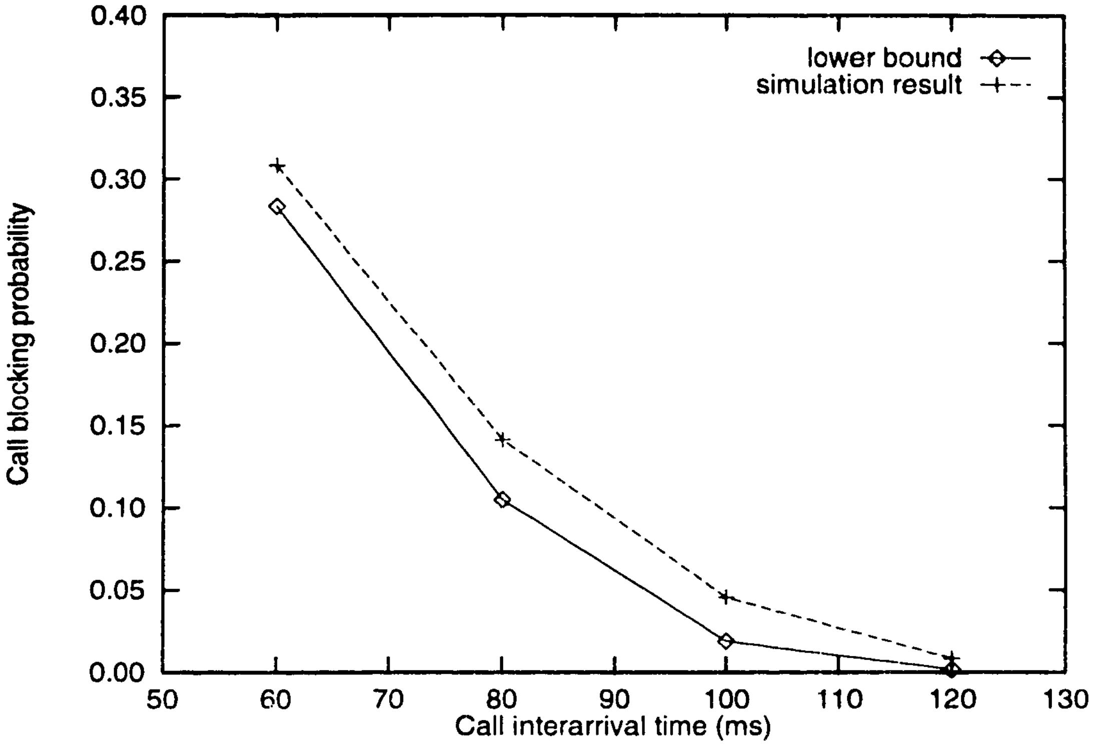


Figure 5.4: Comparison of simulation and theoretical results

CHAPTER 6 Parameters of Simulation

6.1 Introduction

In this chapter we discuss setting the values of the parameters. There are a large number of parameters that must be set in the simulations. This makes the study and analysis of the system complex. We set the parameters by using information from the literature, inspection of experimental runs and reasonable judgement.

We separate the parameters into several groups: network model, call information, call types, refinement and measurement. Each group of parameters will be discussed in a section. The parameters for the simulation control are discussed in the previous chapter. Because the three CACs are compared under the same conditions, the values of some parameters are the same for all the CACs. For example, the parameters of the network model should be the same. The values of parameters for a specific CAC will be indicated. Otherwise, they are the same for all the CACs.

6.2 Setting Parameters of Network Model

The parameters that describe the network model correspond to link length, link capacity, buffer size and propagation time. They are listed in Table 6.1. The parameter "link 1 length" is the length of link 1. The link positions are shown in Figure 5.1.

Parameters	Values	Parameters	Values
Link 1 length	160 km	Link 2 length	160 km
Link 3 length	200 km	Link 4 length	200 km
Buffer size	∞	Link Capacity	150.336 Mbps
Propagation time	0.005 ms/km		

Table 6.1: Parameters for network model

The parameter setting of the network is symmetric. The first hop from each source node is 160 km. The second hop is 200 km. Therefore, the distance for each route is 360 km. Each hop corresponds to the distance between two cities. The propagation time for the link is referenced from [9]. The link capacity for each link is 150.336 Mbps which is the STS-3 payload data rate in SONET standard or the STM-1 payload data rate in CCITT standard [45].

The buffer size at each link is set to infinity since we do not measure the cell loss probability in this case. However, the maximum buffer occupancy is recorded in the simulations. In this way, we can determine how large the buffer size should be and make sure that the size is reasonable in practice.

6.3 Setting Parameters of Call Information

The call information parameters include destination, call type, call arrival rate, call duration. The probability of a call going to either of the two destinations are set to 0.5. The call arrival rate is used to vary the load of the system. It is set to be 20, 22.7 and 25 calls/second in the experiments. The load of the system increases as mean call arrival rate increases. The mean call duration of all the calls is set to 4 seconds. This is obviously short for most applications. Since a large number of cells must be generated, simulation times have to be limited. Also, the desire is to study the effect of having a mix of calls arriving and departing. Other studies have based results on even smaller call durations. The call duration is set to 10 ms in [5], and set to 1500 ms and 800 ms in [28].

Six call types corresponding to two call types per class are defined as described in Section 5.3.3. There are two alternatives for generating a call type. The first is to assume that the traffic competes equally for access to the network. This would imply equal arrival rates for the call types. This may be reasonable since we do not know what demand there will be for new applications. However, due to the wide range of bandwidth requirements, it will result in high blocking probability for certain traffic. The second is to assume that the network could be dimensioned to give low call blocking probability. Therefore, we should study the system with low call blocking probability for all traffic. We considered the second one. In particular, we attempted to adjust the arrival rates for the call types in order to get approximately equal call blocking probabilities. This proved too complex since there were three classes of traffic. The probabilities for generating a call type in data, voice, video classes are set to 0.36, 0.48, 0.16, respectively.

6.4 Setting Parameters of Call Types

This section introduces the values of parameters for call types or source models. In the simulations, each of the three classes contains two call types. In the following discussions about parameters, the peak rate is only used for Peak CAC and Boyer CAC, and the default rate of a class is only used for Refinement CAC.

6.4.1 Call Types of Data Class

The two call types of data class correspond to 2-state MMPP source models with different values of parameters. The values of the parameters for the data class are listed in Table 6.4. The 2-state MMPP model is used for an aggregate source. It has two states and each state has a mean rate and mean duration. The default rate of the data class is set to be slightly higher than the average of the mean rates of the two sources. The mean rates of the two sources are 757 kbps and 735 kbps. The source with higher mean rate is more bursty. In a ?-state MMPP, a Poisson process is used for cell generation in each state. We use 1.2 times the mean rate of the high rate state (state 1) as its peak rate. Appendix A shows how the peak rate of Poisson process is obtained.

Source Models	Mean rate (state 1)	Duration (state 1)	Mean rate (state 2)	Duration (state 2)	Peak rate	Default rate
2-MMPP	1060 kbps	130 ms	321 kbps	90 ms	1272 kbps	750 kbps
2-MMPP	922 kbps	150 ms	385 kbps	80 ms	1106 kbps	750 kbps

Table 6.2: Parameters for data traffic

6.4.2 Call Types of Voice Class

The voice class contains an IPP source model and an On-Off fluid model. The two source models are used for single voice sources and have similar behavior. The values of the parameters for voice class are listed in Table 6.3. Both models have an active state and an idle state where the duration of each state is exponentially distributed. Each state has a mean rate and mean duration and the mean rate of the idle state is zero.

Source Models	Mean rate (active)	Duration (active)	Mean rate (idle)	Duration (idle)	Peak rate	Default rate
IPP	64 kbps	352 ms	0 kbps	650 ms	76.8 kbps	22.5 kbps
On-Off	64 kbps	352 ms	0 kbps	650 ms	64.0 kbps	22.5 kbps

Table 6.3: Parameters for voice traffic

For IPP model, the peak rate is set according to Appendix A because a Poisson process is used in the active state. For the On-Off fluid model, the constant bit rate in an active state is used for peak rate. The mean bit rate in an active period is usually set to be 64 kbps [10, 31]. The default rate of voice is set to be the average of the mean rates of the two sources. The mean durations of active and idle states are usually set to be 352 ms and 650 ms, respectively [31, 8, 11].

6.4.3 Call Types of Video Class

The video class contains the AR source model and subband coder source model. These two source models are used for single video sources.

Parameters	Values	Parameters	Values
a	0.8781	b	0.1108
Mean of w(n)	0.5720	Variance of w(n)	1

Table 6.4: Parameters for AR model of video traffic

Parameters	Values	Parameters	Values
Frame rate	30 frames/s	Pixels per frame	250000
Mean $E(\lambda)$	0.52 bits/pixel	Mean rate	4306 kbps
Default rate	4370 kbps	Peak rate	11042 kbps

Table 6.4: Parameters for AR model of video traffic

The values of the parameters for the AR source model are listed in Table 6.4. They are referenced from [17]. Please see Section 2.4.1.1 for the explanation of the parameters. The default rate of video is slightly higher than the mean rate of the two sources. This is to demonstrate that the default rate is flexible. Note that the peak rate is much higher than the default rate. All the bits generated in one frame are transmitted at a constant bit rate of 10 Mbps. The bit rate is adjusted to include the header bits because an ATM cell contains 5 bytes header in 53 bytes cell length. Therefore, the actual peak rate is 11042 kbps.

Parameters	Values	Parameters	Values
Frame length F	62.5 ms	Bit rate c	10000 kbps
A_{min}	10 ms	A_{max}	40 ms
Default rate	4370 kbps	Peak rate	11042 kbps
Mean rate	4417 kbps		

Table 6.5: Parameters for the model of subband coder

The values of the parameters for the model of subband coder are listed in Table 6.5. They are referenced from [31] except the default rate and peak rate. Please see Section 2.4.1.3 for the explanation of the parameters. The bit rate of each frame is c = 10 Mbps. Therefore, the actual peak rate is adjusted to 11042 kbps due to the 5 bytes of cell header.

6.5 Setting Parameters of Refinement Measurement

In addition to the parameters introduced in the previous sections, more parameters need to be set for refinement and measurement. The parameters for refinement include refinement intervals and thresholds for the three refined subclasses of each class in the Refinement CAC. The values of these parameters are listed in Table 6.6. The low threshold and high threshold are bit rate levels for separating the three refined subclasses. The refine interval in the last row is the constant interval of refinement.

The values of the parameters are obtained by studying the bursty characteristics of the traffic sources in each class. The characteristics of data class sources are shown in Figure 6.1 and Figure 6.2. The voice class sources are shown in Figure 6.3 and Figure 6.4. The video class sources are shown in Figure 6.5 and Figure 6.6. The measurement intervals in the figures are the refinement intervals for recording the moving average bit rate. The moving average bit rate of a call is calculated over the last 5 intervals. The figures show the points from the 10th to 50th or 100th measurement. For example, the point of 30 along the x-axis in Figure 6.1 represents 30th measured bit rate or the bit rate at 30x200 ms because the refinement interval is 200 ms, and the bit rate is calculated over 5x200 ms.

Parameters	Data class	Voice class	Video class
Low threshold	722 kbps	12.5 kbps	3540 kbps
High threshold	772 kbps	32.5 kbps	5200 kbps
Refine interval	200 ms	100 ms	120 ms

Table 6.6: Parameters for refinement

Several runs were made with different refinement intervals. If the refinement interval is too long, the measured bit rate of a call tends toward the mean rate. If it is too short, the long term behavior of a call is lost and the call jumps among the subclasses. The values of the thresholds are determined so that the calls can transfer among the refined subclasses and stay in a refined subclass for a while. The value for high or low threshold is set by adding or deducting the same amount of rate from the mean rate, respectively.

The fraction of link capacity γ in Refinement CAC and Boyer CAC is set to be 0.8. Several runs were also made with different measurement intervals. If the measurement interval is too long, the measurement may not be up-to-date. If it is too short, the measured bandwidth may have less relation to the refined calls. The measurement interval in Refinement CAC and Boyer CAC is set to be 400 ms.

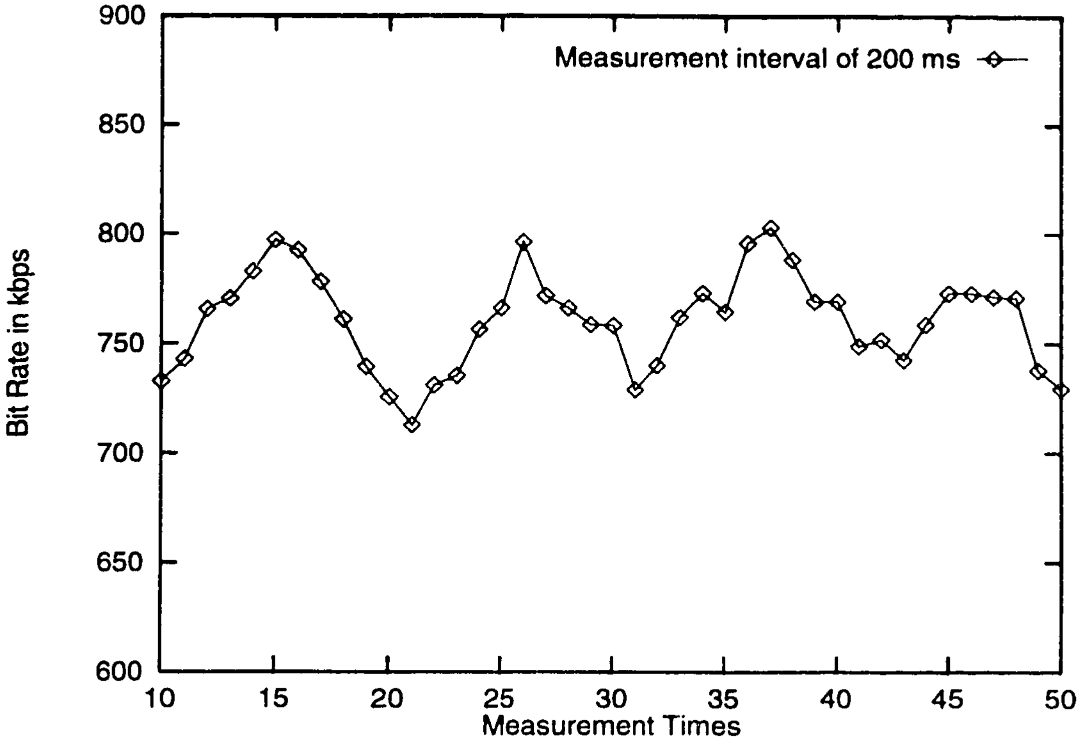


Figure 6.1: Characteristics of the first MMPP source

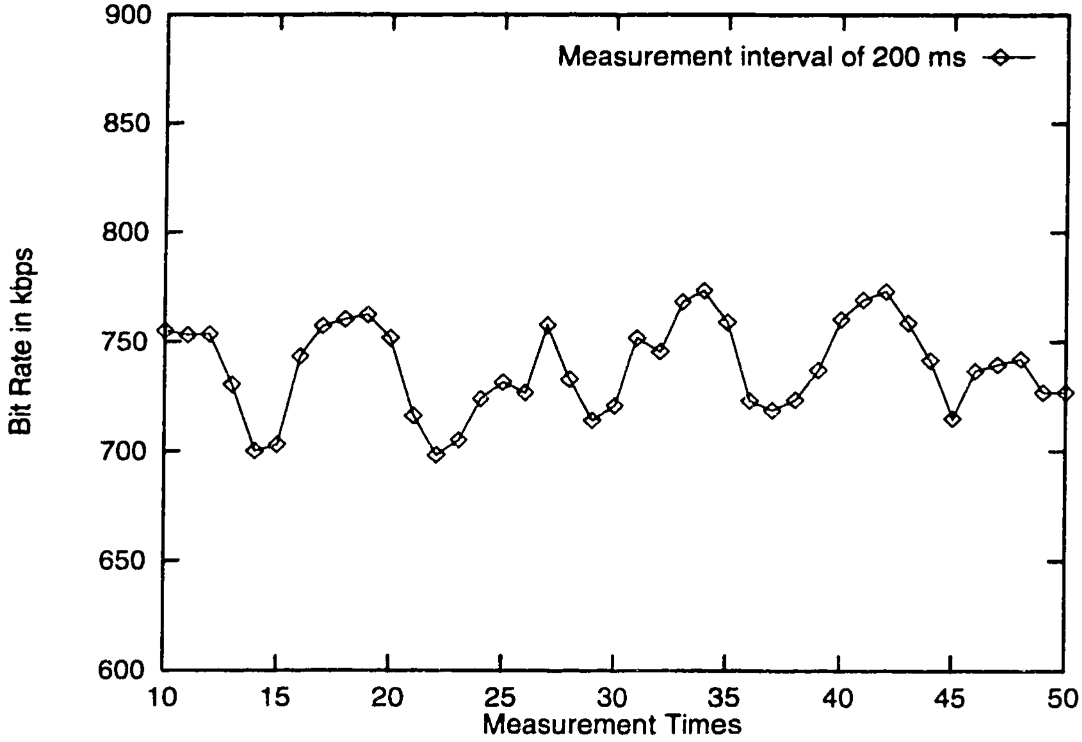


Figure 6.2: Characteristics of the second MMPP source

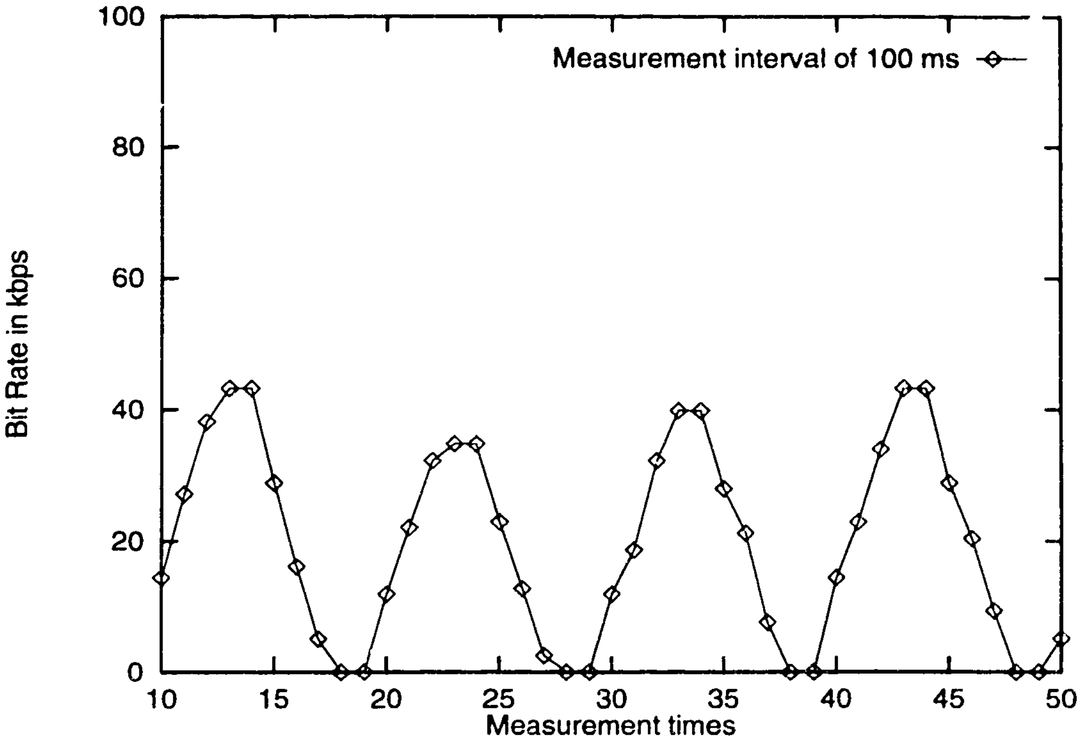


Figure 6.3: Characteristics of IPP source

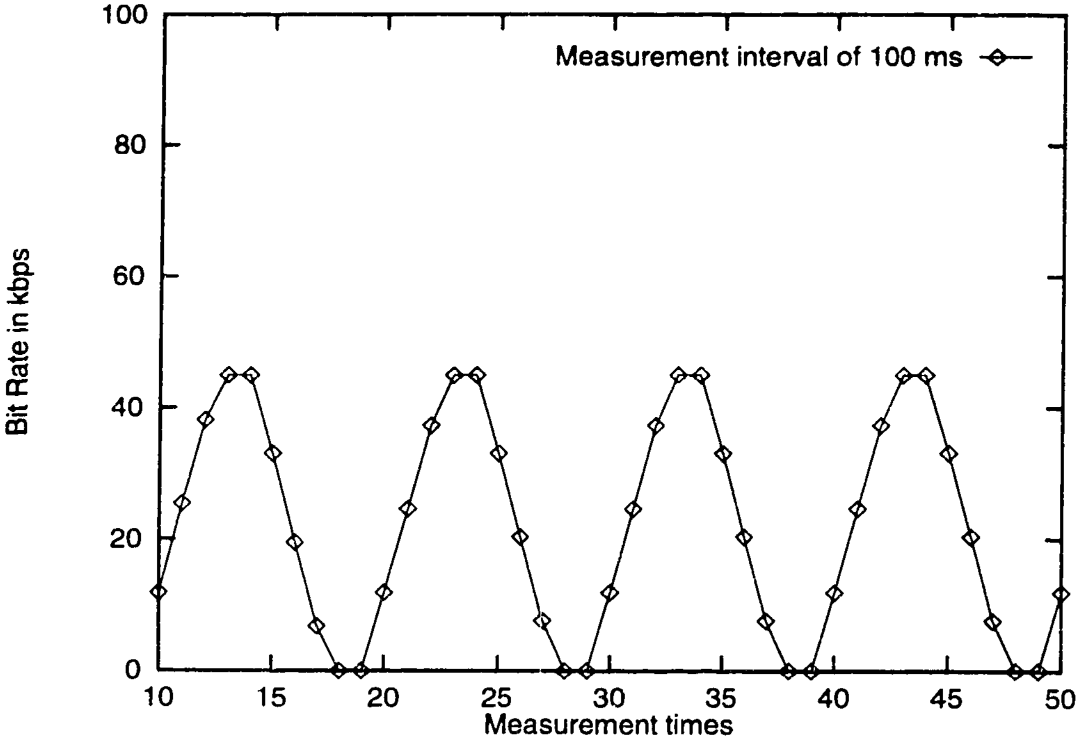


Figure 6.4: Characteristics of On-Off fluid source

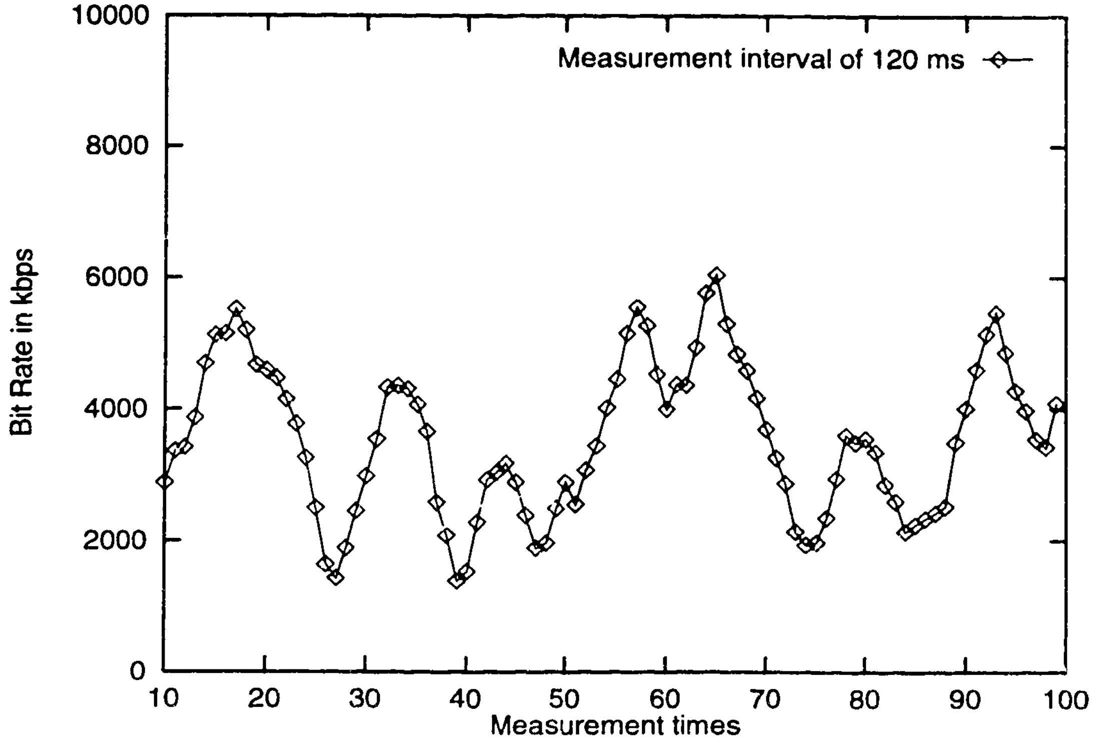


Figure 6.5: Characteristics of AR source

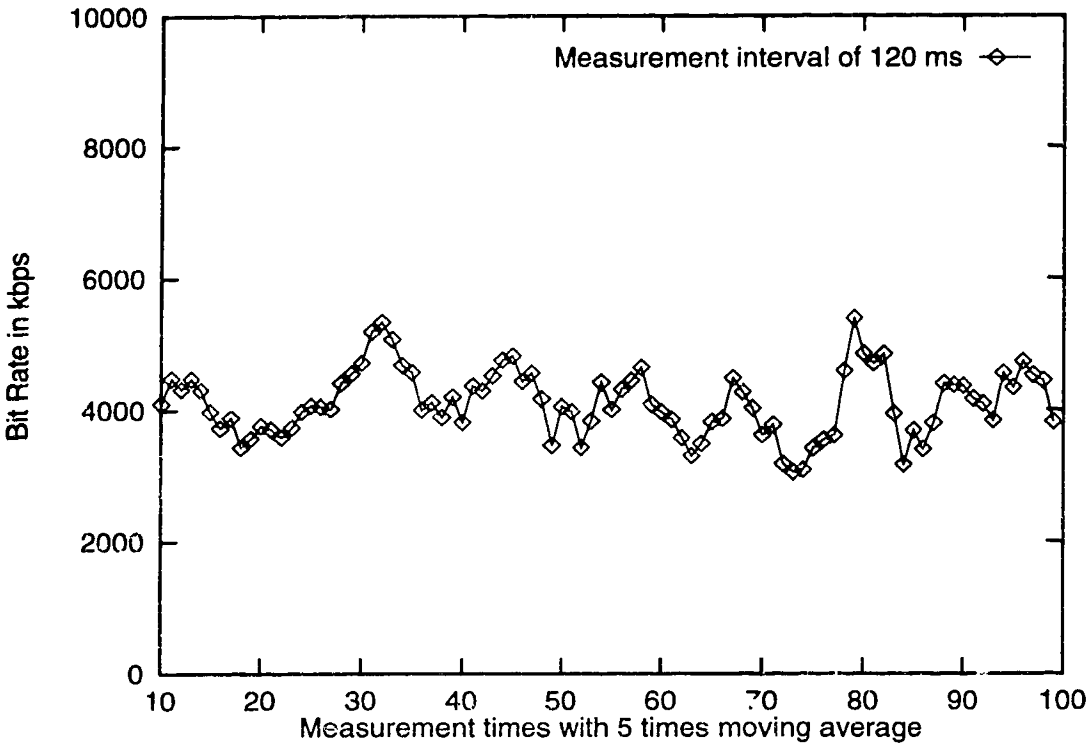


Figure 6.6: Characteristics of Subband source

CHAPTER 7 Results Analysis

7.1 Introduction

The simulation results and analysis are presented in this chapter. The performance of the proposed Refinement CAC is compared with Peak CAC and Boyer CAC. Several measures such as link utilization, cell delay, and traffic blocking probability (TBP) are studied.

The results of the performance of the CAC's are shown in Figures 7.1 to Figure 7.8. We consider the performance of the CAC schemes over a range of loads offered to the system. The call arrival rate is used to vary the load. Each point in a figure corresponds to 8 runs of the simulation and lines between points are used simply to clarify the presentation. The vertical error bars at these points represent 95% confidence intervals.

7.2 Results and Performance

As shown in Figure 7.1, the average link utilization of Refinement CAC's (R_CAC_Cons and R_CAC_Aggr) and Boyer CAC is much higher than that of Peak CAC. The utilization of Boyer CAC is 6.4% (or 0.04) lower than that of Refinement CAC's when call arrival rate is 25 calls/second. The Refinement CAC's accept more calls than Boyer CAC and Peak CAC because the default rate used for call admission in Refinement CAC's is less than peak rate. Therefore, the utilization is higher for Refinement CAC's. The utilization of Boyer CAC is higher than Peak CAC because more free bandwidth is extracted by measurement. The link utilization increases with offered load for all the CAC's because more calls may be accepted. There is no significant difference in utilization between the two Refinement CAC's. However, the difference becomes larger as the load increases.

The offered load is the expected utilization if no call is rejected. The expected utilization can be obtained by (EQ 7.1):

$$Util = \frac{\lambda \cdot D \cdot (P_{data} \cdot R_{data} + P_{voice} \cdot R_{voice} + P_{video} \cdot R_{video})}{C_{link}} \qquad \text{(EQ 7.1)}$$

where P is the call arrival probability of a class, R is the mean bit rate of a class, λ is the call arrival rate, D is the mean duration of a call and C_{link} is the link capacity. Therefore, the expected utilization is 0.52, 0.59, 0.65 for the call arrival rate of 20, 22.7, 25 calls/second, respectively. The difference of the expected utilization and the utilization of Refinement CAC is approximately 6% (or 0.04) at high load. It is even smaller at lower load. This can demonstrate that the blocked traffic is small in the Refinement CAC.

The worst case utilization (or maximum utilization) is reached when all calls are transmitting cells at peak rate if no call is rejected. It can also be obtained by (EQ 7.1) if we replace mean bit rate R with peak bit rate for each class in the equation. The worst case utilization is 1.18, 1.34, 1.48 for the call arrival rate of 20, 22.7, 25 calls/second, respectively. It is higher than the link capacity. Therefore, congestion will occur if all calls are accepted and all sources transmit cells at peak rate.

Since the transmission delay and the propagation delay are fixed, the queueing delay is of interest. There is a trade-off between queueing delay and link utilization. Higher link utilization implies more cells in the system which results in longer queueing delay. Queueing delays occur due to the need to buffer cells when the link is overloaded.

The results of the mean queueing delay of cells is shown in Figure 7.2. The Peak CAC has very short queueing delays which are less than 0.002 ms because the link utilization of Peak CAC is very low. The mean queueing delay of Boyer CAC is longer than the Peak CAC and shorter than Refinement CAC. There is little difference in mean cell delay at low load for the two Refinement CAC's. The mean cell delay is close to 0.01 ms when call arrival rate is 20 calls/second. While at high load, there is more difference. The mean cell delay of conservative CAC is shorter and the aggressive CAC has the longest mean cell delay due to the highest link utilization. When call arrival rate equals to 25 calls/second, the delay of aggressive CAC is 25% (0.005 ms) longer than that of conservative one.

The mean cell delay increases with the offered load for all the CAC's because the link utilization increases too. The end-to-end cell delay can be obtained by adding the queueing delay to the minimum cell delay (1.80564 ms). The minimum cell delay is the cell delay which only includes transmission delay and propagation delay and does not include the queueing delay.

It is instructive to look at the maximum end-to-end cell delay in Figure 7.3. The maximum delays for Peak CAC are close to 2 ms and those for the other CAC's are less than 8 ms. They satisfy the QOS requirements of 250 ms channel delays for multimedia communications [48] and 25 ms delay requirement to avoid the use of echo compensation for voice channels [49]. The end-to-end delay variance for each class at high load (25 calls/second) is displayed in Table 7.1. The largest delay variance is 0.021 ms for video class of aggressive Refinement CAC. Thus, the standard deviation of end-to-end delay is 0.145 ms by taking the square root of delay variance. This can satisfy the delay jitter requirement of 1 ms for video.

Class	Peak CAC	Boyer CAC	R_CAC_Cons	R_CAC_Aggr
Data	0.000004 ms	0.002586 ms	0.008346 ms	0.011804 ms
Voice	0.000004 ms	0.002306 ms	0.007669 ms	0.011007 ms
Video	0.000004 ms	0.004853 ms	0.014893 ms	0.021004 ms

Table 7.1 End-to-end delay variance at high load (25 calls/second)

There is also a trade-off between buffer size and link utilization. The maximum buffer occupancy increases with the link utilization because more cells are queued. Table 7.2 is the maximum buffer occupancy for the CAC's. It can be used to determine the buffer size. In the case of Peak CAC, the maximum buffer occupancy is less than 0.022 Mbits which is very small. The largest buffer occupancy which is for aggressive Refinement CAC is 0.6178 Mbits and is still reasonable. The buffer occupancy of conservative Refinement CAC is lower than aggressive CAC. Thus, cell loss will occur for aggressive CAC with small buffer size such as 0.6 Mbits while there is no cell loss for conservative CAC. Since the utilization of the two Refinement CAC's is comparable, conservative CAC may be better than aggressive CAC if buffer size is small.

CAC name	Maximum buffer occupancy	Corresponding memory size
Peak CAC	50 cells	0.022 Mbits
Boyer CAC	1442 cells	0.612 Mbits
R_CAC_Cons	1276 cells	0.542 Mbits
R_CAC_Aggr	1862 cells	0.790 Mbits

Table 7.2 Maximum buffer occupancy for the CAC's

The results of traffic blocking probability (TBP)[1] are shown in Figure 7.4. We can find that the CAC scheme with higher link utilization has a lower TBP because it allows more traffic into the system. Figure 7.5 shows the relation between traffic blocking probability and link utilization of the CAC's at high load (25 calls/second). Refinement CAC's have higher link utilization and lower TBP's. The relations at the other loads are similar.

The call blocking probability (CBP) is different from TBP. We analyze the CBP's according to the classes because the calls have different bandwidth requirements for the three classes. The CBP results are shown in Figures 7.6 to 7.8. For data class, the CBP's of Refinement CAC's and Boyer CAC are approximately 44% (0.03) lower than those of Peak CAC. The CBP's for data are small and acceptable. For voice class, the CBP's of the CAC's at low load are comparable. At high load, the CBP's of Refinement CAC's are approximately 130% (0.007) higher than those of Peak CAC and Boyer CAC. The CBP's for voice are also small and acceptable. For video class, the CBP's for Peak CAC are 335% (0.3) higher than the Refinement CAC's. The CBP's for Boyer CAC are much lower than Peak CAC and higher than Refinement CAC's. The Refinement CAC's accept more video calls because of the low default rate. The conservative CAC blocks more video calls

[1] See Section 5.3.5 for the definition of TBP.

and accepts more voice calls than the aggressive CAC. The CBP's of video for Peak CAC seem unacceptable. Peak CAC accepts more voice calls at the expense of severely blocking video calls.

The Refinement CAC is a scheme that admits call requests based on a class default mean rate and uses measurement to correct for an inaccurate default mean rate. Note that, in our experiments, the default mean rates for data and video are not exactly the average of the source model mean rates. In addition to the use of default mean rate and measurement, we also consider refinement in order to have more control based on the current mix of calls. However, our results show that the conservative and aggressive schemes do not show much difference in performance at low load. As load increases there is more difference. Conservative CAC blocks more video calls and accepts more voice calls than the aggressive CAC and the aggressive CAC has higher mean delay.

There are a number of reasons for the lack of difference between the two Refinement CAC's. At low load the network can tolerate most of the calls. That is, there is less competition for resources. Calls are blocked when a number of sources are predominantly transmitting at high rates. Generally when there is a call in a high refined subclass, there could be another call in the low refined subclass and they "cancel" each other out to some extent. This reduces the difference that is found between the conservative and aggressive schemes. This is due in part to the symmetry of the source models and to the setting of the default mean rate to be close to the average of the source model mean rates. Although several traffic sources are multiplexed together, they are fairly uniform within a class.

Note that the accurate peak rate is used at call admission in the simulations for Boyer CAC and Peak CAC. However, users may not be able to provide an accurate peak rate [51]. If users over-estimate the peak rate, more bandwidth will be wasted and the performance will be worse. Peak CAC will have significantly poorer behavior. Boyer CAC will also have poorer performance. Although measurement could adjust the bandwidth usage to a more accurate result, the call admission is still based on the over-estimated peak rate and results in fewer accepted calls. If users under-estimate the peak rate, the CAC may not be reliable ?nd congestion may occur. Therefore, the performance of Boyer CAC and Peak CAC from the simulations may be better than reality.

Overall, the performance of the Refinement CAC's is better than Peak CAC and Boyer CAC because the link utilization for Refinement CAC's is much higher. The end-to-end cell delays of Refinement CAC's are longer but meet the QOS requirements. Large buffers are needed for Refinement CAC's but they are less than 1 Mbits.

7.3 Summary

The Refinement CAC scheme achieves significant improvement in channel utilization and TBP over Peak CAC and Boyer CAC schemes. Cell delays become longer with Refinement CAC's. However, this is not a problem if cell delays satisfy the QOS requirements. Large buffers are also needed for Refinement CAC schemes. The link utilization of aggressive Refinement CAC is slightly higher than the conservative one and the cell delays of aggressive Refinement CAC are longer than the conservative one at high load. The difference between the two Refinement CAC's is not obvious. However, refinement may be important in real ATM networks since statistical models for VBR video traffic may not fully represent the real traffic [51]. Further study is needed.

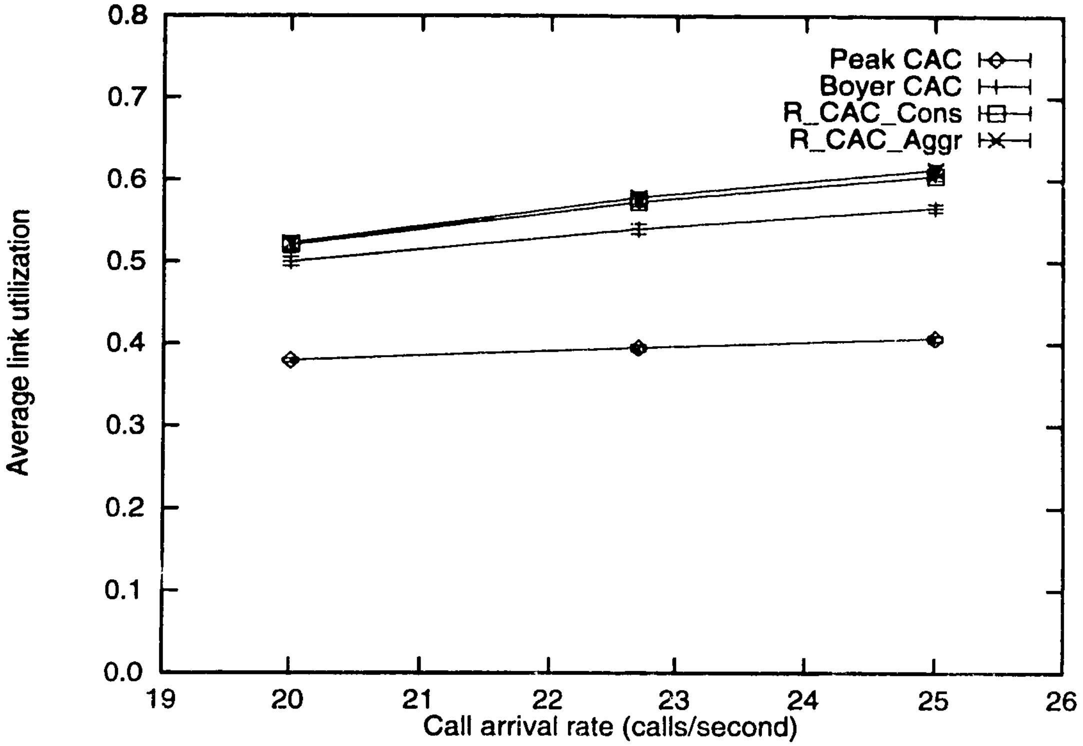


Figure 7.1: Comparison of average link utilization

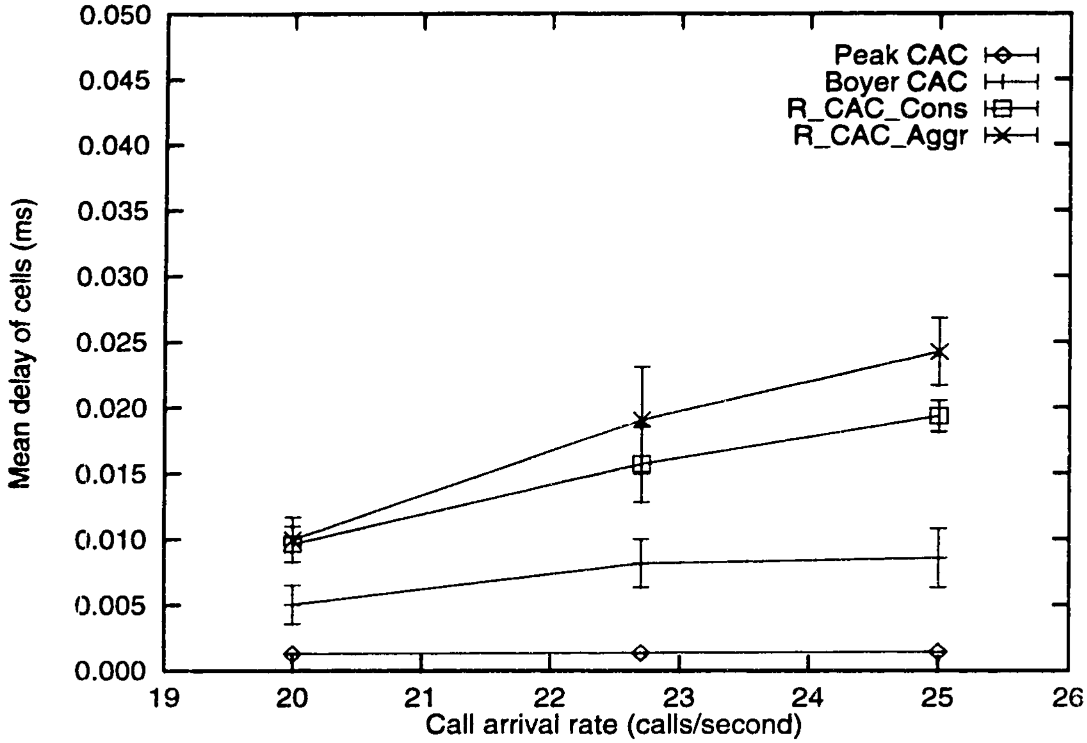


Figure 7.2: Comparison of mean queueing delay of cells

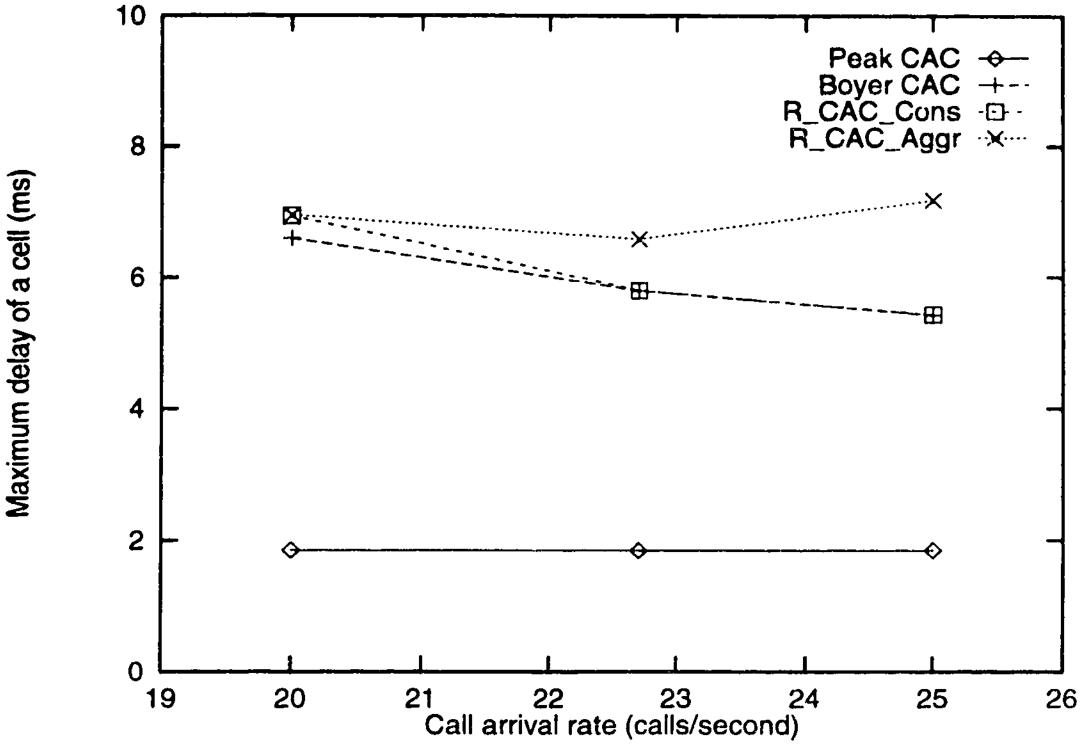


Figure 7.3: Comparison of maximum cell delay

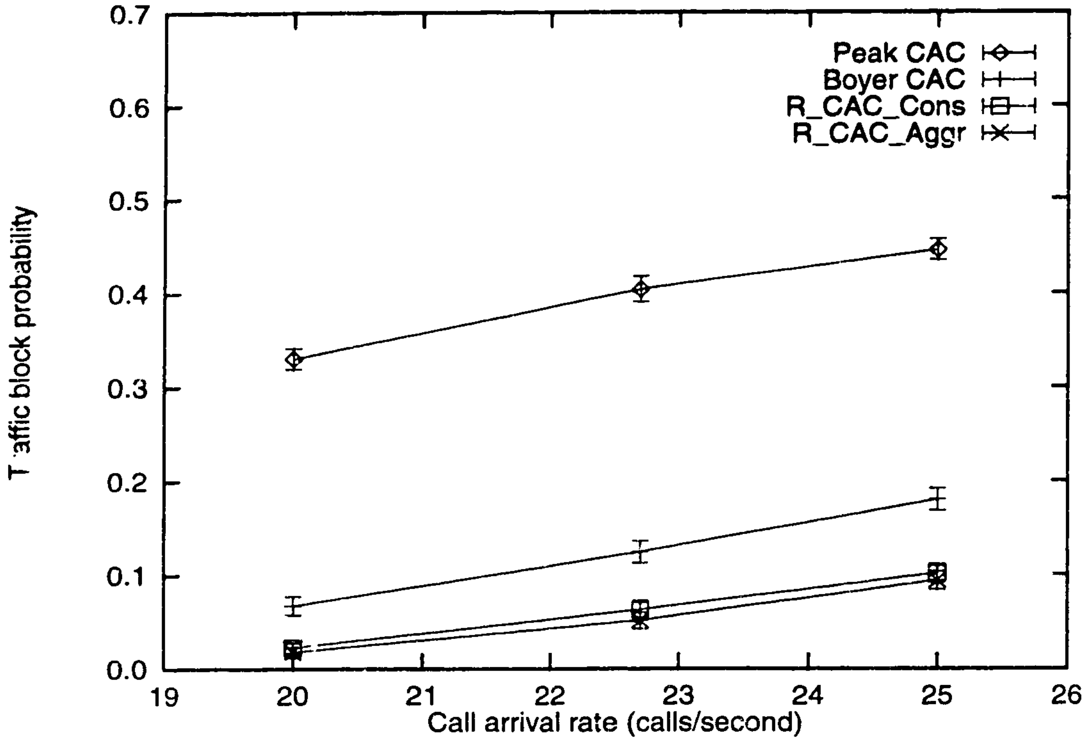


Figure 7.4: Comparison of traffic blocking probability

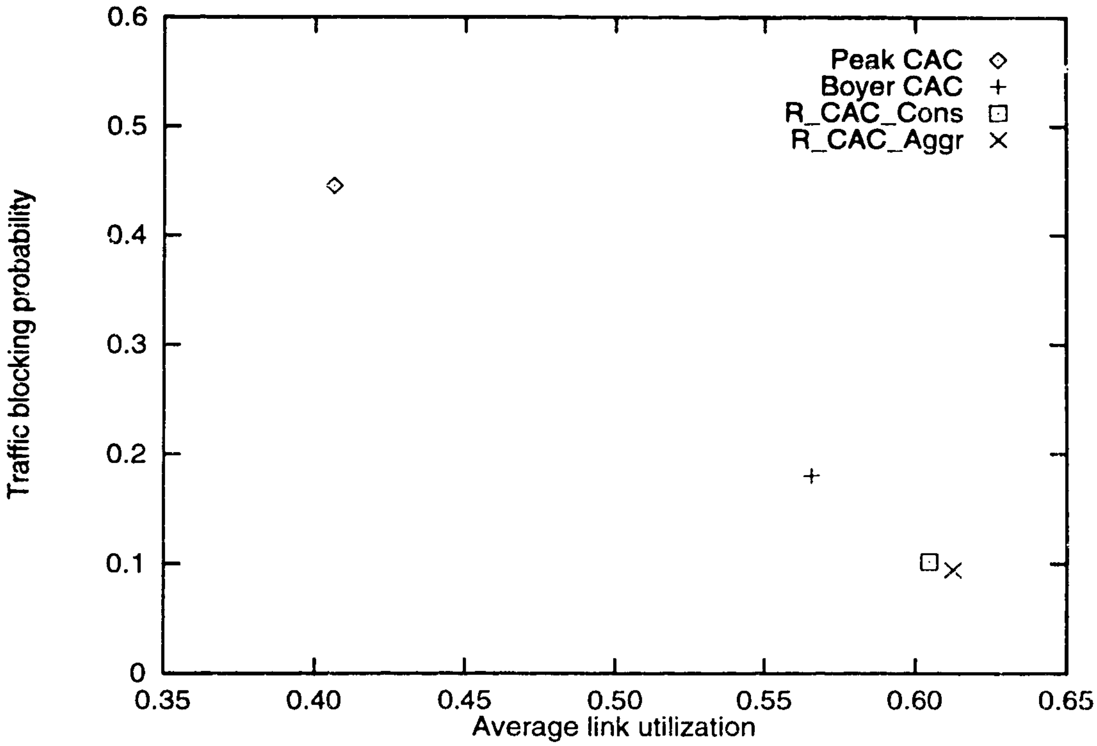


Figure 7.5: Link utilization vs. traffic blocking probability

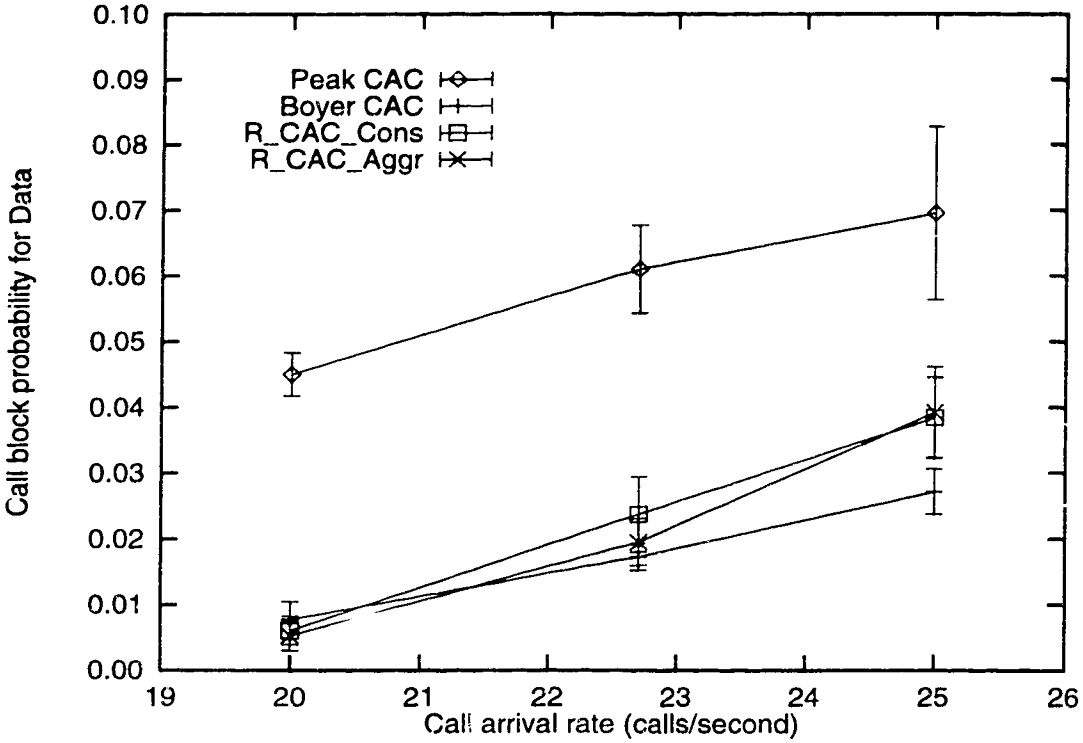


Figure 7.6: Comparison of Call blocking probability for Data

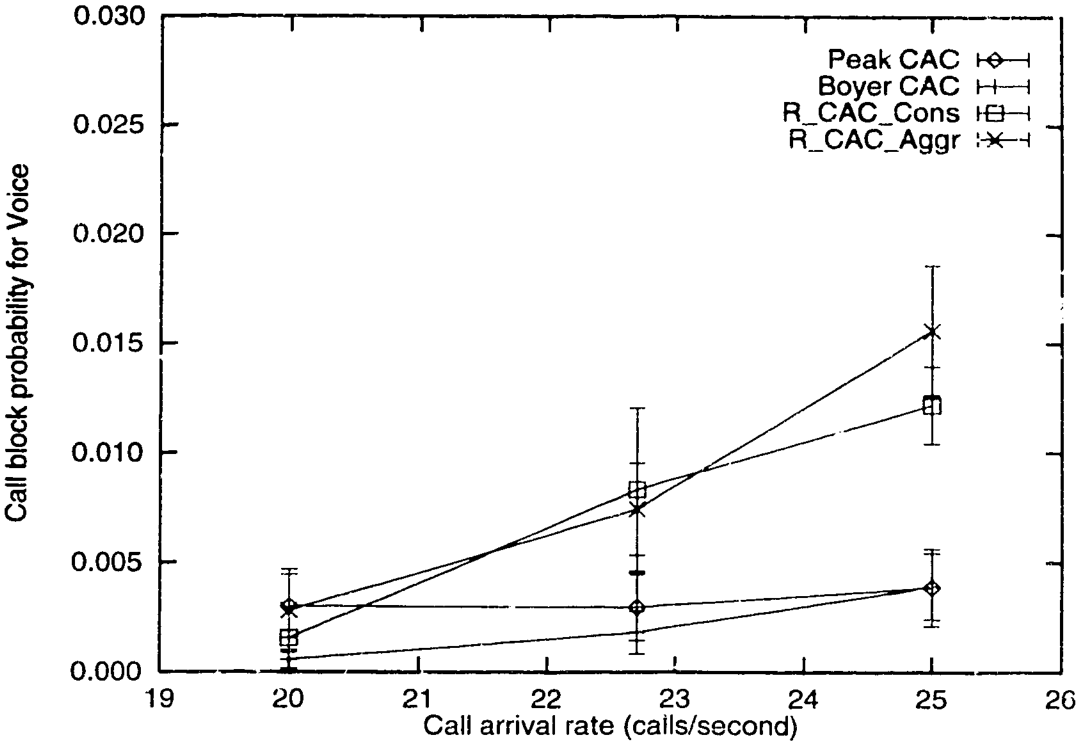


Figure 7.7: Comparison of call blocking probability for Voice

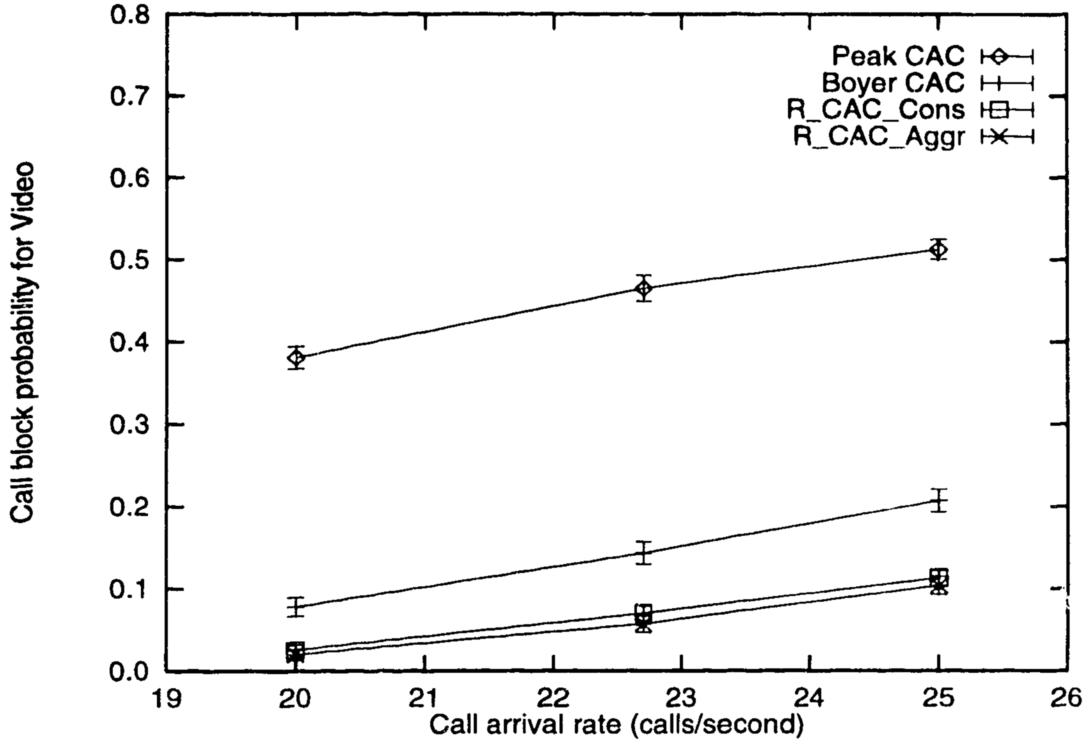


Figure 7.8: Comparison of call blocking probability for Video

CHAPTER 8 Conclusions and Future Work

8.1 Conclusions

The Refinement CAC scheme proposed in this thesis requires minimum information from the users. Users only need to choose a class for their calls. This CAC scheme uses the measured bit rates of the links to assist in making call admission decisions. It also tries to capture the burstiness of the traffic in ATM networks by tracking the bit rate of each call and using refinement to place calls in subclasses. This CAC scheme is compared to two other CAC schemes that rely on one traffic descriptor: peak rate.

The simulation results in Chapter 7 show that the Refinement CAC scheme achieves much improvement in channel utilization and TBP over the Peak CAC and Boyer CAC schemes. Cell delays do become longer with Refinement CAC's, however, this is not a problem if cell delays satisfy the QOS requirements. Buffer size also increases for Refinement CAC's. The Refinement CAC scheme makes better use of the network resources while still providing QOS. Peak CAC severely limits the number of video calls that can be accepted due to the large bandwidth requirement of video.

There is little difference between the performance of the two Refinement CAC's in these simulations because of two reasons. First, calls tend to stay in the medium refined subclass and when there is a call in a high refined subclass, there could be another call in the low refined subclass to compensate that. Second, statistical models for VBR video traffic are too regular and may not fully represent the real traffic [51].

In Section 3.1, a proposal of the criteria for CAC scheme is presented. Now we can discuss if the Refinement CAC meets the criteria as follows.

1. It is demonstrated in Chapter 7 that this scheme can provide QOS with respect to delay measures to all the accepted calls for the traffic sources we used.
2. This scheme uses class as its traffic descriptor. It is very simple and easy to specify by users and therefore is a good traffic descriptor.

3. It is demonstrated in Chapter 7 that this scheme has high utilization. It can achieve better bandwidth saving than Peak CAC and Boyer CAC schemes.

4. The computation of the control is mainly spent on refinement and the calculation of free bandwidth. The refinement does not need much computation since it only calculates the moving average bit rate and compares the bit rate with two thresholds. The calculation of free bandwidth is also simple.

5. This CAC scheme can tolerate users who cannot accurately specify their traffic characteristics because it uses measured information of the network and the accepted calls.

So this scheme can meet most of the criteria for CAC. Several contributions are also made in this tnesis. They are listed as follows.

- A new CAC scheme which tries to optimize the performance and requires minimum information from users is designed. We have demonstrated in Chapter 7 that its performance is better than Peak CAC and Boyer CAC. The new CAC has high link utilization and it can meet the QOS requirements.

- The call blocking probability and traffic blocking probability are used to measure the performance of the CAC schemes. They are important metrics for measuring the performance of CAC schemes. We also measure the mean cell queueing delay, maximum end-to-end cell delay and delay variance because they are important QOS requirements.

- A complete network model with five nodes is used for simulation study. Most of the studies in literature use network models with only one access node. Please see Section 5.3.1 for details. Two network access nodes are designed in the network model. The interaction of the traffic flows from the two access nodes can be studied. Please see Section 5.3.1 for details.

- Six different types of sources are used for the simulation study. They are necessary for modelling the diverse traffic in ATM networks. In this way, the study does not depend on the characteristics of a particular source model.

8.2 Future Work

Future work is needed to explore and improve this CAC scheme. More study on the effect of refinement is needed since there is little difference between conservative and aggressive Refinement CAC's in the experiments of the thesis. Some directions are as follows.

- Study effect of default mean, thresholds, refinement interval in more details.
- Consider new video traffic that is recorded from real video rather than the statistical source models.
- Have the CAC scheme differentiate between sources that tend to always transmit higher (or lower) than default mean rate and sources that move among the refined subclasses more regularly. This implies keeping track of the variability of the calls and adjusting the formula to be more reactive to different calls.
- Treat video and voice calls differently. The CAC scheme could safely accept more voice calls. Therefore, we can use an aggressive scheme when a voice call is considered and use a conservative scheme when a video call is considered.

Other studies could be made based on the following suggestions.

- The probability of a call with either of the two destinations is 0.5 in this study. The cases with other values of destination probability need to be studied because the probability may not be always 0.5 in reality. This could be interesting from the viewpoint of internal congestion.
- The cell delay for each class can be studied. To satisfy the delay requirements of the different classes, we may set priorities for the cells according to the class so that high priority cells can be queued in front of low priority cells.

References

[1] Jonathan S. Turner. "Managing Bandwidth in ATM Networks with Bursty Traffic". *IEEE Network*, Vol. 6, No. 5, pp. 50-58, Sept. 1992.

[2] A. E. Eckberg. "B-ISDN/ATM Traffic and Congestion Control". *IEEE Network*, Vol. 6, No. 5, pp. 28-37, Sept. 1992.

[3] V. J. Friesen and J. W. Wong, "The Effect of Multiplexing, Switching and Other Factors on the Performance of Broadband Networks", *IEEE INFOCOM'93*, pp. 1194-1203.

[4] A. Gersht and K. J. Lee. "A Congestion Control Framework for ATM Networks". *IEEE J. Select. Areas Commun.*, Vol. 9, No. 7, pp. 1119-1130, Sept. 1991.

[5] H. Saito and K. Shiomoto. "Dynamic Call Admission Control in ATM Networks". *IEEE J. Select. Areas Commun.*, Vol. 9, No. 7, pp. 982-989, Sept. 1991.

[6] M. Kawarasaki H. Saito and H. Yamada. "An Analysis of statistical multiplexing in an ATM Transport Network". *IEEE J. Select. Areas Commun.*, Vol. 9, No. 3, 1991.

[7] G. Ramamurthy and R. S. Dighe. "Distributed Source Control: A Network Access Control for Integrated Broadband Packet Networks". *IEEE J. Select. Areas Commun.*, Vol. 9, No. 7, pp. 990-1002, Sept. 1991.

[8] H. Ahmadi, R. Guerin and M. Naghshineh. "Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks". *IEEE J. Select. Areas Commun.*, Vol. 9, No. 7, pp. 968-981, Sept. 1991.

[9] J. J. Bae and T. Suda, "Survey of Traffic Control Scheme and Protocols in ATM Networks", *Proc. of the IEEE*, Vol 79, No 2, Feb 1991, p. 170-189.

[10] S-Q Li, "Study of information loss in packet voice systems," *IEEE Trans. Commun.*, vol. 37, pp. 1192-1202, Nov. 1989.

[11] H. Heffes and D. M. Lucantoni, "A Markov Modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp.856-868, Sept, 1986.

[12] K. Q. Liao and L. G. Mason, "A discrete-time single server queue with a two level modulated input and its applications," in *Proc. IEEE GLOBECOM'89*, pp. 26.1.1-26.1.6.

[13] R. Nagarajan, J. F. Kurose, and D. Towsley, "Approximation Techniques for Computing Packet Loss in Finite-Buffered Voice Multiplexers." *IEEE J. Select. Areas Commun.*, vol 9, No.3, pp.368-377, April 1991.

[14] K. Sriram, R. S. Mckinney, and M. H. Sherif, "Voice Packetization and Compression in Broadband ATM Networks", *IEEE J. Select. Areas Commun.*, Vol 9. No 3, pp. 294-304, April 1991.

[15] P. Sen, B. Maglaris, N. E. Rikli, and D. Anastassiou. "models For Packet Switching Of Variable-Bit-Rate Video Sources," *IEEE J. Select. Areas Commun.*, vol 7, pp.865-869, June 1989.

[16] M. Nomura, T. Fujii, and N. Ohta, "Basic Characteristics Of Variable Rate Video Coding In ATM Environment", *IEEE J. Select. Areas Commun.*, vol. 7, pp. 752-760, June 1989.

[17] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance models of statistical multiplexing in packet video communications", *IEEE Trans. Commun.*, vol 36, pp. 834-844, July 1988.

[18] F. Yegenoglu, B. Jabbari, Y-Q Zhang, "Modeling of motion classified VBR Video Codecs", in *IEEE INFOCOM'92*, pp. 0105-0109.

[19] Z. Q. Zhang, W. W. Wu, K. S. Kim, R. L. Pickholtz, and J. Ramasastry (1991a). "Variable-bit-rate Video Transmission in the Broadband-ISDN Environment," *Proceedings of IEEE*, vol. 79, No. 2, February, pp. 214-222.

[20] P. Pancha, M. E. Zarki, "A Look at the MPEG Video Coding Standard for Variable Bit Rate Video Transmission", *IEEE INFOCOM'92*, pp. 0085-0094.

[21] D. LeGall, "MPEG: A Video Compression Standard for Multimedia Applications.", *Communications of the ACM*, 34(4):305-313, April 1991.

[22] R. Grunenfelder, J. P. Cosmas, S. Manthorpe, and A. Odinma-Okafor, "Characterization of Video Codecs as Autoregressive Moving Average Processes and Related Queueing System Performance", *IEEE J. Select. Areas Commun.*, Vol 9, No 3, pp. 284-293, April 1991.

[23] R. C. Nichol, L. Chiariglione, and P. Schaefer, "The Development of the European Video Conference codec", in *Proc. GLOBECOM'82*, Miami, FL, Nov. 1982, paper D4.3.

[24] P. Skelly, S. Dixit, and M. Schwartz, "A Histogram-based Model for Video Traffic Behavior in an ATM Network Node with an Application to Congestion Control", *IEEE INFOCOM'92*, pp 0095-0104.

[25] P. Skelly, S. Dixit, "Video Traffic Smoothing and ATM Multiplexer Performance." in *Proc. IEEE GLOBECOM'91*, (Phoenix, AZ), 1991.

[26] B. Melamed, D. Raychaudhuri, B. Sengupta, and J. Zdepski, "TES-Based Traffic Modeling for Performance Evaluation of Integrated Networks", *IEEE INFOCOM'92*, pp 0075-0084.

[27] Y. Yasuda, H. Yasuda, N. Ohta, and F. Kishino, "Packet Video Transmission Through ATM Networks," in *Proc. IEEE GLOBECOM'89*, pp. 25.1.1-25.1.5.

[28] R. Bolla, F. Danovaro, F. Davoli, M. Marchese, "An Integrated Dynamic Resource Allocation Scheme for ATM Networks", *Proc. INFOCOM'93*, pp. 1288-1297.

[29] T. Kamitake, T. Suda, "Evaluation of an Admission Control Scheme for an ATM Network Considering Fluctuations in Cell Loss Rate", *Proc. GLOBECOM '89*, Dallas, TX, Nov. 1989, pp. 1774-1780.

[30] T. E. Tedijanto and L. Gün, "Effectiveness of Dynamic Bandwidth Management Mechanisms in ATM Networks", *Proc. INFOCOM'93*, pp. 358-367.

[31] J. M. Hyman, A. A. Lazar, G. Pacifici, "Real-Time Scheduling with Quality of Service Constraints", *IEEE J. Select. Areas Commun.*, Vol. 9, No. 7, Sept, 1991.

[32] A. A. Lazar, G. Pacifici, J. S. White, "Real-Time Traffic Measurements on MAGNET II", *IEEE J. Select. Areas commun.*, Vol. 8, No. 3, April, 1990.

[33] R. Izmailov and E. Ayanoglu, "Priority Statistical Multiplexing Of Mixed VBR Video And CBR Traffic In B-ISDN/ATM With A Threshold Algorithm", *Proc. INFOCOM'93*, pp. 910-918.

[34] D. M. Cohen and D. P. Heyman, "A Simulation Study of Video Teleconferencing Traffic In ATM Networks", *Proc. INFOCOM'93*, pp. 894-901.

[35] E. P. Rathgeb, "Modelling and Performance Comparison of Policing Mechanisms for ATM Networks", *IEEE J. Select. Areas Commun.*, Vol. 9, No. 3, April, 1991.

[36] B. A. Makrucki, "A Study of Source Traffic Management And Buffer Allocation in ATM Networks", *The 7th ITC Specialist Seminar*, Morristown, NJ, Oct., 1990.

[37] T. Kamitake, T. Suda, "Evaluation Of An Admission Control Scheme For An ATM Network Considering Fluctuations In Cell Loss Rate", *Proc. IEEE Globecom'89*, pp. 49.4.1 - 49.4.7

[38] J. N. Daigle and J. D. Langford, "Models For Analysis of Packet Voice Communications Systems", *IEEE J. Select. Areas Commun.*, Vol. SAC-4, No. 6, Sept. 1986.

[39] P. Pancha and M. E. Zarki, "Bandwidth Requirements Of Variable Bit Rate MPEG Sources In ATM Networks", *Proc. INFOCOM'93*, pp. 902-909.

[40] D. Reininger, D. Raychaudhuri, "Statistical Multiplexing of VBR MPEG Compressed Video On ATM Networks", *Proc. INFOCOM'93*, pp. 919-926.

[41] C. A. Cooper and K. I. Park, "A Reasonable Solution to the Broadband Congestion Control Problem", *International Journal Of Digital And Analog Communication Systems*, Vol. 3, pp. 103-115.

[42] Jean-Yves Le Boudec, "The Asynchronous Transfer Mode: a tutorial", *Computer Networks and ISDN Systems*, North-Holland, 24, 1992, pp. 279-309.

[43] P. Boyer, "A Congestion Control for the ATM", *The 7th ITC Specialist Seminar*, Morristown, NJ, Oct., 1990.

[44] A. M. Law and W. D. Kelton, *Simulation Modeling & Analysis*, 2nd Edition, McGraw-Hill, 1991.

[45] W. Stallings, *ISDN and Broadband ISDN*, 2nd Edition, Macmillan Publishing Company, NJ, 1992.

[46] A. I. Elwalid, D. Mitra, "Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks (Extended Abstract)", *Proc. INFOCOM'93*, pp. 256-265.

[47] L. K. Reiss, L. F. Merakos, "Shaping of Virtual Path Traffic for ATM B-ISDN", *Proc. INFOCOM'93*, pp. 168-175.

[48] A. Campbell, G. Coulson, F. Garcia, D. Hutchison and H. Leopold, "Integrated Quality of Service for Multimedia Communications", *Proc. INFOCOM'93*, pp. 732-739.

[49] R. O. Onvural, *Asynchronous Transfer Mode Networks*, Artech House, Boston, London, 1994.

[50] J. W. Wong, *Notes on Discrete Simulation*, Department of Computer Science, university of Waterloo.

[51] E. P. Rathgeb, "Policing of Realistic VBR Video Traffic in an ATM Network", *International Journal of Digital and Analog Communication Systems*, Vol. 6, pp. 213-226, 1993.

[52] L. Kleinrock, *Queueing Systems Volume 2: Computer Applications*, John Wiley &Sons, 1976.

Appendix A Peak Rate of Poisson Process

We use simulation experiments to determine the peak rate of Poisson process because it cannot be obtained by theoretical analysis. The Poisson process with rate $\lambda>0$, has the property that the interarrival times are IID exponential random variables with mean $1/\lambda$. The Poisson process is implemented by exponential variates X with the following formula,

$$X = -(1/\lambda)\ln U \qquad \text{(EQ A.1)}$$

where U is uniform random variable $U(0, 1)$.

We run simulations of Poisson process to determine its peak rate. The whole simulation time T is separated into n constant intervals t_i ($1 \le i \le n$) as shown in Figure A.1. The mean rate R_i of each interval is measured. The peak rate R_{peak} is obtained with (EQ A.2).

$$R_{peak} = max(R_i) \qquad \text{(EQ A.2)}$$

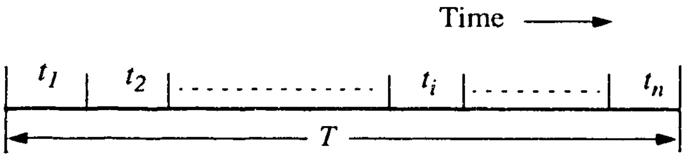


Figure A.1: The time intervals for simulations of Poisson Process

We use cell rate for λ. The bit rate is $424 \times \lambda$ since each cell contains 424 bits by CCITT standard. We set $\lambda = 1$ cell/s and made several simulations by varying the constant interval t_i. The results are shown in Figure A.2. The peak cell rate decreases when the measurement interval increases. This is reasonable because the cell rate tends to the mean rate (1 cell/s) when the interval increases. When the interval becomes larger than 100, the decreasing trend of peak cell rate slows down.

We will compare the performance of Peak CAC and Refinement CAC. The choice of peak rate of Poisson process can affect the performance of Peak CAC. For Peak CAC, the lower the peak rate, the better the performance. We choose a point near the interval of 300 seconds. So the peak rate is expressed by (EQ A.3).

$$R_{peak} = 1.2 \times \lambda \qquad \text{(EQ A.3)}$$

With a measurement interval of 300 seconds, the average of 300 cells are generated in each interval.

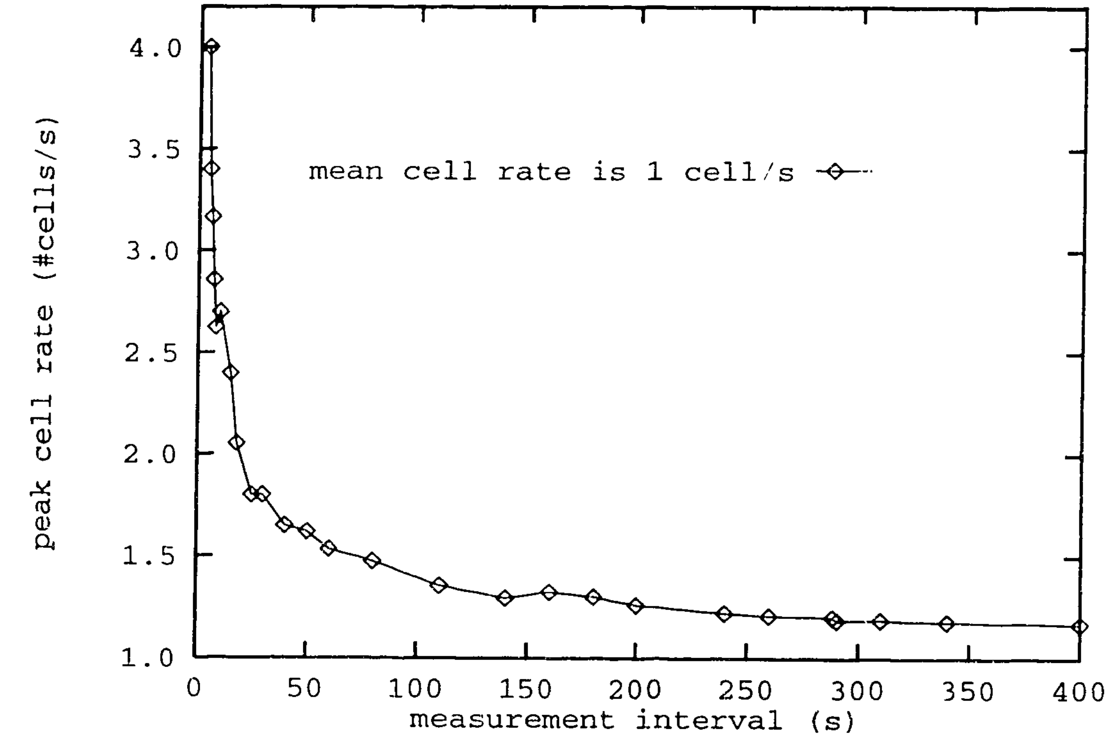


Figure A.2: The peak rates of Poisson Process from simulations

In the experiments of this thesis, we have mean bit rates of 64 kbps and 1000 kbps. Their peak rate are chosen by (EQ A.3). Their peak rates can also be obtained by setting the measurement interval during which an average of 300 cells are generated.