1997

# Traffic Management and Congestion Control in the ATM Network Model.

Sundararajan Vedantham

*Louisiana State University and Agricultural & Mechanical College*

# TRAFFIC MANAGEMENT AND CONGESTION CONTROL IN THE ATM NETWORK MODEL

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Computer Science

by
Sundararajan Vedantham
B.E. in Electronics and Instrumentation, Annamalai University, India, 1986
M.S. in Electrical Engineering, Louisiana State University, 1991
December 1997

UMI Number: 9820756



UMI
300 North Zeeb Road
Ann Arbor, MI 48103

# Acknowledgments

I would like to thank my advisor Dr. S.S.Iyengar for his guidance in this research effort, and for the freedom and continued support he provided all these years. I am grateful to Dr. J.L.Trahan who meticulously went through the dissertation and helped me verify and improve the proofs, and writing as a whole. More than just thanks are due to my friend and philosopher Dr. Amit Nanavati for everything he did to enrich me. I should also gratefully acknowledge my wife Maya for putting up with me during the writing phase, and all my friends who supported me throughout my eight year stint at LSU.

# Table of Contents

# List of Tables

# List of Figures

# Glossary of Terms

**AAL** ATM Adaptation Layer

**ABR** Available Bit Rate. A type of ATM traffic that uses the available bandwidth on the network to transmit the information.

**ADPCM** Adaptive differential pulse code modulation.

**ASCII** American Standards Committee for Information Interchange.

**ATM** Asynchronous Transfer Mode, in this context it refers to the ITU- T's standard 53 byte cell based transfer mode defined for B-ISDN.

**BECN** Backward Explicit Congestion Notification.

**B-ISDN** Broadband Integrated Services Digital Network.

**BAP** Bandwidth Allocation Problem.

**CAC** Connection Admission Control.

**CBR** Constant Bit Rate.

**CCITT** International Telegraph and Telephony Consultative Committee.

**CDV** Cell Delay Variation; variation in the inter-cell arrival time over a given period.

**CELP** Code-excited linear prediction; audio encoding method for low-bit rate codes.

**Cell** A fixed length 53-octet packet used in ATM.

**CH** Cell Header.

**CIF** Common Interchange Format; interchange format for video images with 288 lines by 352 pixels per line of luminance, and 144 lines by 176 pixel per line of chrominance information.

**Circuit Switching** Where a path through a switch is established for the duration of a connection.

**CLP** Cell Loss Priority.

**CLR** Cell Loss Ratio.

**Codec** Short for coder/decoder; device or software that encodes and decodes audio or video information.

**CP** Cell Payload.

**CPE** Customer Premises Equipment

**CSMA/CD** Carrier Sense Multiple Access/Collision Detection.

**DARPA** Defense Advanced Research Projects Agency (USA).

**DCE** Data Communications Terminating Equipment.

**Demultiplexing** Extraction of multiple data paths that have previously been multiplexed over a single underlying medium or channel.

**DTE** Data Terminal Equipment.

**Encoding** Transformation of the media content for transmission, usually to save bandwidth but also to decrease the effect of transmission error.

**Ethernet** CSMA/CD based Local Area Network technology.

**FDM** Frequency Division Multiplexing. Virtual channels are placed on different carrier frequncies within the bandwidth of the physical medium.

**FPS** Fast Packet Switching.

**GCRA** Generic Cell Rate Algorithm.

**GFC** Generic Flow Control.

**GIF** Graphical Interchange Format.

**GUI** Graphical User Interface.

**H.261** ITU-T recommendation for the compression of motion video at rates of p*64 kbps (where p=1..30). Originally intended for narrowband ISDN.

**HDTV** High Definition Television.

**HEC** Header Error Check.

**IEEE** Institute of Electrical and Electronics Engineers.

**IETF** Internet Engineering Task Force.

**IP** DARPA Internet Protocol; the Internet Protocol, defined in RFC 791, is the network layer for the TCP/IP protocol suite.

**ISDN** Integrated Services Digital Network; refers to an end-to-end circuit switched digital network intended to replace the current telephone network.

**ISO** International Standards Organization.

**ITU** International Telecommunication Union.

**ITU-T** ITU Telecommunication Standardization Sector (was CCITT).

**JPEG** ISO/CCITT Joint Photograph Experts Group. Designation of a variable rate compression algorithm using discrete cosine transforms for still frame color images.

**kbps** Kilo Bits per second.

**LAC** Last Conformance Time.

**LAN** Local Area Network.

**LBA** Leaky Bucket Algorithm.

**LLC** Logical Link Control layer.

**LPC** Linear predictive coding. Audio encoding method that models speech as a parameter of linear filter; used for very low bit rate codecs.

**MAC** Medium Access Control layer.

**Mbps** Mega Bits per second.

**MPEG** ISO/CCITT Motion Picture Experts Group. Designates a variable rate compression algorithm for full motion video at low bit rates.

**Multicast** Where a single PDU is sent over a single interface and is delivered to multiple destinations.

**Multimedia** Integration of multiple presentation media into a single user interface.

**Multiplexing** Interleaving of multiple data paths over a underlying medium or channel.

**NISDN** Narrow Band Integrated Services Digital Network.

**OSI** Open System Interconnection; a suite of protocols designed by ISO committees.

**Packet Switching** Network model transmitting information packets from source to destination in a store and forward manner where a path through a switch is established only for the duration of a packet.

**PDU** Protocol Data Unit.

**PCM** Pulse-code modulation; speech coding where speech is represented by a given number of fixed-width samples per second. Often used for the coding employed in the telephone network at 64k eight-bit samples per second.

**PTI** Payload Type Identifier.

**PVC** Permanent Virtual Circuit.

**QoS** Quality of Service - characteristics such as throughput, cell loss error rates and CDV that may be associated with a virtual connection.

**QCIF** Quarter CIF; format for exchanging video images with half as many lines and half as many pixels per line as CIF.

**RFC** Request For Comments. Documents issued by the IETF describing standards for use within the internet.

**RPE/LTP** Residual Pulse Excitation/Long Term Prediction.

**SAP** Service Access Points.

**SDH** Synchronous Digital Hierarchy

**SMDS** Switched Multimegabit Data Service.

**SONET** Synchronous Optical NETwork. Similar to SDH.

**Study Group 13** Group within ITU-T responsible for the development of B-ISDN. Was Study Group XVIII under CCITT.

**Switch** In the ATM context, a point where cells are copied from one physical medium to another, possibly having their VPI/VCI fields changed in the process.

**TAT** Theoretical Arrival Time (of an ATM cell). A parameter used in congestion control.

**TCP** Transmission Control Protocol; an internet standard transport layer protocol defined in RFC 793.

**TDM** Time Division Multiplexing.

**UBR** Unspecified Bit Rate.

**UDP** User Datagram Protocol; unreliable, non-sequenced connectionless transport protocol defined in RFC 768.

**UNI** User Network Interface.

**Unicast** Where a single PDU is sent over a single interface and is delivered to a single destination.

**UTP** Unshielded Twisted Pair.

**UPC** Usage Parameter Control.

**VBR** Variable Bit Rate.

**VC** Virtual Circuit.

**VCI/VPI** Virtual Channel Identifier and Virtual Path Identifier. A virtual connection is identified on a given section of fiber by the VCI and VPI it has been allocated.

**VCR** Video Cassette Recorder.

**VOD** Video On Demand.

**Virtual Connection** A 'bit pipe' between two endpoints between which data may be exchanged. The connection may have only transitory or notional relationship to physical paths between the two endpoints.

**VS** Virtual Scheduling.

**WAN** Wide Area Network.

**WDM** Wavelength Division Multiplexing. Optical FDM.

**X.25** CCITT recommendation for the interface between packet switch DTE and DCE equipment.

# Abstract

Asynchronous Transfer Mode (ATM) networking technology has been chosen by the International Telegraph and Telephony Consultative Committee (CCITT) for use on future local as well as wide area networks to handle traffic types of a wide range. It is a cell based network architecture that resembles circuit switched networks, providing Quality of Service (QoS) guarantees not normally found on data networks. Although the specifications for the architecture have been continuously evolving, traffic congestion management techniques for ATM networks have not been very well defined yet. This thesis studies the traffic management problem in detail, provides some theoretical understanding and presents a collection of techniques to handle the problem under various operating conditions. A detailed simulation of various ATM traffic types is carried out and the collected data is analyzed to gain an insight into congestion formation patterns. Problems that may arise during migration planning from legacy LANs to ATM technology are also considered. We present an algorithm to identify certain portions of the network that should be upgraded to ATM first. The concept of *adaptive burn-in* is introduced to help ease the computational costs involved in virtual circuit setup and tear down operations.

# Chapter 1

# Introduction

The dramatic technological advancements in computers and communication in the last few decades has made the end of twentieth century *The Information Age.* There are various sociological, technical, business, economic and political factors that fuel the research and development process in these areas by providing ample justification and the required resources. Consider the following trends:

- Computers are becoming ubiquitous and are becoming easier to use, driving individual usage further.

- Rather than being used as stand alone equipment, computers are being networked more and more.

- The richness (variety and multimedia content) and the volume of information available on the internet is raising rapidly.

- National as well as global level interactions between various entities are becoming easier by the day.

- Use of fiber optics to carry digital data (thus bringing in enormous bandwidth) is becoming popular, and the trend in turn is generating more traffic.

- Our entire society is becoming increasingly dependent on the ready availability of numerous types of information for its normal functioning.

It is becoming increasingly evident that the world is well poised today for such trends to continue well into the twenty-first century. In addition to these trends, another development that has marked this decade is the merger between the domains of computers and communication we are witnessing now. The increased use of digital technology in long distance telephone services initiated the merger process. The arrival of the Internet in a big way and its slowly increasing ability to handle multimedia traffic has made the ambience quite conducive for such a merger.

From the end user's perspective, this propensity is easily perceived by the use of computers for voice and video conferencing, email, information dissemination through the world wide web, live radio and television broadcasts over the internet and the like. Thus, computers are no longer being used for number crunching applications exclusively. In fact, they are used more and more as communication tools rather than computing tools. Beyond the current experiments, the possibility of delivering voice, video and data services on demand through one fiber optic network maintains the excitement and is pushing the technology further. The origin of the concept of *Broadband ISDN* and more specifically, the development of *Asynchronous Transfer Mode* networking is the logical next step in this evolution. In the next few sections of this chapter we analyze in detail the factors that justify the development and deployment of the ATM model.

## 1.1 Raising Bandwidth Requirements

The development of B-ISDN was initiated in the 1980's in order to be able to support the bandwidth requirements of high quality images and real-time video delivery over a data network. It is a known fact that digital delivery of high quality images

and real-time TV requires bandwidth in the 10s and 100s of Mbps. The following subsections discuss how compression techniques have resulted in a wide variety of mechanisms for delivering images, with varying trade-offs between bandwidth consumption at the network level and processing load at the end node level.

### 1.1.1 Still image transfer

Raw bit mapped still images frequently consume hundreds and thousands of kbytes of storage space. For example, a 24 bit 'true color' picture of 480 by 640 pixels consumes 900 Kbytes. Hence, high resolution image retrieval services can impose a significant peak load on the network. Studies at Bellcore [26] have concluded that users will expect retrieval systems to respond with images shortly after a selection is made (0.2 seconds for choices that are perceived to involve little processing, and 2 seconds for 'complex' selections). Rapid retrieval and display of such images places a greater peak load on the network than more traditional ASCII text retrieval. Bringing a reasonable quality 'true color' picture (960 by 1280 pixel) to the screen within 2 seconds would require network traffic in excess of 14.4 Mbps (ignoring processing time). The required bandwidth would increase by a factor of 10 if a 0.2 second response time is imposed [3]. The ubiquitous 10 Mbps Ethernet LANs that are widely used today are not capable of handling such requirements with ease. So various schemes are being developed for encoding and storing still images. The Graphical Interchange Format (GIF) offers exact reproduction of an image using non-lossy encoding techniques. For example, a 1152 by 800 pixel, 8 bit per pixel color image consumes only 470 KB in GIF format (the exact compression achieved depends on the source material).

The ISO/CCITT Joint Photographic Experts Group has developed a lossy image storage algorithm called JPEG, intended for the storage of photo quality 'real world' images. The JPEG encoding algorithm allows a user to specify the acceptable level of loss on a per image basis, with decoding being independent of the encoded loss level. The 470 KB GIF file consumes only 90 KB when JPEG encoded at a quality value of 75%. At a quality value of 50% the image consumes only 57 KB. Subjective assessment of the images at 1 meter distance from the screen suggests that not much degradation could be perceived at the 75% level. At the 50% level the picture is still quite acceptable, although the image brightness starts to look exaggerated. In [36] the JPEG standard is described as producing almost perfect images with a compression ratio of 5:1, and moderate picture quality with compression as high as 30:1. JPEG encoding and decoding hardware is becoming available for current work stations. The major limitation of JPEG is that it does not cope well with 'sharp edges' in images. Designers can now trade network bandwidth for local processing load. Rapid retrieval of GIF or JPEG encoded images will require lower peak bit rates, but at the expense of the processing time it takes for decompression and display.

### 1.1.2 Video transmission

The possibilities for use of audio and video together is ever growing. If we consider, video conferencing applications, some conceive of video conferences where approximately life size images are distributed across the digital network. The VideoWindow system developed by Bellcore [32] is one such scheme, using 3ft by 8ft display screens at each end of a link. Research with VideoWindow revealed that public video con-

ferencing was relatively insensitive to variations in transmission bandwidth ranging from 384 kbps to 45 Mbps.

Alternatively, small video windows sharing screen space with other windows on the traditional work station or PC screen is also a popular video conferencing model. A decrease in acceptable image size decreases the required image resolution. This has implications for the types of video compression and encoding algorithms that may be profitably used. CCITT's H.261 standard provides digital video at rates between 64 kbps and 1920 kbps (in increments of 64 kbps). At 64 and 128 kbps, H.261 is considered acceptable for videophone style applications (at 176 by 144 pixels - known as QCIF, or Quarter Common Interchange Format). 384 kbps is considered as a minimum to reasonably support a video conference (at 352 by 288 pixels - CIF) [24, 25]. The image quality increases with bit rate, but "has been perceived as less than VCR quality at 1.544 Mbps" [35]. While it was originally conceived for use on fixed rate circuits, H.261 video is already appearing on the Internet, using software codecs and carrying the data within User Datagram Protocol (UDP) packets.

MPEG-1, which is an encoding scheme to allow the delivery of real time VCR quality video and audio within the constraints of 1.544 Mbps data links, has been developed by the ISO Moving Pictures Experts Group (MPEG) [23]. Presently it is processing intensive at the encoding end, but decoding is easily achieved in real time. MPEG-1 appears well suited for consumer Video On Demand services. It will not be applied to video conferencing, as standards such as H.261 involves substantially less encoding processing than MPEG. A subsequent development is MPEG-2, which aims to deliver broadcast quality video and audio within the constraints of links up to 10Mbps, and HDTV at even higher rates [15]. Developments such as JPEG, MPEG, and H.261 are reducing the network capacity needed to support fairly basic

image and videophone applications. Careful structuring of interactive multimedia documents can also reduce the peak bit rates needed by spreading out the intervals between user requests for new information.

### 1.1.3 Audio transmission

Audio encoding and compression schemes also require compromises between the quality of signal reproduction and reducing the bit rate. A common telephony standard is CCITT G.711, which defines an encoding method commonly known as $\mu$-law encoding. It uses the standard PCM data rate of 64 kbps (the format used by the audio chip in work stations such as the Sun SPARC family). Two further encoding schemes are CCITT G.721 and G.723, utilizing Adaptive Differential Pulse Code Modulation (ADPCM). G.721 specifies ADPCM at 32 kbps, while G.723 specifies ADPCM at two rates - 24 or 40 kbits/sec. CCITT G.722 uses sub-band ADPCM encoding to provide 7kHz audio bandwidth at 64 kbps, allowing for broadcast quality wide-band speech to be distributed across long distance digital networks [3]. Linear Predictive Coding (LPC, or 'vocoding') and Code-Excited Linear Predictive coding (CELP) schemes have also emerged to provide voice transmission with exceedingly low bit rates. The US Department of Defense's Federal Standards 1015 and 1016 specify LPC at 2.4 kbps and CELP at 4.8 kbps respectively. Qualcomm have produced a variable rate QCELP coder for their cellular phone 'Common Air Interface', where the bit rate varies between 1 and 8 kbps. LPC and CELP schemes operate by sending parameters to excite a vocal synthesising system at the decoding end, limiting their effective use to speech transmission. A newly found internet company called RealAudio has managed to deliver FM radio quality audio (both speech as well as music) on 14 kbps speed in 1996. As with image and video techniques, lower

data rates are achieved at the expense of higher processing loads for encoding and decoding.

Such developments indicate that the digital domain can be effectively used as a common medium to carry audio, video and data traffic simultaneously. While we do use digital technology to transport all these types of information, it is not done through an efficient single transportation mode. ATM seems to be a natural answer to such a situation. While the basic philosophy of ATM lends itself well to multimedia material transportation, the inherent nature of most of multimedia traffic requires the transmission to be of very high quality and speed. Thus, resolving traffic congestion management issues takes an added significance in the ATM model.

## 1.2 The Role of ATM

ATM has evolved as the standard for future networking that is expected to carry voice, real time video and a large volume of still images in addition to the growing volumes of computer data. It was formally adopted as the fundamental networking technology of the Broadband Integrated Services Digital Network (B-ISDN) in the late 1980s by the CCITT (International Consultative Committee for Telecommunications and Telegraphy) (now renamed ITU-T, International Telecommunication Union - Telecommunication Standardization Sector). ATM works on the assumption that the required bandwidth for transmission will be available throughout the connection time; the Quality of Service (QoS) deteriorates drastically when the bandwidth requirements of the source are not met by the network. ITU-T, which is in the process of developing the specifications for world wide ATM networks, has issued a Recommendation titled I.371 dealing with Traffic Control and Conges-

tion Control in B-ISDN. The I.371 recommendation defines terminology for traffic parameters, a traffic contract, conformance checking, resource management, connection admission control, prioritization and implementation tolerances. But it does not mandate as to how traffic congestion should be handled. As of now, the standards development process is on a year long hiatus to enable the industry to catch-up with the standards set so far. So at this point in time, traffic congestion management is left to the discretion of the implementor. There are a few mechanisms described in the literature and implemented on the networks. But each one has one or more disadvantages that makes it unsuitable for general ATM networks.

## 1.3 Scope of the Dissertation

This research approaches the traffic management problem in the ATM network model both from a theoretical as well as a practical perspective. So the problem of allocating the required bandwidth for the incoming calls and the problem of planning a migration from a legacy LAN/WAN to the ATM environment is considered from a theoretical perspective. Better understanding of the existing problems and plausible solutions are provided. Although the problems in these two areas are approached from a theoretical perspective, the solutions developed are very practical ones. In a separate section, we emphasize the importance of carrying out empirical simulations so that practical problems that may slip through theoretical models are brought to light. A number of simulations are also carried out and the interpretation of the collected data is also presented.

We also analyze the problem of improving the negotiation process between the traffic source and the network for the establishment of traffic contract. We present

a new concept that is bound to reduce the computation costs involved in the negotiation process. Thus, throughout the work, we have tried to maintain a balance between theoretical as well as practical analysis of the problem at hand.

## 1.4 Outline

Chapter 2 provides a brief introduction to the ATM protocol and network. Section 2.5 discusses the motivation behind studies on traffic congestion problems from the ATM perspective. Chapter 3 presents a brief survey of the literature and cites existing methods to control the problem. Chapter 4 proves that the bandwidth allocation problem in the ATM networking model is NP-Complete. It also discusses the possibility of using genetic algorithms for effective bandwidth utilization. Chapter 5 considers the simulation of ATM networks on legacy LANs in order to study the performance and suitability characteristics of ATM for migration planning. Chapter 6 considers migration planning from a theoretical perspective and provides an algorithm for selecting network links for conversion from legacy LAN links to ATM links while ensuring maximum possible efficiency. Chapter 7 analyzes the bandwidth negotiation process between the network and the end user equipment and suggests ways to make the negotiation more intelligent where input from the end user equipment also helps network routers in setting up virtual paths for transmission. Chapter 8 summarizes our contributions and presents the conclusions with direction for future research efforts.

# Chapter 2

# ATM Networking

Consider voice telephone, cable television and the internet. These three networks span the entire US and a large part of the world. All three of these networks are meant for telecommunication purposes, and are predominantly laid on the ground using metal or optical cable. In spite of such striking similarities in characteristics, and their ubiquitous nature, they are largely incompatible amonst themselves. We do have a few more electronic networks that span the entire planet based on satellite technology, microwave networks, etc. Still no one network is suitable for a wide range of transportation services. Since the transportation mechanisms involved are incompatible, they exist as independent networks where free bandwidth available on one system cannot be used by the other. In addition to this inefficiency, these networks are not even capable of taking full advantage of break-throughs in technology. Improvements in audio and video coding, VLSI technology and end user terminals rapidly change the service requirements of the networks. For example, developments in video compression techniques may decrease the bandwidth requirement for cable TV transmission into one half of its existing value. But the present analog cable TV system cannot take advantage of this development. This status suggests developing a single universal network that is

1. Capable of handling various types of traffic,

2. Adept at taking advantage of new technological innovations,

3. Effective in using the available bandwidth efficiently, and

4. Less expensive.

Narrow band Integrated Services Digital Network (N-ISDN) is a step in this direction but is restricted to data and voice. Its bandwidth limitations preclude video transmission.

## 2.1 Integrated Broadband Solution

ATM Network model is a compromise between the transmission requirements of audio, video and computer data that provide a Broadband ISDN solution. The basic idea behind ATM is to break down any data to be transported, into fixed size cells of 53 bytes each (48 bytes of data and a 5 byte header) and handle transmit the information as a flow of cells regardless of the type of source generating the data. Therefore, the underlying switching fabric or the transmission medium need not be aware of the service being transported. This approach ensures that the new network will be well placed to take advantage of any high speed transmission medium (fiber or whatever comes next) as well as improvements in digital data compression algorithms; it will also be capable of utilizing the available bandwidth effectively. In addition, ATM is expected to be implemented entirely on fiber optic networks that are capable of handling 155 to 650 Mbits per second (compared to 2 to 10 Mbits per second speed of today's LANs and internet) data streams making it highly suitable for real time video. The protocol itself is capable of handling traffic rates of the order of 2.4 Gbits per second [28]. The cell format is now stable and the technology to build ATM switches is currently available. But fundamental issues concerning traffic control and connection usage enforcement are yet to be solved.

ATM networks will still have a set of sources and destinations trying to communicate, as is the case in existing networks. Since several different data types may travel on an ATM network, rate of information generation at the source gains added importance. Let us say we represent the natural information generation rate of a source (say digital video) as a stochastic process $s(t)$ which lasts for time duration $T$. Then the *peak natural bit rate* $S$ and the *average natural bit rate* $E[S(t)]$ are two important parameters that describe the traffic generation. $S = max\ s(t)$ and $E[S(t)] = \frac{1}{T}\int s(t)\,dt$. The ratio between the maximum and the average natural information rate is called the *burstiness* $B = \frac{S}{E[S(t)]}$. The system that is generating the traffic (for example, a computer to which a video camera and a telephone are attached) is expected to understand the natural bit rate of each device attached and should present a Source Traffic Description for the whole system while negotiating with the network for bandwidth allocation. Although it is basically a packet transmission network, a source attempting a transmission first has to negotiate with the network for the required bandwidth. A standard set of traffic parameters are used in the negotiation to describe the traffic type. These parameters are

1. Average connection holding time

2. Peak cell rate

3. Mean cell rate

4. Average burst duration

5. Source type (telephone, video camera, etc.)

Once the negotiation is over, the network statistically guarantees the negotiated bandwidth between the source and the destination by setting up a virtual path

between the two. Although the transmission medium may be multiplexed as in the case of packet switching networks, the existence of a virtual path makes the system resemble circuit switching, making it a viable option for on line video transmission and the like. Development of switches that can handle traffic of this magnitude is an area of enormous interest to the industry [14, 28].

Since the transmission rate is very high compared to X.25 type networks, addressing schemes are kept very simple with highly reduced header functionality so as not to take too much time during transmission. Since the transmission medium is supposed to be fiber as opposed to copper in case of present day LANs and WANs, the error rate in transmission is expected to be very low. ATM takes advantage of this scenario by reducing the error detection and correction carried out during the transmission thus increasing the rate of transmission. Since the network functions under the assumption that transmission will be error free and congestion free, if those assumptions fail even briefly, it results in significant deterioration of transmission quality.

## 2.2 Cell Format

ATM networks use the *cell* as the basic unit of information that is transported across the network. Although so many issues in ATM are still being debated, the format of a cell is one item that is properly defined in the standards today. A cell is a fixed size information unit (as opposed to *packets* in other network models that can vary in size) that is 53 bytes long that comprises of a 5 byte long header and a 48 byte long payload. As shown in Figure 2.1, the five byte header has specific bits allocated for carrying very specific information. The 53 byte long cell size

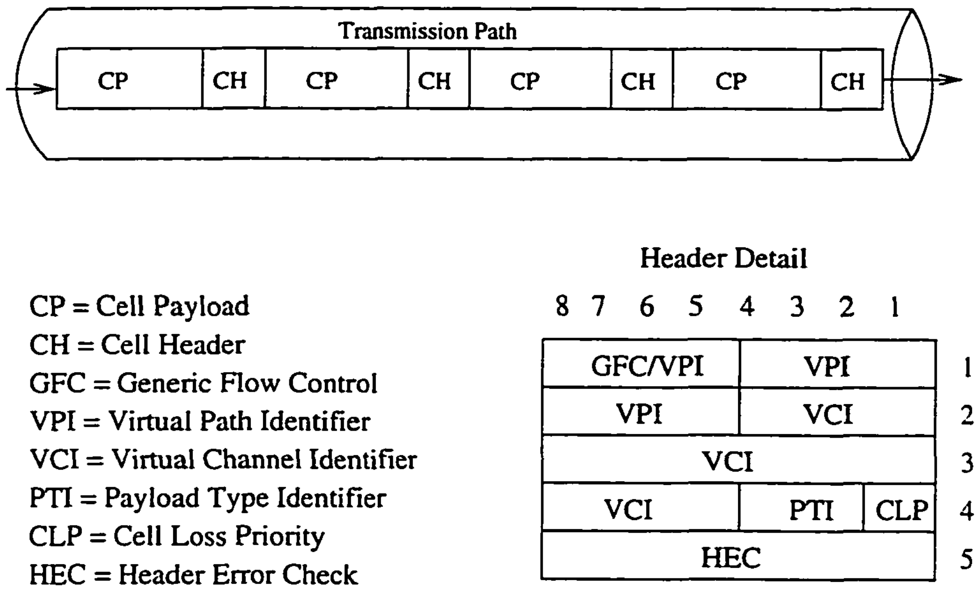


Figure 2.1: Cell Transmission and Header Detail

itself is a compromise reached after prolonged deliberation between North American, European and Japanese sides that debated between a 32 byte and a 64 byte format! All information is switched and multiplexed in the network in these fixed-length cells only.

The cell header identifies the destination, cell type and priority. The VCI and VPI information, which have only local significance and continuously change as the cell travels from one ATM switch to another, identify the immediate destination for the cell. The PTI field indicates the type of data (maintenance, signaling or user) contained in the cell. The CLP bit (that can take a 0 (high priority) or 1 (low priority) value) defines the relative importance of the cell with respect to other cells that are found in the same traffic. Congestion control mechanisms may choose to discard low priority cells wherever demand exceeds the available bandwidth. The HEC field helps in error detection and correction in the header. Payload error correction is expected to be carried out by higher level layers. This fixed size cell

usage simplifies the implementation of ATM networks. Since even naturally long information streams are chopped and accommodated into these tiny cells, it prevents longer packets delaying the transmission of smaller packets that may be carrying time sensitive (live audio and video for example) information thus making ATM a better choice for time sensitive as well time insensitive traffic.

## 2.3 Virtual Connection Setup

Figure 2.2 shows a schematic representation of an ATM switch. A switch has a set of input and output ports that are used to receive and send out the stream of cells forming the ATM traffic. At the end of the initial call setup transaction, a series of switches are selected that bridge the distance between the source and the sink of the traffic to be generated. The switches agree on how the received cells will be directed from one switch to another (cells coming into port $i$ of the switch $P$ will be sent to output port $j$, which is in turn linked to input port $k$ of switch $Q$, etc.). All this information is stored in the *Switching Table* inside the switches. The switching table is a lookup table that retains the information as to which stream of cells coming through which VC and VP should be routed to which outgoing VC and VP. These switch entries are made into the switching table at the time of call negotiation.

Once the table entries are made, incoming cells need only contain a small (actually less than 4 bytes as shown in Figure 2.1) VPI/VCI address field. The switch will read this information and determine the subsequent VPI/VCI information for the cell based on the table entries, change the header entries to reflect the subsequent path information and route the cell through the correct output port. This system obviates the need for each cell carrying the complete destination address (that can

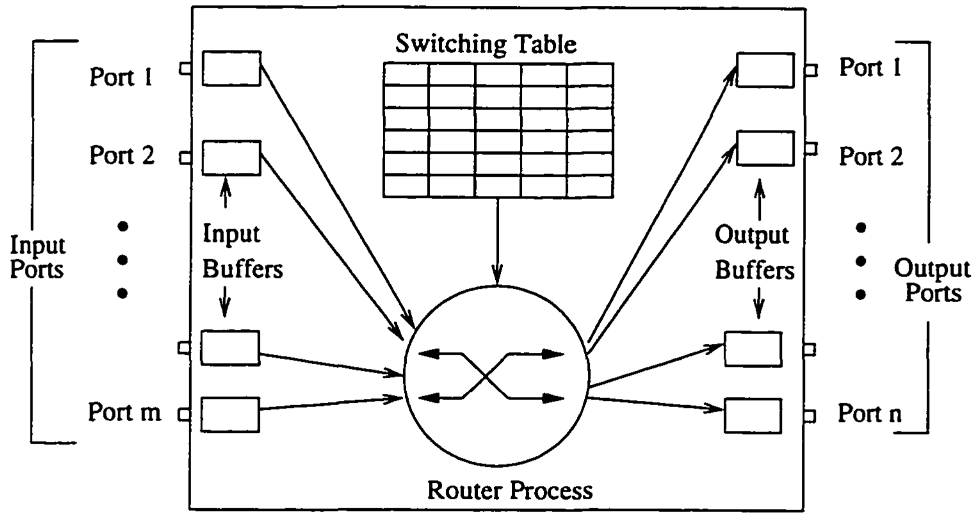


Figure 2.2: Schematic Representation of an ATM Switch

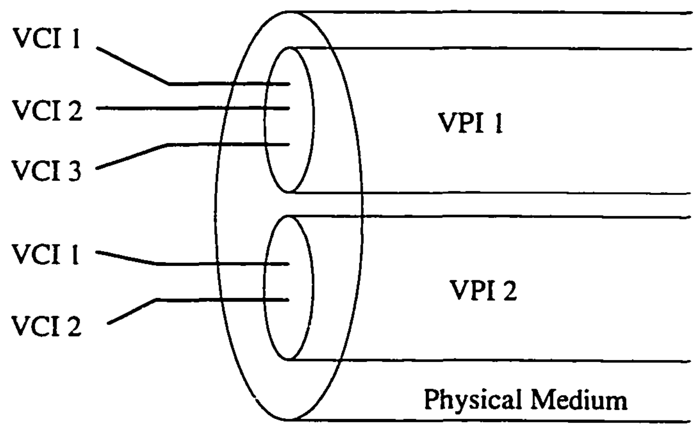


Figure 2.3: Virtual Paths and Channels

be approximately 20 bytes long) in its header. Switches invariably have some buffer storage added to their input and output ports so that they can handle some amount of congestion in the flow without resorting to dropping the excess traffic from the stream. As shown in Figure 2.3, the physical medium (a fiber-optic cable, for example) can contain several virtual paths in it. Each virtual path in turn may contain several virtual channels. The combination of physical media, VPI and VCI together define a given ATM link.

## 2.4 Quality of Service

The QoS factor, as the name indicates, is a measure of the quality of the connection. It involves traffic characteristics such as peak and average cell rates, burstiness, Cell Loss Probability (CLP), end to end cell delay, ease of call acceptance and Cell Delay Variance (CDV). Depending upon the kind of information transmitted, one or more of the factors gain importance. To give an example, a packet oriented connection, like file transfer protocol, will be affected by peak and average cell rate; alternatively, a CBR connection, like video transmission, will be affected more by the CDV factor.

### 2.4.1 QoS Classes

The ATM Forum UNI specification version 3.0 has defined five numbered QoS classes and sample applications that may require such levels of QoS. Table 2.1 summarizes these listings. Each class is defined by CLR for CLP type 0 and 1 flow, CDV for the aggregate (CLP type 0 and 1) flow and average delay for the aggregate flow. Network equipment designers are expected to define the ATM performance parameters for at least service classes A (circuit emulation, CBR video), B (VBR audio and video), C (connection oriented data transfer) and D (connectionless data transfer) from ITU-T Recommendation I.362 in a reference configuration that may depend on the physical length of the network links, etc. In the future, more QoS classes may be defined for a given service class. Presently ATM Forum has defined the QoS classes 1, 2, 3 and 4 as capable of supporting the performance requirements of service classes A, B, C and D respectively.

The QoS class 0 does not specify any objective that should be met. But services using the Class 0 may have specific traffic parameters. Still such traffic will be

Table 2.1: QoS Classes defined by ATM Forum

| QoS Class | QoS Parameters | Applications |
|---|---|---|
| 0 | Unspecified | "Best Effort", "At Risk" |
| 1 | Specified | Circuit Emulation, CBR |
| 2 | Specified | VBR Video/Audio |
| 3 | Specified | Connection-Oriented Data |
| 4 | Specified | Connectionless Data |

handled as *best-effort* service by the network operator. Such traffic contracts, that can adapt well to the varying resource availability situations, are still being defined under the names of Unspecified Bit Rate (UBR) or Available Bit Rate (ABR).

## 2.5 Motivation for the Study

CCITT Recommendation I.371 states the primary role of traffic control in BISDN is to protect the network and the user in order to achieve predefined network performance objectives in terms of cell loss probability or cell transfer delay [10]. It is easy to understand that in an ideal situation, where each one of the accepted traffic sources strictly adhere to the negotiated traffic parameters, traffic congestion will not pose a major threat to QoS. But due to the distributed nature of ATM network control as well as the enormous number of source and sink nodes that form the dynamic network characteristics, it is hard to hope for "stream-like" flow of traffic. CCITT itself states that "the network cannot rely on the user's compliance when declaring his traffic parameters" [7]. Thus, congestion control acquires added significance in the ATM model for two important reasons:

1. Most of the multimedia traffic ATM networks are expected handle is very sensitive to problems like CDV and packet loss caused by traffic congestion. Promising

good QoS and handling high speed traffic are two important virtues of ATM. Traffic congestion can quickly deteriorate the network performance making it unsuitable for such traffic.

2. The heterogenous mix of traffic sources with widely varying traffic generation characteristics and QoS requirements that need to coexist on the network, brings down the statistical probability that all the traffic sources will behave properly never exceeding the limits of the contract entered into during call initiation.

Therefore, policing mechanisms that prevent network buffer overflow and improve statistical multiplexing become inevitable. Following are a few required properties of ATM layer traffic control for B-ISDN.

**Flexibility:** the control effort should support a set of ATM layer Quality of Service (QoS) classes for existing as well as future services.

**Simplicity:** Simply the speed of the network precludes the use of elaborate, complicated and/or slow solutions. Implementation of the solutions should be simple, consuming very little time while in operation. In fact since the time available for error detection and correction is extremely small, ATM designers are forced to implement testing using hardware capable of handling such speeds, contrary to the use of software as is the case at present [29, 33].

**Robustness:** In the end, the control mechanism should yield high quality and quantity transmission, maximizing the utilization of network resources.

## 2.6 Traffic Management

Normally several sources of traffic are integrated before the consolidated traffic (albeit with a separate virtual channel for each source) reaches the public ATM net-

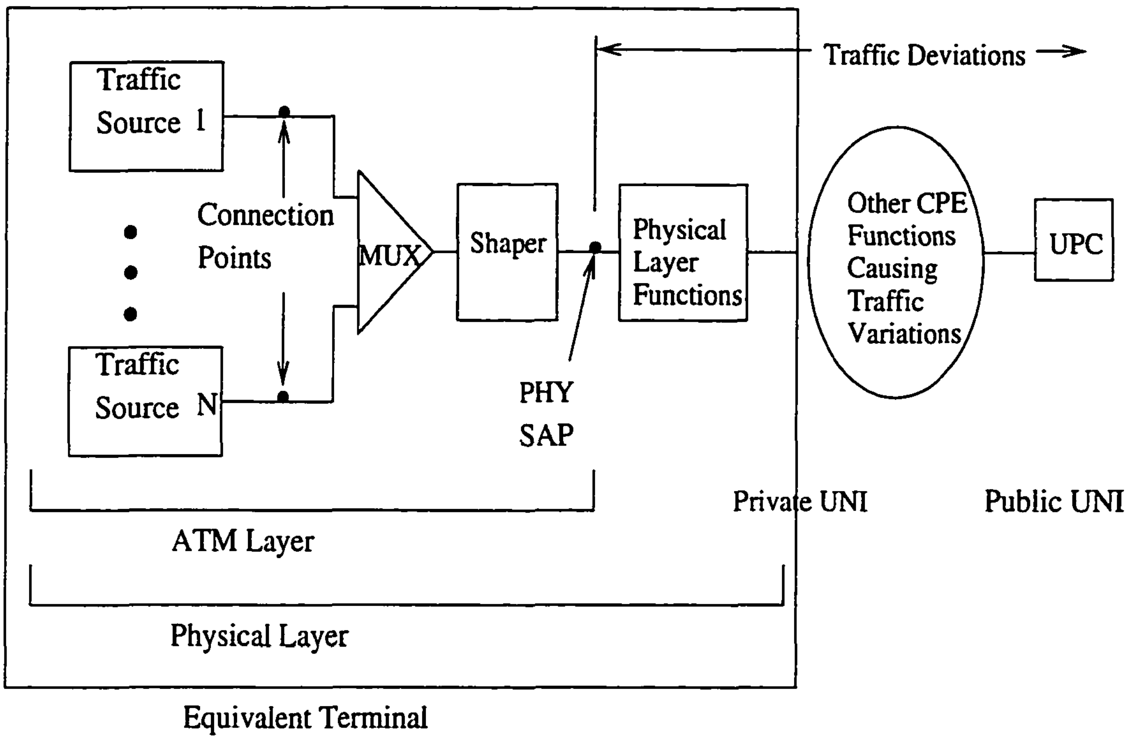


Figure 2.4: Equivalent Terminal Reference Model

work. This arrangement that integrates traffic from several sources, as depicted in Figure 2.4, is called an *equivalent terminal* reference model in the standards. It is generally not a real device but a collection of devices as shown in the figure. The traffic sources 1 through N (say N work stations) are connected to a multiplexer that hands over the traffic to a traffic shaper. The shaper ensures that the cell stream adheres to the agreed traffic parameters. In the OSI reference model, the output of the shaper corresponds to the physical (PHY) layer Service Access Point (SAP).

Even after being drawn through the shaper, some other physical layer function may modify the flow when it reaches a private ATM UNI resulting in non-conformance to the agreed traffic contract. This stream may subsequently be switched through other Customer Premises Equipment (CPE) before it reaches the public ATM UNI. As Figure 2.5 shows, the journey of the cells may encounter one or more intervening

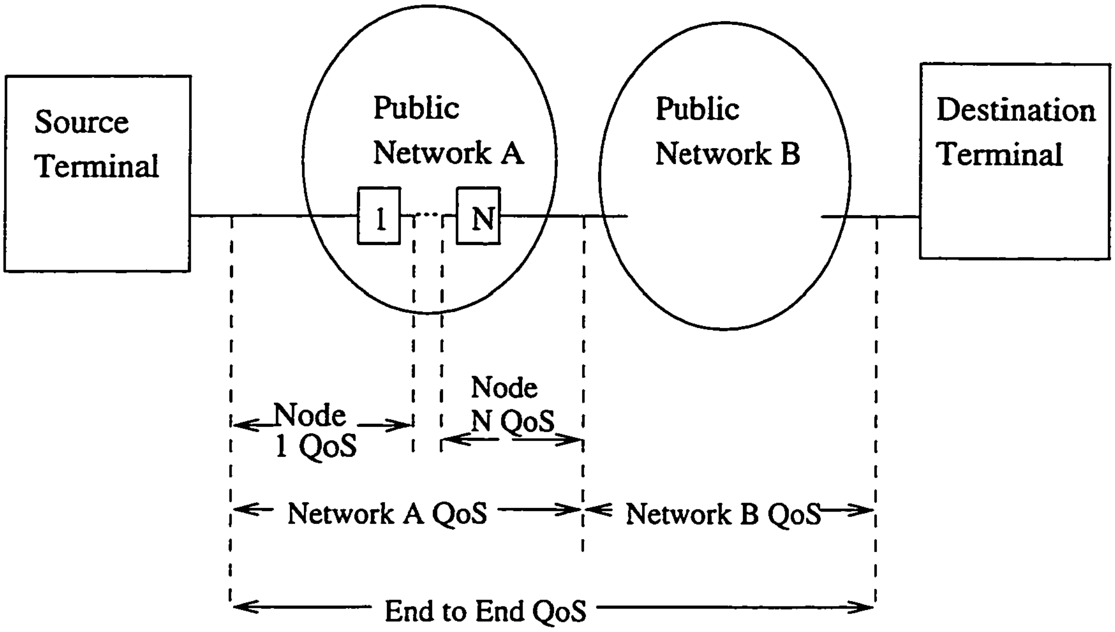


Figure 2.5: End to End QoS

networks before the destination is reached. Every one of these intermediate networks may introduce extra fluctuations in the flow, thus affecting the QoS. Since the end users are not required to understand the subtleties of the systems in between, they would always expect the best QoS transmission irrespective of the distance or systems separating the end points. So not only the multimedia and high speed nature of the traffic but also the subtleties of the numerous other possible trouble spots on the path contribute to the need for the development and implementation of very effective and efficient traffic management techniques.

## 2.7 Traffic Engineering

The field of Traffic Engineering deals with the modeling of various ATM switches and traffic sources so that performance analysis can be carried out.

### 2.7.1 Source Modelling

Traffic sources are invariably modelled using deterministic parameters with verifiable, cell by cell conformance to the designated traffic contract or probabilistic parameters that lend themselves to conformance measurements only over very long periods of time. Exactly which theoretical model represents what kind of realistic traffic type is still a hotly debated issue. To give an example, *memoryless process* or alternatively called *Markov Process* and a hyperexponential distribution based model used to model highly bursty traffic sources are two popular source traffic models used often in studies . But some recent studies carried out at Bellcore using actual LAN traffic show that these traditional models are too optimistic. In reality the LAN traffic are found to be *self-similar,* which means that the traffic has similar properties regardless of the time scale on which it is observed. This finding contradicts the traditional models where the traffic tends to become smooth and more predictable as the time span for the averages is increased.

### 2.7.2 Performance Measurements

QoS is the factor that is studied against all the other characteristics of the system. QoS may be measured at individual VPC or VCC or at higher levels for several VCCs or VPCs as a total aggregate. It is easy to understand that measurements at the individual VCC level are going to be resource-intensive to model, measure and compute. But it may be justified where very high levels of QoS are mandatory on specific links.

Parameters like CLR and CDV are used as a measure of QoS. We have carried out such sample studies to understand traffic and switch performance characteristics. The results are discussed in Chapter 5. While carrying out simulations, it is

important to remember the gains that may be realized when several traffic sources of varying characteristics are statistically multiplexed in real life situations, as the overall traffic will tend to be a lot smoother and so easier to handle than any of the individual traffic sources.

Another active area of research is the study of different types of queuing models that are used in the ATM networks. Queuing systems are usually represented by a five tuple $A/B/s(/w)(/p)$, where $A$, $B$, $s$, $w$ and $p$ represent Arrival process (Markovian, General or Deterministic), Departure process (again Markovian, General or Deterministic), Number of Queue Servers, waiting room (buffers) and source population, respectively.

## 2.8 Congestion Control

Congestion in an ATM network can be defined as the condition where the offered load (demand) from the user to the network is approaching or exceeds the network design limits for guaranteeing the QoS specified in the traffic contract [27]. Such a situation arises when the resources are overbooked or when a component in the network fails or when the source of traffic fails to abide by the agreed traffic contract. Thus, congestion may occur at different parts of the overall network such as switch ports, buffers, transmission links, ATM AAL processors, Connection Admission Control (CAC) processors, etc.

Congestion control can be broadly classified into three subcategories, viz., congestion avoidance, management, and recovery. Congestion avoidance mechanisms are designed to prevent and recover from congestion during periods of peak network loads. For example, in a situation where a network link or node failure occurs, avoid-

ance mechanisms are triggered to control the situation. Congestion management systems operate with the objective of ensuring that the congested network scenarios are never entered. Thus, allocation of resources, working under fully booked or bandwidth guaranteed systems, administering CAC, and using network engineering techniques, falls under the purview of congestion management. Congestion recovery procedures are used to ensure that any congestion that may be encountered will not result in severe degradation of user perceived QoS. Such procedures include cell discarding, modifying the UPC, FECN and BECN. Most of these procedures and techniques that may be found in network equipment today are ad hoc implementations carried out by individual vendors designing the equipment. The differences that exist are used as selling points by the vendors without much regard for standardization and interoperability. We will take a closer look at some of the techniques used for traffic control on ATM networks in the next chapter.

## 2.9 Theoretical Insight

Even before entering into the congestion management arena, gaining better insight into the problems at hand is bound to help network designers and managers utilize the resources more efficiently. Some of our work focuses on gaining such an insight into the problems of efficiently allocating the available bandwidth to a set of incoming calls, determining the bottlenecks in a network that has been a cause of severe congestion, and the like. Although the literature is replete with studies on queuing theory and small scale empirical analyses, studies aiming to provide theoretical insight into the existing problems are not found that often.

## 2.10 Virtual Circuit Setup Process

Another area of concern is the simple nature of bandwidth negotiations between a source and the network. Basically, the source communicates a standard set of parameters to the network which assigns a communication channel to the source. This virtual circuit setup process is repeated endlessly even if the nodes involved in the transmission are often the same pair. Constructing permanent virtual circuits is a plausible solution. But developing PVCs for large networks is computation intensive and most of the paths developed may never be used.

Once a VC is setup by the network, the source simply accepts the setup that was handed over by the network. The possibility of the source itself helping the system in channel selection has not been considered so far. In our studies we attempt removing these drawbacks by introducing a new method of VC setup that is robust and scalable while not very complicated to implement.

## 2.11 Summary

ATM is the only technology currently designed from 'the ground up' to simultaneously support image, voice, video, and traditional LAN and WAN based data services. Since the technology is fairly new, there are lot of areas that need to be defined well, improved and standardized. Since the operating principles of the ATM network seem to be well suited for future networking requirements, it is time to resolve these pending issues.

Technologies like SONET, Frame Relay, fast ethernet, etc. are posing stiff competition to ATM at present. Though some of these technologies are well suited to the world as of now (cheaper, easier migration paths), ATM seems to be better suited

for carrying a wide variety of traffic types which will be a fundamental requirement of networks in the not too distant future. None of the other technologies boasts QoS guarantees, scalability or capacity to handle very high speed traffic as part of their specifications. ATM does have problems in the areas of present implementation costs, steep learning curve and multicasting. But the basic idea of handling all types of traffic in the form of cells using a circuit oriented switching mode with guaranteed QoS makes ATM technology stand above the rest. So at this point it is important to work out the problems in the technology and make it better suited for the next millennium.

# Chapter 3

# Review of Literature

The problem of congestion control in B-ISDN networks has been an area of vigorous research during the past few years [13, 17, 30, 31, 34]. Commonly found congestion handling procedures identify the cells that are in violation of the agreed contract and tag them for possible discarding in the future. Algorithms used to identify the cells in violation are often not sophisticated enough to address all the concerns. This handicap is enhanced by the limited time window available for such processes due to the high speed of the traffic. Guillemin and Dupuis [18] have shown how cell clusters of a given connection can pass transparently through pick-up policing mechanisms in multiplexed environments, resulting in bursty traffic causing severe degradation of transmission quality. There are few techniques designed to handle bursty traffic conditions. Some of them emphasize controlling the source very strictly to prevent any unexpected traffic getting into the network [2]; others suggest improved bandwidth allocation algorithms that attempt handling bursty traffic more efficiently at the network level [5, 8]; there are some that attempt a mixed approach [38]. Since CCITT clearly states that the traffic sources may not always be obedient [7], control at the network level gains more importance. Solutions at the network level can be broadly classified into two groups, viz., *statistical approaches* and *operational approaches*.

## 3.1 Statistical Approach

ATM packet traffic is measured in terms of parameters like average cell rate, average burst duration, peak cell rate, etc. Statistical approaches use these parameters to regulate the flow. Whenever the set (or previously agreed upon) thresholds are exceeded, they trigger correction mechanisms like dropping excess packets. Unfortunately, for this approach to work properly, packet traffic should be monitored for prolonged periods of time making sure that the traffic codes have indeed been violated. If the corrective measures trigger too soon, they might choke the traffic frequently, unnecessarily. This inherently sluggish characteristic makes this approach rather unsuitable for real time applications, video transmission, for example.

## 3.2 Operational Approach

Operational approaches on the other hand, govern the traffic using preset rules. The algorithm clearly identifies a set of cells as non-conforming using set rules. Hence they are called *parameterized conformance-testing algorithms.* CCITT Recommendation I.371 specifies these rules, referred to as Genetic Cell Rate Algorithm (GCRA).

CCITT has defined two equivalent versions of the GCRA, viz., Virtual Scheduling (VS) and Continuous-state Leaky Bucket (LB) Algorithm. Presented with the same sequence of cell arrival times $(t - a, a \geq 1)$, both algorithms mark the same set of cells as conforming or non-conforming. Figure 3.1 shows the virtual scheduling algorithm. TAT stands for theoretical arrival time, $t_a$ time of arrival of a cell. Initially during the connection establish time, TAT is set equal to $t_a$. Figure 3.2 depicts the Leaky Bucket algorithm; $X$ is the value of the leaky bucket counter, $X'$

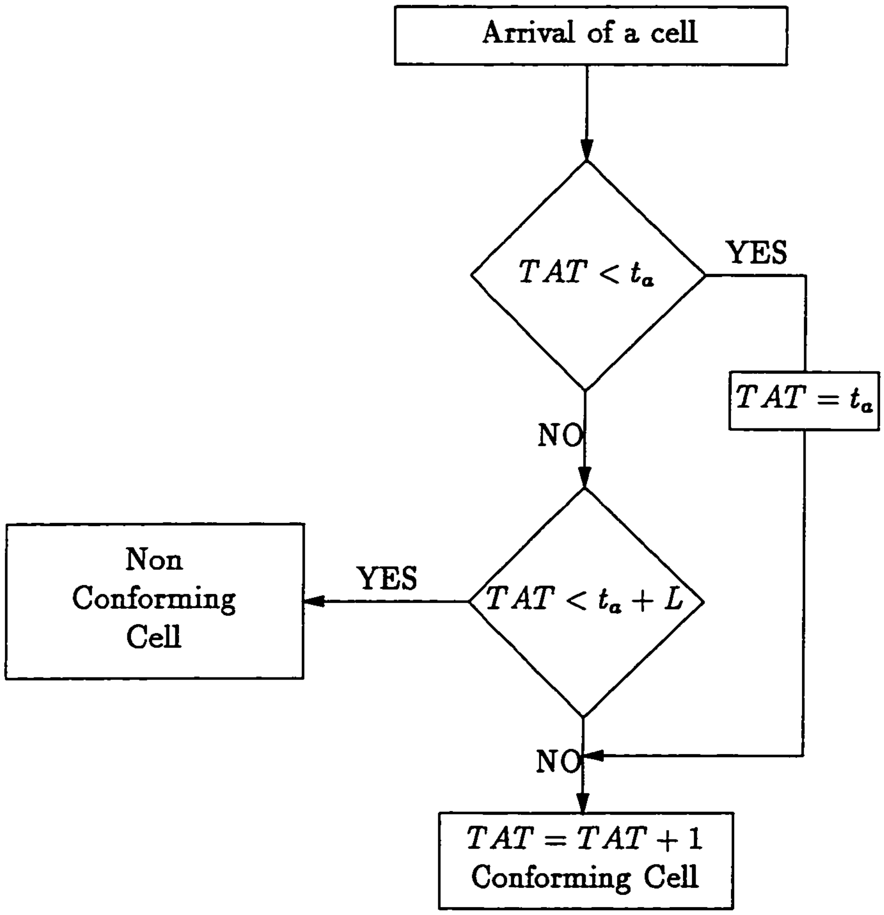


**Figure 3.1: Virtual Scheduling Algorithm**

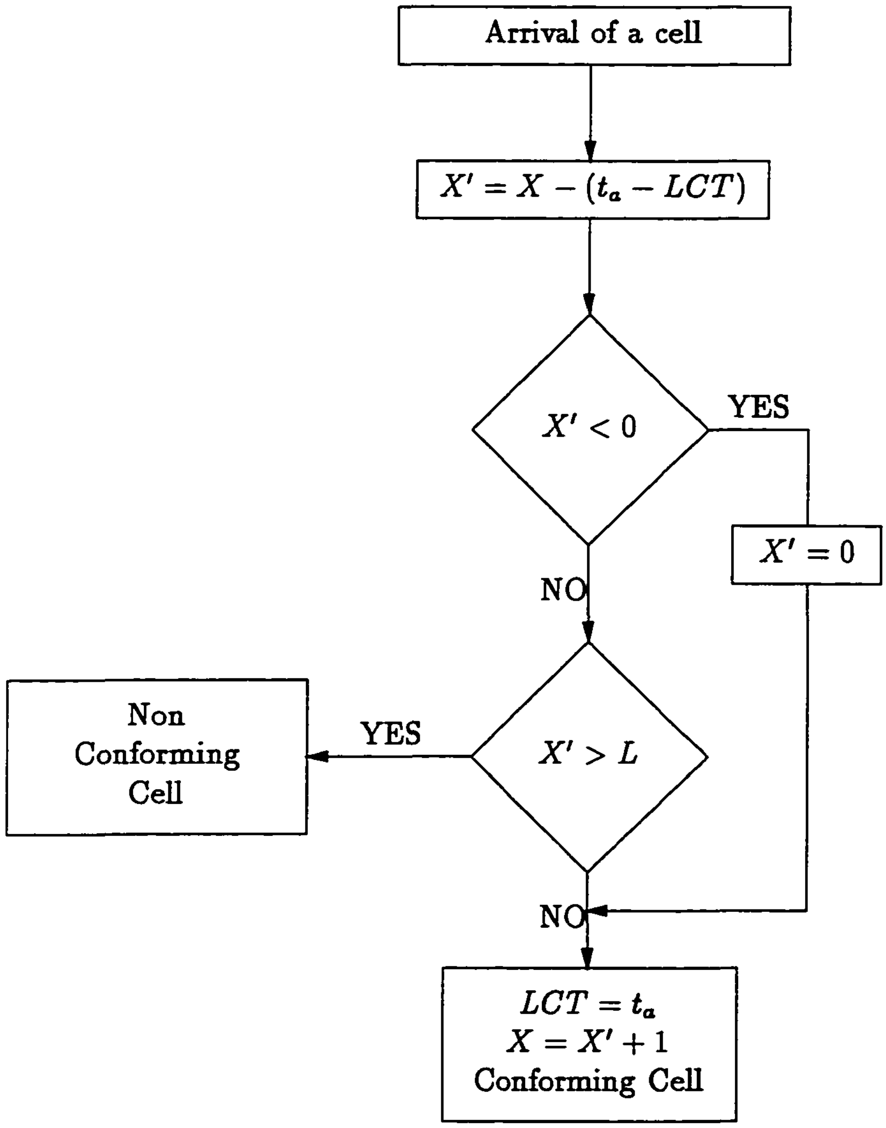


**Figure 3.2: Continuous State Leaky Bucket Algorithm**

is an auxiliary variable. LCT stands for the Last Conformance Time. At the time of arrival $t_a$ of the first cell of the connection, $X$ is 0 and LCT is set equal to $t_a$.

## 3.3 Performance Evaluation

The statistical approach to traffic control renders itself unsuitable for real time environments due to its sluggish nature in initiating correction procedures. Since it depends on continuously monitoring the traffic and drawing traffic violation inferences over a long period of time, it tends to start correction procedures late and continue them beyond the required point in time. In order to correct this situation, if the triggering thresholds are set too low, it tends to overcorrect the network and choke the traffic unnecessarily.

On the other hand, operational approaches discussed in Section 3.2 are quick to react. But the correction procedure is, in general, simply dropping the packets that are found to be in violation. Traffic sources depend on the expiration of a timer while waiting for an acknowledgement from the receiving end to understand that the transmission was not completed.

## 3.4 Traffic Shaping and Congestion Notification

Traffic shaping is a process in which a gate is introduced at the point where a private cell stream is about to enter a public network to control the private cell flow. The mechanism controlling the gate has the traffic contract that is agreed upon during the negotiation process. Using the bandwidth value allocated, the mechanism divides each second into $n$ time slots and allows exactly one cell to pass through the gate during each time slot. Thus the frequency with which cells enter

public network becomes well streamlined eliminating the possibility of unexpected bursts of cells entering the network causing congestion. Although this technique helps eliminate congestion in the public network, if the cell generation process is erratic in the private network, congestion or buffer overflows may be encountered frequently in the private network domain. Since such a situation may be much more manageable than a congested public network, traffic shaping techniques are invariably used near the sources to streamline the cell flow.

In spite of shaping techniques that are employed, cells may encounter congestion due to various other factors discussed before or perhaps due to other private networks that may not be employing shaping techniques. When congested areas are encountered, the cells may be tagged for possible future deletion. If the cells are not discarded, then, when the cells reach the destination, the traffic receiver will be able to understand that the received cells passed through areas of the network that were congested. In certain other implementations, even when the cells are dropped, the device dropping the cells may send an explicit notification to the destination saying that the cells meant for that destination were discarded. This technique is called *Forward Explicit Congestion Notification* (FECN). If a message is sent to the source of the cells when they are discarded or marked for possible future discarding, the process is called *Backward Explicit Congestion Notification* (BECN). The source and sink thus notified may take appropriate corrective action. Unlike FECN, BECN is harder to implement as this requires the establishment of a return path from the sink to the source so that the notification can be sent to the source from any point in the circuit.

Thus, there are a set of techniques that are defined in the literature, implemented in the networks or at least being studied. Well perfected traffic control techniques are still evolving and have a long developmental process in front.

# Chapter 4

# Bandwidth Allocation

A traffic source in need of establishing a communication link with one or more destinations initiates a request. Depending upon the design of the network, several different parameters that define the characteristics of the source traffic are communicated to the network during this initialization procedure. This is the same as lifting a telephone receiver and dialing a number we wish to reach. In an ATM network designed to handle audio traffic, it could indeed be a placement of a (local, interstate or international) telephone call. The network needs to understand the requirements for this call to go through successfully and determines if there is enough bandwidth available from the source to destination(s) to complete the call properly. The transmission conduit in such a scenario could be a fiber optic cable with a predefined bandwidth limiting its traffic carrying capacity.

In practical situations, there can be a series of audio, video and data calls with varying traffic characteristics being initiated by different sources simultaneously. Such a collection of a large number of calls will be vying for the available bandwidth in the fiber. Depending upon the bandwidth requirements and the rigor with which the network is expected to ensure a high QoS, the revenue generated by different calls may vary widely. Other than the revenue, there can be other factors making one call more important than another (an emergency medical video conference as opposed to a movie playing through the network for entertaining a household). Under such circumstances, the network needs to allocate the available bandwidth

effectively. This process of call selection for efficient bandwidth utilization is called the *Bandwidth Allocation Problem* or BAP. In this chapter we first prove that the bandwidth allocation problem in ATM networks is NP-Complete. Then we show that the genetic algorithm paradigm can be effectively used to handle the problem.

## 4.1 Bandwidth Management

In order to analyze the Bandwidth Allocation Problem, we assume that connection types are classified by classes $1, ..., K$. All connections within one class have identical traffic parameters (such as peak and average cell generation rate, for example) and identical QoS requirements. The call acceptance policy implemented in the network should consider the Link Capacity Criterion first. It can be described as follows:

**Problem** Given $N_1, ..., N_K$ connections of classes $1, ..., K$ are offered to a link of bit rate $B$, can the link carry them simultaneously so that the Quality of Service requirements are satisfied for all classes? [22]

In a conservative ATM solution, based on *peak rate* bandwidth allocation procedure $\sum_{k=1}^{K} B_k N_k \leq B_{max}$, where $B_k$ is the rate required for connections of class $k$ and $B_{max}$ the maximum bit rate that can be sustained in order to meet the most stringent QoS requirement among all classes, as we discussed in Section 2.4.1. Policies that are not peak rate allocation based are in existence [1, 11, 12]. But the advantage of increased call throughput that results from such policies might be compensated by the congestion or cell loss that may be encountered resulting in poor QoS whenever the sources start generating bursty traffic with cell flow close to the peak rate defined in the traffic contract. Even in the case of one single class, solution to this problem becomes difficult if the sources are bursty [22].

In addition to the Link Capacity Criterion, the network is required to consider the Connection Acceptance Criterion as well. It can be defined as follows:

**Problem** Given that $N_1, ..., N_K$ of classes $1, ..., K$ are present on a link of bit rate $B$, should a new call of class $k$ be accepted or not, given that the connection blocking requirements have to be met for all classes? [22]

The blocking requirement stated above is a rule that new calls that may hinder calls that are already in progress (due to the added strain new calls may pose on resources) should be blocked so that presently accepted calls can complete their transmission successfully. This is to ensure that accepted calls get a fair treatment until completion and are not preempted in favour of newer incoming call requests. Such a rule may or may not be implemented in a given network based on operational requirements. One can argue that a necessary condition to accept a connection is that the link can support it, namely, that the Link Capacity Criterion is satisfied. This is not sufficient, because accepting all connections as they are offered might cause a greater blocking probability for connections requesting large bandwidth. So in the next section, we pose and analyze a more comprehensive question taking the revenue generation factor also into consideration.

## 4.2 BAP is NP-Complete

In this section, we pose the the bandwidth allocation problem (BAP) in the ATM network model and prove that it is NP-Complete. Therefore, no polynomial time, optimal algorithm can be developed to solve the problem. The problem is defined as follows:

**Problem** Given that $N_1, ..., N_K$ connections of classes $1, ..., K$ are offered to a link

of bit rate $B$, can the network select enough calls to utilize the available bandwidth fully and generate the maximum possible revenue while the QoS requirements are satisfied for all classes?

**Theorem 4.1** *BAP is NP-Complete.*

*Proof:* We propose to prove the problem NP-Complete by the following two steps:

1. We first show that BAP belongs to the class NP.

2. Then we show that the Knapsack problem, well known to be NP-Complete, is a restricted version of the BAP and this simple restriction can be carried out in polynomial time. This implies that BAP is NP-Hard, hence, NP-Complete.

We initially redefine the BAP stated above more precisely.

*Problem Instance:* $N_1, ..., N_K$ connections of classes $1, ..., K$, bandwidth requirement $B_k$ and revenue generation value $R_k$ for each connection class, a bit rate $B$, and a revenue goal $R$.

*Question:* Can the network select calls generating revenue $\geq R$ and total bandwidth requirement $\leq B$ while satisfying QoS requirements for all classes?

It is easy to see that BAP $\in$ NP since a nondeterministic algorithm designed to solve the problem has to simply guess a collection of transmission requests and verify the following in polynomial time:

- check whether the selected calls can be accommodated in the available channel with bit rate $B$
- Check whether the generated revenue is more than $R$.
- Check if the QoS is satisfactory.

The Knapsack problem instance is defined as follows [16].

*Problem instance:* A finite set $U$, a "size" $s(u) \in Z^+$ and a "value" $v(u) \in Z^+$ for

each $u \in U$, a size constraint $B \in Z^+$, and a value goal $L \in Z^+$.

*Question:* Is there a subset $U' \subseteq U$ such that $\Sigma_{u \in U'} s(u) \leq B$ and $\Sigma_{u \in U'} v(u) \geq L$

Now we can show that the Knapsack problem is a special (restricted) case of BAP. The set of incoming calls in the BAP is the finite set $U$ defined in Knapsack, $B_k$ is $s(u)$, $R_k$ is $v(u)$, $B$ is Knapsack size, revenue goal $R$ in BAP is value goal $L$ in the Knapsack problem. We restrict BAP to instances with QoS requirements that are always satisfied. This simple reduction, known as Restriction, can be carried out in constant time. Hence, BAP problem as defined above is NP-Complete. ∎

Since the BAP is proven to be NP-Complete, there is probably no polynomial time deterministic algorithm that solves the problem optimally. So in the next section we attempt using the Genetic Algorithm technique as a tool to develop a solution that may be close to the optimal solution.

## 4.3 Genetic Algorithm

There are several computational paradigms in existence today that provide heuristic solution to problems that are difficult to analyze deterministically. Simulated annealing, Neural networks and Genetic Algorithms (GA) are some such techniques. Each paradigm boasts certain virtues and strengths that may be very useful in handling a set of hard problems. A special algorithm designed to handle BAP specifically or modified versions of simulated annealing or neural networks might compete well with the Genetic Algorithm in providing effective solutions for BAP. In general, simulated annealing works well with large size problems that does not require on-line solutions. Compared to GA, neural networks may be harder to implement

and modify frequently as the selection criteria for calls changes continuously. So we decided to experiment with a GA solution.

Genetic algorithms are inspired by the process of *Natural Selection* explained by Darwinian Theory in biological sciences. The *Theory of Evolution* explains the development of living creatures of profound complexity that exist today through the process of Natural Selection. It is the process by which the fittest (from the view point of reproduction and adaptation to changes in the environment) species thrive and multiply at the expense of other not-so-fit cohabitants. The theory of evolution requires only a few simple rules be applied for successive evolution to occur:

- The entities involved must be capable of reproduction
- Unfit entities in the pool should be eliminated
- Possibility of *mutation* must exist so that changes can occur

Although these rules may not look too complicated, since natural evolution has been taking place over an extremely long period of time (billions of years) and over an extremely large genetic pool of species (millions, if not billions), it has produced entities (animals, insects, human beings,...) of such exquisite characteristics. In biological entities, qualities that make a being fit or unfit for survival are encoded in the genes found in the entity's DNA. Thus, the values of the genes that make up the DNA of a biological entity defines all the characteristics of that entity. For example, a gene defining the claw of an entity may be set to *sharp* or *blunt*. The destruction of entities with blunt claws in a hostile environment and the survival of the entities with sharp claws in the same environment defines the *sharpness* value for the *claw gene* as desirable as opposed to the *bluntness* value. Since such unfit entities can not survive for long without adapting well, their qualities are all purged from

the gene pool. Thus, only successful, good qualities are passed on from generation to generation in the biological world. Since biological creatures reproduce with the occasional possibility of mutation, the process works very well in continuously refining the gene pool and in developing very successful beings that survive well on earth.

The Genetic Algorithm paradigm applies that technique of evolution on a much smaller scale to real world problems that may be too hard to tackle by deterministic means. The problem under consideration is mapped into the scenario discussed above by some creative means so that the virtues of the natural selection process can help one develop increasingly better solutions to a problem that initially looked unsolvable. In our problem of call selection, we consider each call that is accepted as a gene. So the gene can take one of the three possible values (audio, video or data call type). The string of genes that make up the chromosome reflects the collection of calls accepted for transmission. The revenue generation process is mapped to the environment. Thus, the actual revenue generated defines the fitness value (of the gene collection) that need to be maximized. The basic outline of a Genetic Algorithm is as follows [20]:

```
Initialize pool randomly
for each generation
{  select good solutions to breed new population
    create new solutions from parents
    evaluate new solutions for fitness
    replace old population with new ones with rare mutations
}
```

Although solutions developed initially by this means may be poor, due to the selection and mutation process, the good quality for the genes (that is, calls that deserve preference over others based on the evaluating criteria) will propagate through

the entire population eventually, resulting in excellent solutions in the end. But since the pool size and the number of generations that can be allowed for the development of a good solution in our problems are orders of magnitude smaller than the real world, solutions of Genetic Algorithms can not be as good as the the results produced by Natural Selection. Still it proves to be a very effective paradigm in handling NP-Complete problems that are otherwise very hard to tackle.

## 4.4 Call Selection

In this section we consider a situation where a router has to select and accept a subset of incoming calls for transmission so as to maximize the use of the available bandwidth. The maximization is defined by the value of revenue generated by the calls that are accepted. We assume that there are more incoming calls than the available bandwidth. The essence of the problem lies in selection of calls based on their lower bandwidth requirement characteristics as opposed to the revenue they generate per unit bandwidth usage. Since we proved this problem (BAP) to be NP-Complete in the last section, we develop a Genetic Algorithm based solution in this section.

In this study, we mainly classify the incoming calls into Type 1 (Data), Type 2 (Audio) and Type 3 (Video) calls. Obviously, more detailed study can be conducted with more types of calls under consideration. Calls vying for the bandwidth are a mix of all three types. We develop two greedy algorithms that allocate bandwidth to the incoming calls based on a specific criterion. We compare the revenue generated under this scheme to the revenue generated by the call allocation made as a result of the genetic algorithm. Results show that the allocation made by the genetic

algorithm is better or equal to the allocation made by the greedy algorithm for the situations we analyzed.

The genetic algorithm can learn from the variations that occur in the types of incoming calls during different times of the day or month and modify the selection criteria accordingly so that the utilization of the bandwidth is close to optimum most of the time. The down side is the extra computational resources, both time and computing power, required to execute the GA. But given a reasonably powerful processor, time requirement for even thousands of iterations can be kept under a few milliseconds, thus making it suitable for a real time scenario. Alternatively, the execution of GA can be carried out off line and the results can be used to modify the selection criteria in successive iterations of call selection. That is, a simpler algorithm that chooses the calls based on a few easy to use criteria can be used to actually select the calls, while the Genetic Algorithm is executed in the background to analyze the performance and to modify the selection criteria based on the inferences. Such a system would obviate the need for the call selection process to wait till the GA completes its execution each time before the selections can be made.

We used SUGAL (SUgarland Genetic ALgorithm), a Genetic Algorithm software package, to study the various possibilities [20]. We mainly classify the incoming calls into Video, Audio and Data calls. Presuming that there is a series of incoming calls of all three types, we ran the GA to pick a good selection of calls that would maximize the revenue generated by using the available bandwidth most effectively. Since this exercise is to illustrate the point that the genetic algorithm technique can be used to select the calls, we have used a simple example, so as to improve the clarity of discussion.

Table 4.1: Comparison of Performance

| Type of Algorithm | Data | Audio | Video | Bandwidth Used | Revenue |
|---|---|---|---|---|---|
| Hand Computation | 0 | 50 | 0 | 100% | $150 |
| Greedy Algorithm 1 | 100 | 0 | 0 | 100% | $100 |
| Greedy Algorithm 2 | 0 | 0 | 25 | 100% | $125 |
| Genetic Algorithm | 8 | 34 | 6 | 100% | $140 |

In the situation considered, the data calls require 1 unit of bandwidth; the audio and video calls require 2 and 4 units of bandwidth respectively. The revenue generated by each type of call is $1, $3 and $5 per call for data, audio and video respectively. In this scenario, the total bandwidth available for transmission is 100 units. Now Table 4.1 presents the bandwidth utilization and revenue generated information for solutions provided by two different greedy algorithms, the genetic algorithm, and a brute-force optimum solution. We note that the solution provided by the genetic algorithm is better than the ones provided by the greedy algorithms.

The listing below shows the parameters we were able to manipulate in the genetic algorithm to suit our requirements.

```
# Sugal v2.0 Output. Sunderland Genetic Algorithms, Copyright A. Hunter, 1994
# Sugal v2.0 Configuration
# System-defined parameter settings:

            alphabet_size  4
    annealing_temperature  10
          annealing_decay  0.95
                 autoseed on
                     bias  1
                crossover  onepoint
         crossover_points  1
           crossover_rate  1
                 datatype  integer
                 elitism on
```

```
                file_output on
           first_generation  500
                generations  500
                       init  uniform
               init_minimum  1
               init_maximum  4
                  init_mean  0
                    init_sd  1
               integer_size  8
                     length  100
                   mutation  step
        mutation_decay_rate  1
              mutation_rate  1
         mutation_rate_type  per_chromosome
              mutation_size  1
              normalisation  direct
                output_file  bap.out
                 population  500
              real_exponent  8
              real_mantissa  8
                 reevaluate off
                replacement  uniform
      replacement_condition  unconditional
           replacement_rate  1
      replacement_rate_type  proportion
replacement_tournament_size  2
              report_banner on
           report_diversity off
          report_every_pool off
          report_final_pool on
         report_fitness_max on
        report_fitness_mean on
         report_fitness_min on
          report_fitness_sd on
          report_generation on
        report_hamming_mean off
         report_pool_ranked on
          report_pool_stats on
   report_pool_stats_header on
            report_settings on
              screen_output on
            report_interval  1
                       seed  0
                  selection  roulette
  selection_tournament_size  2
```

```
# Sugal v2.0 Statistics

Gen Min Max Mean SD
00 0 125 0.25 5.5902
..
500 0 140 0.552 8.7194

# Sugal v2.0 Pool (ranked)

# Size 500 Length 100 Type integer
# -------------------------------------------
C00 [140,0.1804] data 8; audio 34; video 6.
# -------------------------------------------
```

## 4.5 Analysis of Results

We used a simple example above to illustrate the concept, so that the actual optimum solution could be computed easily by brute force. We can see that the bandwidth utilization is 100% in all the cases. The Greedy Algorithm 1 tried filling up the available bandwidth with as many number of calls as possible. Since data calls require the least bandwidth, all the available data calls will be accepted first. In our example, this results in 100 data calls getting accepted, generating $100 revenue. Greedy Algorithm 2 paid attention only to the cost of the calls. Since video calls produce the maximum revenue, all the available bandwidth was used to push video transmission resulting in 25 video calls getting accepted, generating $125 revenue. While doing so, it fails to note that audio calls, while producing lower revenue per call compared to video calls, generate better revenue since their revenue generation per unit bandwidth is higher than video calls.

The Genetic Algorithm on the other hand, can adapt very well to changing revenue generation parameters. In the static case we discuss, it outperforms the

greedy algorithms as we can observe from Table 4.1. It gave a solution accepting 8 data, 34 audio and 6 video calls generating $140 with 100% bandwidth utilization. Even in more complicated situations, depending upon the resulting value of the evaluation function, it can adapt and change the priority given to different calls. For example, depending upon the fee structure, it may give priority to one call type in the day time and another at night or accept more broadcast calls compared to one-to-one calls, etc. The attractive feature here is that no periodic intervention will be required to reprogram the algorithm since the genetic algorithm can directly learn from the successive revenue generated figures and adapt itself sufficiently.

### 4.5.1 The Gene Selection Process

Each chromosome in the GA pool represents a plausible solution to the given problem. It is made up of a collection of genes. While evaluating the fitness value of a chromosome, we need to ensure that there are no invalid genes found in the chromosome. To give an example, consider a chromosome of length $n$ where each gene making up the chromosome is a type of accepted call (meaning genes are represented by 1, 2 or 3). Now, when the chromosome is evaluated for fitness, depending upon the bandwidth taken up by the calls, the available bandwidth might get consumed by the first $n$-$x$ calls, making the last $x$ genes in the chromosome invalid (as accepting those calls will increase the total bandwidth requirement beyond the available value). So each chromosome in the pool should be made a valid gene using some mechanism. One idea suggested is replacing each invalid gene with a valid one manually after the completion of the GA run. For example, replacing the $n$-$x$ calls with call types taking up the lowest possible bandwidth or deleting them from the selection list so that the total bandwidth requirement is limited to the available

bandwidth. We feel that this idea spoils the spirit of the genetic algorithm. We used a dummy call type (Type 4), which did not contribute any revenue or bandwidth. The genetic algorithm while considering the incoming calls is presented with a large number of Type 4 calls as well in addition to the other three calls types. Thus, when the algorithm picked chromosomes, it also contained some genes of type 4 that simply filled up some gene space. We ignored the genes that were found in the chromosome once the allowed bandwidth of 100 is reached cumulatively. By this technique we avoid manipulating the chromosome/gene pool outside the execution of the GA.

Another way of correcting an invalid state would be ensuring in advance that any kind of chromosome generated will be legal. This can be achieved by developing a formula that maps each generated chromosome to a valid string of calls. Thus, since the mapping process ensures that each generated chromosome results in a valid string of selected calls, the problem of invalid genes is eliminated.

## 4.6 Summary

We showed that the problem of effective bandwidth utilization in the ATM network model is NP-Complete. We showed that the Genetic Algorithm paradigm can be used to solve the problem. Designing more efficient evaluation functions that will retain the characteristics of the calls received and selected for acceptance over a period of time would definitely be of more use since the acquired knowledge can be used in subsequent iterations of call selection.

If the calling terminal is forced to pay differing charges for different links it uses during a transmission, there will be another version of the BAP. Variations in

charge could be in terms of fee paid, congestion encountered, delay in call setup, etc. depending upon the resource availability on the network. Under such circumstances, routing calls through a path that will result in lower fee, lower congestion, less delay in the call setup process, etc. may be desirable. Allocating bandwidth taking such criteria under consideration could be equally difficult and may be handled using techniques presented in this chapter.

# Chapter 5

# Simulation of ATM on LAN

In this section, we carry out a *slow-motion* study of ATM traffic, so that the problems associated with ATM traffic management and congestion control can be analyzed. By slow-motion we mean simulating an ATM network using a 10base2 ethernet (alternatively called *ThinNet*, supporting just 10 Mbps bandwidth) and monitor software. The focus of the simulation is to ignore the speed factor that dominates ATM problems and concentrate on behavioral patterns of different traffic sources and sinks. In high speed ATM networks that carry traffic of the order of 655 Mbps, time available to analyze the flow in real time is so little that it precludes the possibility of using software based traffic monitors. So network designers are forced to introduce hardware based monitoring tools that can handle the traffic without disrupting the flow. So a slow-motion study of this kind is of enormous relevance when a switch to ATM from an existing 10base2, 10baseT (UTP) or slower network is being contemplated. Such popular networks carry data using copper cable that does not support very high bandwidth. They are also error prone compared to fiber optic cable based high speed networks. Implementing an ATM solution, that is meant for high speed transmission media, on such slow speed networks may not work very well due to some underlying assumptions of ATM networks (very low error rate in transmission, for example) not being satisfied. But a test bed of this kind helps us study the following:

1. Compatibility issues between the ATM protocol and the existing hardware.

2. Possibility of multiplexing several slow traffic sources into one high speed ATM channel using a multiplexing switch as the bridge.
3. Studying the congestion patterns based on the source of traffic alone, ignoring the problems introduced by the speed of the network.

## 5.1 Methodology

### 5.1.1 The Basics

An ATM network consists of traffic sources, sinks, a network that interconnects the sources and the sinks and switches that route the traffic from source to sink. During the initial negotiation process, the source requests the network for bandwidth allocation to communicate with the sink. At this time, from the source's perspective, the network can be represented by the first switch with which the source directly communicates. Accepting an incoming call, the switch that received the request from the source sets up a virtual channel from the source to the sink with possibly several switches in between. The virtual channel/path details are stored in the switches that are required to route the traffic. This arrangement obviates the need for individual packets traveling on the network to carry complete destination addresses that may be up to 20 bytes long.

From this description, we understand that simulation packages attempting to simulate an ATM network should have configurable switches, traffic sources and sinks and a network on which this simulation needs to be carried out. Since the simulation will be carried out on slower networks of the type 10Base2 or 10BaseT ethernets, the results of the simulation will give an insight into the characteristics of the traffic without the difficulties posed by the speed of an ATM network itself.

### 5.1.2 The Detail

In order to carry out the study, we used an ATM Switch Simulator software modified to suit our requirements [21]. The simulation package consists of a *Switch* module, different types of traffic generators, a traffic sink, and a switch control program that is used to manipulate the Switch. The Switch module has a manually configurable routing table, a set of input ports and output ports and a set of buffers each associated with individual ports or the entire switch. The module can be initiated with the required number of ports and the directions for channeling the traffic (i.e., traffic coming into the switch on port $i$ should be pushed out through port $j$, etc.). A copy of the Switch can be located on different machines on the internet. Thus, we can interconnect port $i$ on the switch (program) running on machine $A$ to port $k$ on the switch (program) running on machine $B$ across the internet. Setting up several switches of this kind on different machines on the internet and interconnecting the ports to set up virtual channels for traffic flow, allows us to set up and simulate as complicated a network as we want. There are three different types of traffic generators that generate constant bit rate, variable bit rate and random bit rate traffic. We carried out a series of simulations. The results are discussed in the next section.

### 5.1.3 Different Types of Traffic

In these simulations, we studied the effect of buffer size (storage space where cells are stored in a switch while they are processed for subsequent redirection) on the Quality of Service. In order to understand the effect of buffer size on different kinds of traffic we generated CBR, ABR and VBR traffic. The CBR or Constant Bit Rate traffic resembles ftp, email, and http types of traffic where the traffic generation is easily maintained at a constant cell generation frequency level. This type of traffic

is considered CBR because the traffic generation rate may get altered slightly by the speed of the hard disk that reads and sends out the data. Other than that there is not much variation in the generation process.

ABR or Available Bit Rate model is used to handle traffic that is willing to accept and make use of any available bandwidth on the network. This kind of traffic will mostly be the type that is not affected by variation in the available bandwidth and time delays. The VBR or Variable Bit Rate traffic, on the other hand, resembles video traffic. This is due to the fact that depending upon the nature of the video that is being transmitted, the cell generation rate may go up or down.

The main thrust behind the series of simulations we conducted is to understand the effects of various traffic types (both homogenous as well as heterogenous) and switch characteristics on the Quality of Service obtained. We used the CLR and the traffic processing delay at the switch as two factors used to represent the QoS. Processing delay is a good measure of cell delay variation or CDV that is important for time sensitive traffic such as video transmission.

In one study, we used three different types of traffic sources, one at a time, to simulate CBR, VBR and ABR traffic. Each traffic generator produced about 400 cells per minute. Since we wanted to have not just one but several traffic sources, we set up the switches with six input ports and six output ports. Each input port is dedicated to receiving cells from one specific traffic generator. This arrangement makes the simulation set up easier without any loss of accuracy as several traffic sources feeding one non-dedicated input port can be replaced by one super source providing the collective traffic feeding a dedicated input port. All the inputs received at the six input ports are directed to the seventh port in the switch. Port seven was connected to port eight so that all the traffic that reaches port seven is delivered to

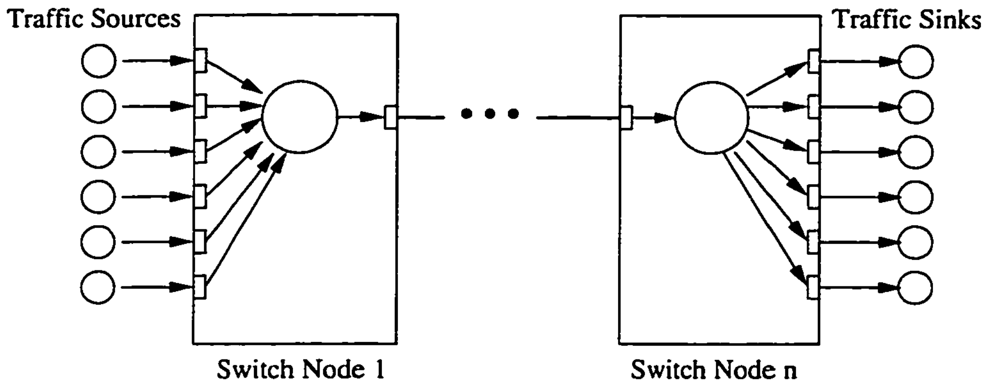


Figure 5.1: Schematic showing the Simulation Setup

port eight. Incoming traffic at port eight is split again into six parts and delivered to six different output ports. The collection of traffic from six input ports to deliver into port seven and the splitting of traffic on port eight to several output ports is carried out through the routing table entries made when the switch is initialized for the simulation. Six traffic generator processes were started and linked to the input ports and six sink processes were attached to the output ports. The schematic of this set up is shown in Figure 5.1. Port seven of a switch process running on one machine is connected to port eight of the switch process running on another machine (using the DNS names across the internet) to direct traffic flow across switches in different machines with the flow directed to sink processes in the last switch.

Once the processes are initiated, we waited for thirty seconds for the system to stabilize so that transient flow does not affect the observations made. Then we allowed the sustained flow to continue for five minutes, paused the switches, collected the required statistics and shut the processes down. The total number of cells that came into a port as well as the entire switch, the total number of cells that went out of a port as well as the entire switch, the number of cells dropped due to congestion at the port level and at the switch level, the average delay encountered

Table 5.1: Effect of Buffer Size on Cell Loss

| Buffer | CBR | | | VBR | | | ABR | | |
|---|---|---|---|---|---|---|---|---|---|
| | Drop | Total | %age | Drop | Total | %age | Drop | Total | %age |
| 5 | 8332 | 16703 | 33.28 | 11759 | 13119 | 47.27 | 8194 | 13095 | 38.49 |
| 50 | 7951 | 15999 | 33.20 | 8629 | 13303 | 39.34 | 5337 | 12173 | 30.48 |
| 100 | 8157 | 16511 | 33.07 | 7737 | 14676 | 34.52 | 4686 | 11938 | 28.19 |
| 150 | 7819 | 15935 | 32.92 | 7425 | 14919 | 33.23 | 3894 | 13577 | 22.29 |
| 200 | 8025 | 16447 | 32.79 | 7908 | 14184 | 35.80 | 3904 | 14219 | 21.54 |
| 250 | 7783 | 16063 | 32.64 | 5834 | 12520 | 31.79 | 3299 | 14325 | 18.72 |
| 300 | 8117 | 16831 | 32.54 | 5648 | 15043 | 27.30 | 2779 | 14135 | 16.43 |
| 350 | 7843 | 16383 | 32.37 | 6987 | 15502 | 31.07 | 2819 | 13498 | 17.28 |
| 400 | 7441 | 15679 | 32.18 | 8786 | 15428 | 36.28 | 2610 | 12666 | 17.09 |
| 450 | 7263 | 15423 | 32.02 | 7268 | 15283 | 32.23 | 2748 | 13754 | 16.65 |
| 500 | 7501 | 15999 | 31.92 | 7021 | 14923 | 32.00 | 2512 | 13641 | 15.55 |
| 1000 | 6393 | 14783 | 30.19 | 5181 | 13101 | 28.34 | 2206 | 13946 | 13.66 |
| 2000 | 4529 | 13055 | 25.76 | 4093 | 11309 | 26.57 | 921 | 12346 | 6.94 |
| 3000 | 3209 | 12415 | 20.54 | 2506 | 10800 | 18.83 | 242 | 14010 | 1.70 |
| 4000 | 1953 | 11903 | 14.09 | 1595 | 10529 | 13.16 | 0 | 13434 | 0.00 |
| 5000 | 953 | 11903 | 7.41 | 107 | 10652 | 0.99 | 0 | 12922 | 0.00 |
| 6000 | 0 | 11967 | 0.00 | 0 | 12387 | 0.00 | 0 | 14021 | 0.00 |
| 7000 | 0 | 11967 | 0.00 | 0 | 10202 | 0.00 | 0 | 13818 | 0.00 |
| 8000 | 0 | 12287 | 0.00 | 0 | 9945 | 0.00 | 0 | 14010 | 0.00 |
| 9000 | 0 | 12159 | 0.00 | 0 | 11280 | 0.00 | 0 | 12602 | 0.00 |
| 10000 | 0 | 11775 | 0.00 | 0 | 10434 | 0.00 | 0 | 13690 | 0.00 |
| 15000 | 0 | 12159 | 0.00 | 0 | 11295 | 0.00 | 0 | 12410 | 0.00 |

in the ports for the processed cells, and number of simulated time units were the list of statistics collected. By trial we found that the characteristics of the traffic flow was well captured in the observation once we ran the simulation for five minutes. Running the individual simulations for longer time periods increased the volume of total traffic handled in a linear scale without any difference in the QoS. This entire process was repeated for each buffer size and each traffic type.

## 5.2 Analysis of Results

Table 5.1 lists the results of a series of simulations. As it shows, we varied the buffer size on the port carrying this traffic from five cells to 15,000 cells. We used a total of 22 different buffer sizes listed in column 1. Since we wanted to study the effect of very small to very large buffer sizes, the sizes used are not evenly scaled across the entire range. Figure 5.2 shows a plot of CLR for the three traffic type for varying buffer sizes. As we can see from the plot, the cell loss ratio decreased overall as the buffer size of the handling port was increased. Since the CLR is more than thirty percent for very small buffer sizes, we understand that we may lose one-third the data transmitted if there is no buffer at the port level at all. This shows the importance of building buffers at the port levels in ATM switches. The increased loss experienced by the VBR traffic is explained by the burstiness normally found in the traffic of that kind that is more difficult for the switches to handle. The CLR of the CBR traffic is the easiest one to comprehend as it remains steady irrespective of the changes in the buffer size initially and declines almost linearly beyond a point. On the other end of the spectrum, ABR traffic provides the best QoS characteristics possible as its CLR remains lower than the other two all the time and reaches zero at the earliest, when the buffer size reaches 5000 cells. The plot indicates that the ABR traffic is the easiest one to handle (as one can expect) since the flow adjusts itself depending upon the available bandwidth. But in reality ABR mode is useful only for transmissions that are not hindered much by variations in CDV. This factor limits its use to regular data traffic. It does not work well with audio or video traffic.

Table 5.2 lists the resulting processing delay when the buffer size is varied. Figure 5.3 shows the CDV characteristics plot against varying buffer size for the three

Table 5.2: Effect of Buffer Size on Processing Delay

| Buf. Size | CBR | | VBR | | ABR | |
|---|---|---|---|---|---|---|
| | Drop %age | Delay | Drop %age | Delay | Drop %age | Delay |
| 5 | 33.28 | 5 | 47.27 | 4 | 38.49 | 4 |
| 50 | 33.20 | 50 | 39.34 | 43 | 30.48 | 33 |
| 100 | 33.07 | 99 | 34.52 | 83 | 28.19 | 70 |
| 150 | 32.92 | 148 | 33.23 | 128 | 22.29 | 106 |
| 200 | 32.79 | 196 | 35.80 | 179 | 21.54 | 133 |
| 250 | 32.64 | 244 | 31.79 | 223 | 18.72 | 182 |
| 300 | 32.54 | 292 | 27.30 | 259 | 16.43 | 226 |
| 350 | 32.37 | 339 | 31.07 | 316 | 17.28 | 268 |
| 400 | 32.18 | 385 | 36.28 | 369 | 17.09 | 301 |
| 450 | 32.02 | 430 | 32.23 | 416 | 16.65 | 387 |
| 500 | 31.92 | 477 | 32.00 | 458 | 15.55 | 574 |
| 1000 | 30.19 | 899 | 28.34 | 876 | 13.66 | 842 |
| 2000 | 25.76 | 1541 | 26.57 | 1483 | 6.94 | 1257 |
| 3000 | 20.54 | 1913 | 18.83 | 1598 | 1.70 | 1421 |
| 4000 | 14.09 | 1984 | 13.16 | 1567 | 0.00 | 1395 |
| 5000 | 7.41 | 1985 | 0.99 | 1714 | 0.00 | 1323 |
| 6000 | 0.00 | 1996 | 0.00 | 1777 | 0.00 | 1378 |
| 7000 | 0.00 | 1996 | 0.00 | 2057 | 0.00 | 1406 |
| 8000 | 0.00 | 2049 | 0.00 | 2089 | 0.00 | 1439 |
| 9000 | 0.00 | 2028 | 0.00 | 1622 | 0.00 | 1319 |
| 10000 | 0.00 | 1963 | 0.00 | 1797 | 0.00 | 1399 |
| 15000 | 0.00 | 2027 | 0.00 | 1567 | 0.00 | 1288 |

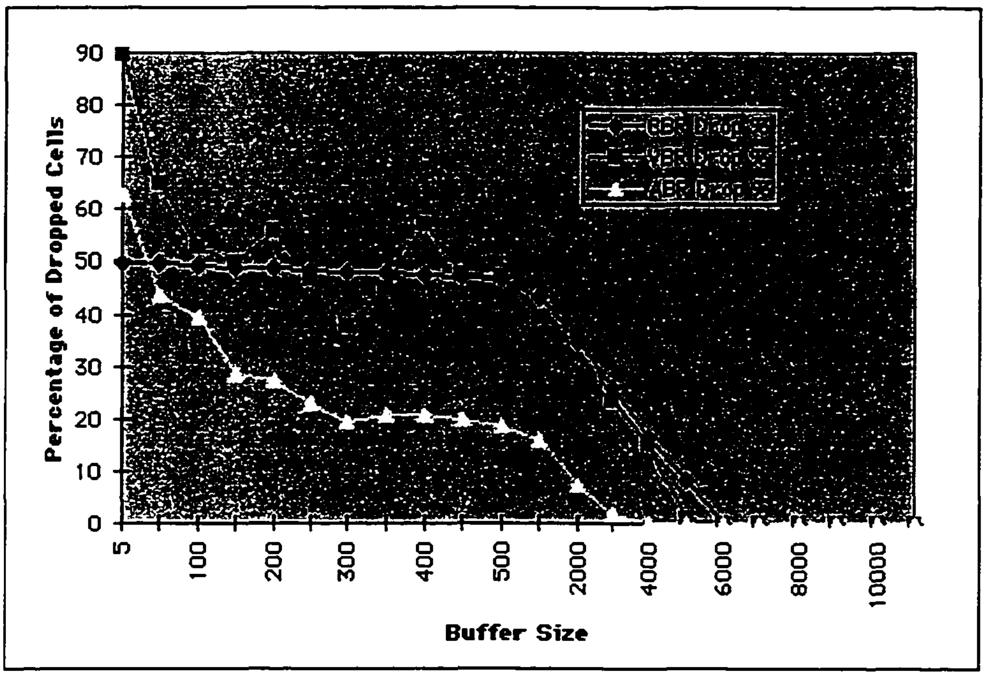


Figure 5.2: The Effect of Buffer Size on Cells Dropped

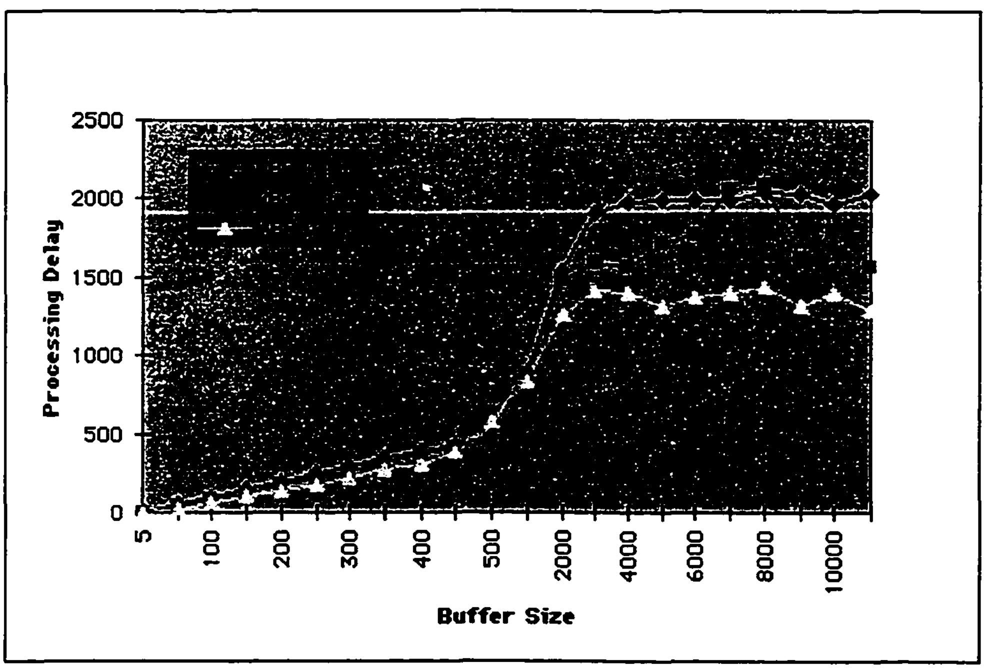


Figure 5.3: The Effect of Buffer Size on Delay

different traffic types. As we can see, the VBR traffic is the one most affected with the maximum delay, while ABR traffic is the least affected. Figures 5.4, 5.5 and 5.6 show both CLR and CDV together plotted against the size of the buffer for CBR, VBR and ABR, respectively. We can see that the delay experienced is the maximum in case of VBR. This indicates the difficulty involved in handling VBR traffic. On one hand the burstiness in VBR traffic suggests maintaining large buffers in the switch ports handling such traffic so that the variation in the rate of cell generation by VBR sources can be accommodated without significant increase in the CLR. On the other hand, large buffer sizes increase the CDV experienced by the traffic which is unacceptable for video and (to a lesser extent) audio traffic that belong to the VBR category. The only good solution to handle this difficulty is to ensure that there is enough bandwidth readily available throughout the entire traffic route and the switches are fast enough in handling these cells so that there is no requirement for large buffer sizes to reduce CLR. This realization leads us to conclude that when migrations are planned from lower speed networks to ATM, it is important to identify the areas of network that may generate a lot of VBR traffic and give priority in increasing the bandwidth in those links first. ABR traffic sources can be brought on board towards the end of the upgrading process.

Alternatively, when we introduced a smoothing function called (traffic shaper) in between the VBR traffic source and the port receiving the flow, the burstiness in the flow got ironed out making it CBR traffic. The extent to which a VBR flow is converted into CBR flow can be fine tuned through the design factors in the traffic shaper. Thus, a good traffic shaper implemented at the VBR traffic source itself will smoothen flow considerably making it more suitable for slow speed ATM networks that may not be able to handle congestion well.

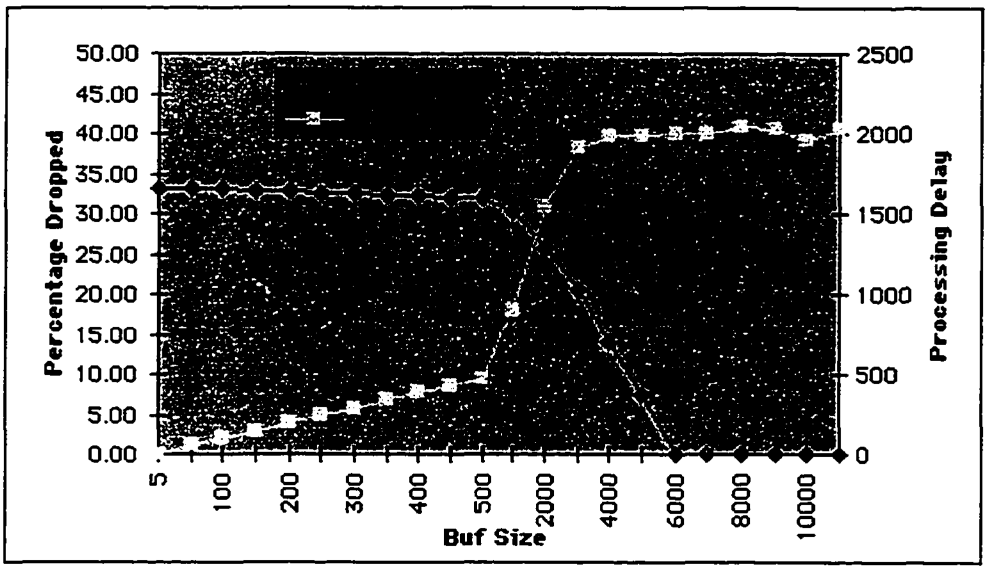


Figure 5.4: Cell Loss and Delay under CBR Traffic

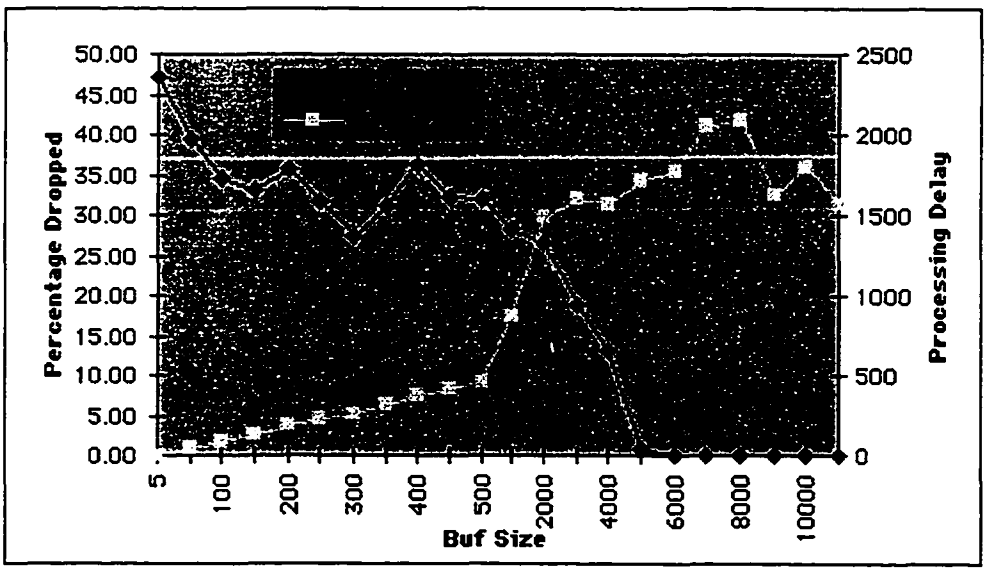


Figure 5.5: Cell Loss and Delay under VBR Traffic

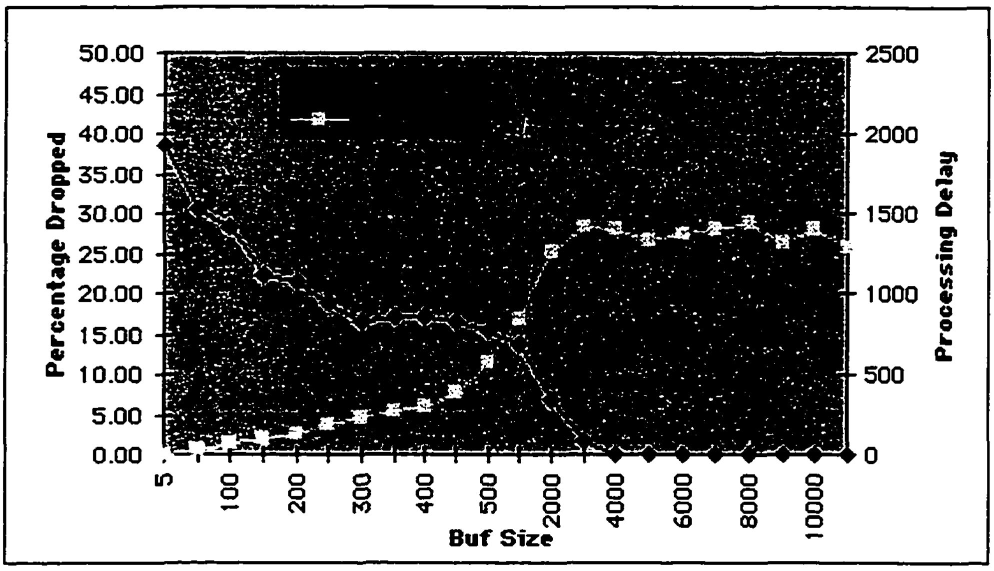


Figure 5.6: Cell Loss and Delay under ABR Traffic

### 5.2.1 Heterogeneous Traffic Condition

In addition to the studies based on homogenous traffic generators, we also wanted to analyze the effect of mixed traffic types flowing into and out of a switch. So in another set of experiments, we mixed the traffic sources feeding into the six ports of the ATM switch. Instead of all six sources generating either CBR or VBR or ABR traffic, we started two generators each of the three traffic types and fed all the generated flow into the six input ports of the switch. These results are shown in Table 5.3 and Figure 5.7. Although approximately 12000 cells were transmitted in each case (both under homogenous as well as heterogenous traffic conditions), the flow under mixed traffic conditions had much better QoS resulting in lower CLR as well as delay. This is explained by the fact that the bursts introduced by the VBR traffic were compensated by the ABR traffic, resulting in a much more manageable total traffic flow. This shows that even slow speed networks switching to ATM

Table 5.3: Effect of Buffer Size on CLR and Delay in Mixed Traffic Condition

| Buffer Size | Mixed Traffic | | | |
|---|---|---|---|---|
| | Total | Dropped Cells | Delay | %age Drop |
| 10 | 9999 | 2862 | 5.58 | 22.25 |
| 20 | 11763 | 2546 | 11.01 | 17.79 |
| 30 | 12320 | 2343 | 15.79 | 15.98 |
| 40 | 11523 | 1711 | 21.23 | 12.93 |
| 50 | 13092 | 1234 | 25.31 | 8.61 |
| 60 | 11993 | 1379 | 26.95 | 10.31 |
| 70 | 13234 | 1443 | 38.30 | 9.83 |
| 80 | 12472 | 924 | 38.59 | 6.90 |
| 90 | 12246 | 909 | 45.76 | 6.91 |
| 100 | 12146 | 768 | 49.53 | 5.95 |
| 200 | 13750 | 470 | 121.53 | 3.31 |
| 300 | 12494 | 0 | 113.66 | 0.00 |
| 400 | 13400 | 0 | 83.15 | 0.00 |
| 500 | 13243 | 0 | 154.07 | 0.00 |
| 600 | 12567 | 0 | 151.33 | 0.00 |
| 700 | 13896 | 0 | 105.99 | 0.00 |
| 800 | 13859 | 0 | 163.05 | 0.00 |
| 900 | 13102 | 0 | 123.84 | 0.00 |
| 1000 | 12926 | 0 | 124.57 | 0.00 |

networks may provide better overall QoS when the traffic sources are a good mix of different kinds.

## 5.3 Summary

An ATM network is a complex set up with several different parameters governing the traffic flow. What we tried to do in our simulations is to characterize the significant QoS factor variations with respect to different types of traffic. Our simulations can be modified taking into consideration factors that may be specific to one network so that the results obtained are more accurate and relevant to the network under

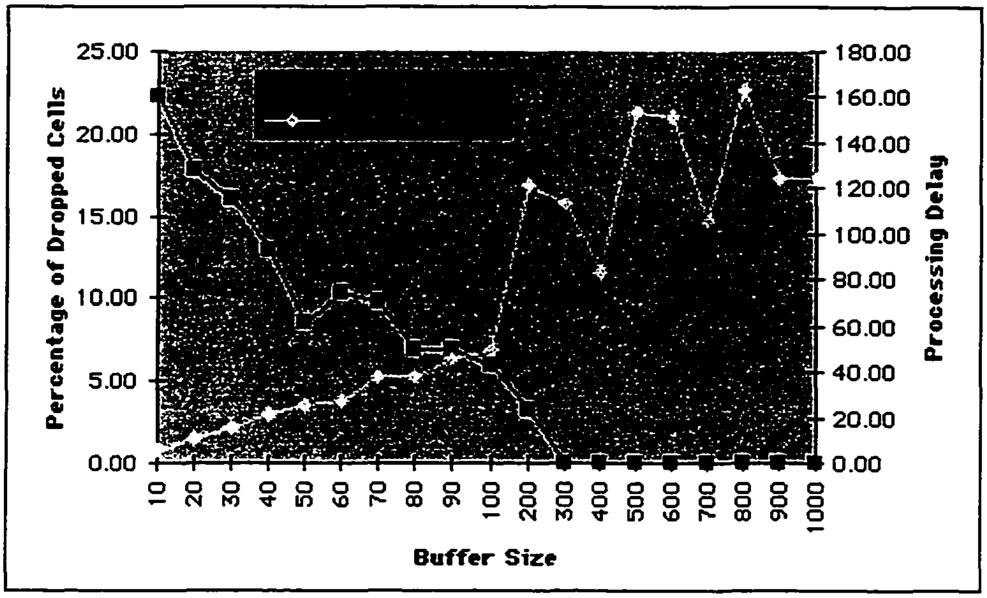


Figure 5.7: Cell Loss and Delay under Mixed Traffic

analysis. These factors could be the mix of traffic sources, the burstiness found in the generated flow, the number of switches the traffic has to pass through, individual port and switch specifications, etc. Simulating every possible combination is simply impossible due to the number of variations possible. It will not lead to any better understanding of the fundamental parameters either. So we have stopped our simulations after gaining an insight into some of the important characteristics.

Interpretation of the results can be summarized as follows:

- The buffer size of the individual ports in ATM switches plays a significant role in traffic congestion management.
- Results show that it is ideal to keep a large buffer in CBR traffic handling switches.
- VBR traffic that may be sensitive to time delays in the transmission gets adversely affected by large buffer sizes.

- This is due to the fact that although large buffer sizes may ensure that the cells are not dropped, they may introduce significant delay that may not be acceptable to certain types of traffic that are sensitive to time delay variation.

# Chapter 6

# Migration Planning

The superiority of ATM technology is widely recognized today. But costs involved in migrating existing legacy LANs to ATM remains prohibitively high. So when there are budgetary constraints, network designers are quite often required to implement such migration in phases. The goal of such efforts will be to identify and enhance the capacity of a minimum number of edges to realize overall improvement in the traffic flow. In this chapter, we analyze this difficulty and provide an algorithm to identify and prioritize network links that deserve a switchover to ATM. Our algorithm is based on a graph theoretic approach to identify flow congestion areas in a given network.

## 6.1 LAN to Directed Graph

Given an existing LAN, it can be represented in the form of a directed graph $G(V, E)$, where $V$ is the set of vertices, each representing a node on the LAN, and $E$ is the set of edges, each representing an existing link between two nodes on the network. Bandwidth available on each link can be defined as the edge capacity. The nodes in the LAN that generate traffic can be represented as the source nodes in the graph. Similarly, the traffic receivers in the LAN can be the sinks of the directed graph. Depending upon the actual usage of the bandwidth, each one of the edges in the graph can be either saturated or unsaturated. If we color the saturated edges red and unsaturated edges blue, existence of a blue path from the source to the sink

reflects the presence of an unsaturated path from the source to the sink. From a practical point of view, graphs where there are such paths are not of much interest to us as it indicates that the flow is not saturated and so none of the edges need capacity enhancement. But the algorithm we present still works on such graphs and identifies the bottlenecks assuming that the network is pushing the maximum possible flow.

Graphs in which all the paths from the source to sink contain red edges contain paths that are already saturated by the traffic flow. What is of interest to us is developing a systematic way to identify specific red edges that when converted into blue ones (i.e., made unsaturated by capacity enhancement) will increase the maximal flow of the graph significantly. In a flow graph with a maximal flow, not all edges may have flows equal to their capacity. The edges that have flows equal to their capacity are the bottlenecks, and are hence candidates for enhancement.

The problem of maximal flows in graphs as defined below has been well-studied:

**Problem:** Given a directed graph with capacities on the edges, determine the maximal flow possible from the source to the sink [4, 37].

We pose the following problem in the context of flow-graphs:

**Problem:** Given a directed graph with a source, a sink, and capacities on the edges (and therefore a maximal flow), identify the smallest set of edges such that increasing the capacity on each of these edges leads to an increase in the maximal flow of the modified graph.

This maximal flow of the modified graph is called the *enhanced flow* of the original graph. Before we present the algorithm to compute enhanced flow for a given digraph, we discuss the required preliminary details below.

**Definition 6.1** An edge for which the flow equals the capacity is called a *saturated* edge.

**Definition 6.2** A *saturated* graph is one in which all the edges are saturated. Otherwise, it is *unsaturated.*

**Definition 6.3** The *enhancement set of the graph* is the smallest set of edges such that increasing the capacity on each of these edges leads to an increase in the maximal flow of the modified graph.

If there are more than one set with the same (minimum) number of edges, then the *enhancement set* is the one that provides maximum increase in flow. If the increase in maximal flow is also identical, then any one of those sets can be named the enhancement set.

**Definition 6.4** The process of increasing the capacity of an edge is called *infinitizing.*

In reality, the term *infinitizing* might be a misnomer as upgrading a LAN link will increase the bandwidth of that particular link only by a finite amount and not to infinity. But the enhancement is expected to be substantial compared to the original bandwidth of the link. So in order to make the analysis of the graph easier, we consider this new bandwidth as infinity (as it is not expected to pose any bottleneck for the traffic flow until all the links of the LAN are upgraded to ATM).

**Lemma 6.1** If every vertex in a graph with a maximal flow satisfies the constraint that incoming flow equals outgoing flow, the graph is saturated.

*Proof:* Obvious. ∎

**Lemma 6.2** In a graph with a maximal flow, each path from the source to the sink has at least one saturated edge.

*Proof:* If not, then the flow can be increased along this path, and so the flow is not maximal. ∎

**Lemma 6.3** If the graph is saturated, then the edges on the shortest path constitute the enhancement set for the graph.

*Proof:* By definition, enhancement set is the smallest set of edges that need to be infinitized to realize increase in overall flow. In a saturated graph, the shortest path from source to sink contains the least number of edges that form the bottleneck. ∎

**Lemma 6.4** Enhanced flow of every saturated graph is the infinite flow.

*Proof:* Computation of the enhancement set in a saturated graph results in a list of all the edges found in the shortest path. When an entire path from source to sink is enhanced, the resulting flow is infinite. ∎

Since we are interested in upgrading as few edges as possible to realize the maximum increase in the overall flow, shorter paths from source to sink are better candidates. In addition, paths with many unsaturated edges are desirable since they may require capacity enhancement for only a few edges in them. Keeping this perspective, we may use the words enhance or upgrade (an edge) to mean the same idea of increasing an edge's capacity. We consider graphs with only one source and one sink, since graphs with multiple sources and multiple sinks can be reduced to the single source and single sink case easily [4].

## 6.2 A Congestion Locator Algorithm

This section presents a new algorithm called **Wave-Front** that identifies the areas of a network that present the most restrictive bottleneck for traffic flow. Once identified, if the capacities of these edges are enhanced, it will result in better overall traffic flow. It is loosely based on the Breadth First Search technique.

To explain the functioning of the algorithm intuitively, the search process exploring the edges can be considered as a wave front moving from the *source (S)* outward till either a saturated edge or the *destination (Q)* is reached. If a saturated edge is found along one of the paths, that path is not extended further until all the other paths also encounter a saturated edge. Thus, the paths are extended in synchrony, synchronised by the encounter of a saturated edge or $Q$. The purpose is to find all paths from $S$ to $Q$ with the smallest number of saturated edges. Therefore the paths are progressively examined and extended in such a manner that all of them have almost the same number of saturated edges (they may differ in at most 1 at any time). Once the destination is found along any path, the number of saturated edges to be enhanced is determined.

- $C$ - set of $n$-tuples of saturated candidate edges for enhancement. Each $n$-tuple corresponds to candidate edges in one path from $S$ to $Q$.
- $W$ - denotes the set of vertices forming the wavefront
- $adj(v)$ - denotes the set of vertices adjacent to vertex $v$
- $adj(W)$ - denotes the set of vertices adjacent to the set of vertices $W$
- $out(e)$ - denotes the vertex at the head of the directed edge $e$
- $in(e)$ - denotes the vertex at the tail of the directed edge $e$

- $out(v)$ - denotes the set of outgoing edges from $v$
- $out(W)$ - denotes the set of outgoing edges from the set of vertices $W$

1. begin {**wave-front**}
2. $W = \{ S \}$; $W' = \{ \}$; $MAX = 0$; $SET = 1$; UNSET = 0; $NUM\_SAT = 0$;
3. Scan all $adj(W)$: /* BFS */
   - if $(Q \in adj(W))$, goto step 6.
   - if $e \in out(W)$ is saturated,

     $\mathcal{C}_i = \mathcal{C}_i \cup e$ /* add $e$ to candidate list specific to this path $i$ */

     if $MAX = UNSET, NUM\_SAT{+}= 1$; $MAX = SET$;

     $W = adj(W) - out(e)$. /* do not extend this path */

     $W' = \{out(\mathrm{e})\}$ /* add this to the next wavefront */

   else $W = adj(W)$. /* make next set of vertices the new front */
4. if $W \neq \phi$ /* is non-empty */

   goto step 3.

   else /* first wavefront over, $Q$ not found yet */

   $MAX = UNSET$

   $W = W'$

   $W' = \{\}$

   go to step 4.
5. Continue until $Q$ is reached or no paths left to explore. If there is no path from $S$ to $Q$, exit with C = { }. /* As each iteration is completed, each $\mathcal{C}_i$ in $\mathcal{C}$ corresponding to path $i$ gets one edge added */

6. $Q$ reached, so $NUM\_SAT$ indicates the smallest number of saturated edges in any path from $S$ to $Q$. Continue algorithm till $W$ is empty. By this time, $\mathcal{C}$ is a set of $n$-tuples of edges that must be enhanced and the best tuple or subset of a tuple that yields the maximum increase in flow must be selected.

   /* For example, $\mathcal{C}$ = {{e1,e3}, {e5,e9}} So, infinitize $e1, e3$. Compute flow. Then infinitize $e5, e9$. Compute flow. Select max of the two. If there are common edges among the paths with least number of saturated edges, the least number of such common edges that provide an increase in flow should be selected. */

7. end {**wave-front**}

### 6.2.1 Proof of Correctness

In order to establish the validity of the algorithm, we need to show that:

1. The algorithm completes execution and provides a list of edges for every graph submitted.

2. The resultant list correctly identifies the minimum number of edges that yield maximum increase in the maximal flow of the graph when their individual edge capacities are enhanced.

3. The list does not include more than the minimum number of edges required.

The algorithm searches for all the paths from source to sink that are minimum in length. If there is no path from $S$ to $Q$, it is a trivial scenario and the algorithm stops listing an empty set as the list of edges to be enhanced. Step 5 of the algorithm handles this condition.

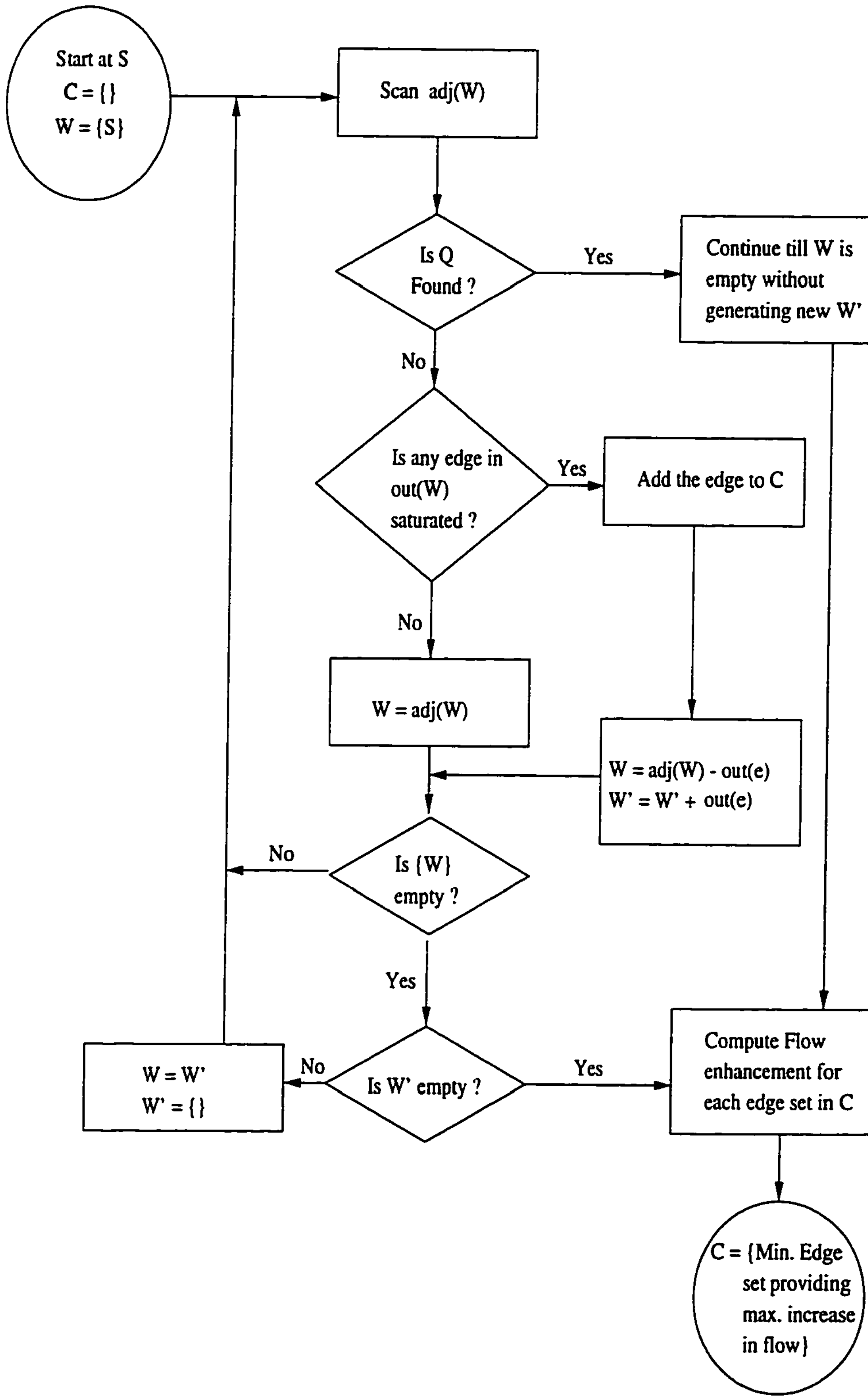


Figure 6.1: The *Wave-Front* algorithm for flow enhancement

In case there is a path, saturated edges in that path are the ones causing the bottleneck. Steps 3 and 4 in the algorithm iteratively searches the paths and lists those saturated edges individually for each path. Step 6 computes the flow by infinitizing each set of saturated edges and selects the edges that provide maximum gain. Again, all three conditions are satisfied. We note that in this scenario, the list can not be empty; if the edge capacities differ from each other, one or more edges with the lowest capacity get listed; if the edge capacities are all the same, all the edges in the path get listed as they all are saturated.

The $MAX$ variable is set and $NUM\_SAT$ is incremented at most once in each iteration (when a saturated edge is found). Thus, at the end of the search, $NUM\_SAT$ will hold the value of the least number of saturated edge found among all the paths from $S$ to $Q$. In a situation where there is more than one path of the same minimum length from $S$ to $Q$, the algorithm computes the increase in maximal flow achieved for the entire graph when one of the candidate paths is selected and all its saturated edges enhanced. This computation is carried out for each one of the candidate paths. Since the candidate paths are finite, and the number of incoming and outgoing edges on a vertex are finite, this computation will come to an end. The increase in maximal flow gives a clear indication of the extent to which saturated edges in each candidate path create bottlenecks. So when the least number of saturated edges that yield maximum flow enhancement is selected, it will contain the minimum number of edges that need enhancement.

Thus, in each scenario all the three conditions are satisfied and so the algorithm is valid. Well known BFS algorithm searching a digraph takes $O(m + n)$ time, where $m$ is the number of edges and $n$ the number of vertices. In the Wave-Front algorithm, we can avoid multiple visits to a path by marking the edges and vertices

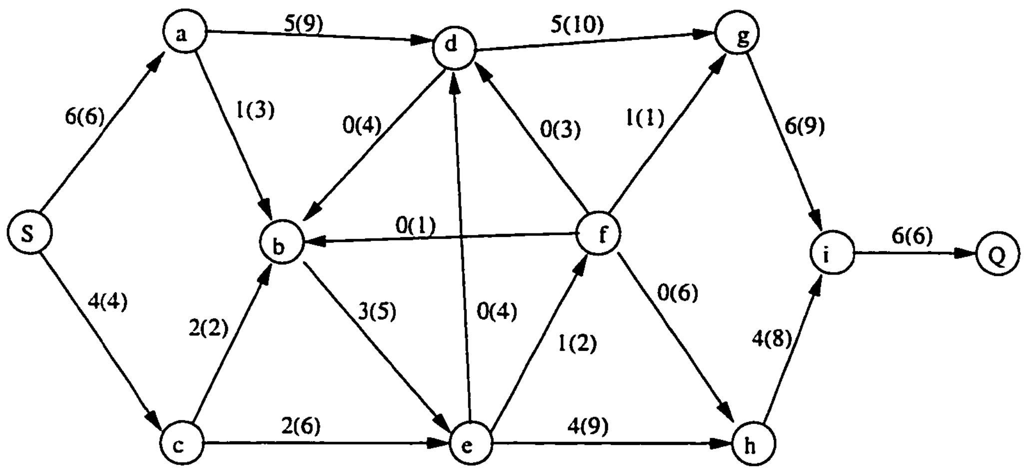


Figure 6.2: An implementation example.

visited and appending the results of the previous search to the existing paths of the newer searches whenever a previously visited path is encountered. Thus, the search for shortest paths from source to destination can be carried out in $O(m+n)$ time. The subsequent flow computation process will depend on the number of paths and saturated edges found.

## 6.3 An Illustration

Figure 6.1 presents the algorithm in a flow chart form. In this section, we consider a sample digraph, shown in Figure 6.2 and apply the algorithm discussed, to identify the minimum number of edges that need to be upgraded to realize an overall increase in the flow. The nodes $S$ and $Q$ represent the the source and the sink in the network, respectively.

1. Starting at the source node $S$, we search for the sink node $Q$ in the next level of the tree. It is not found. We reach the nodes $a$ & $c$ instead. So we proceed.

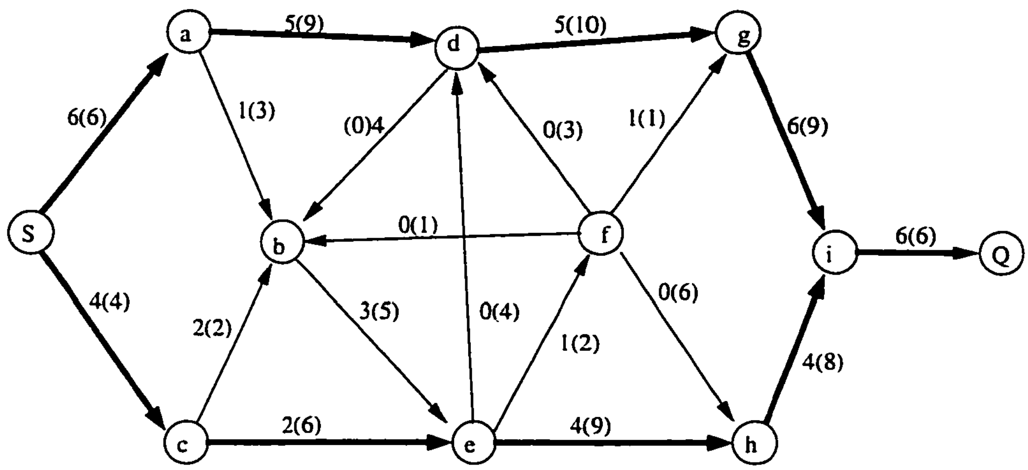


Figure 6.3: Chosen shortest paths.

2. We search the graph breadth first and look for paths leading from $S$ to $Q$. On the third hop a path ($Scbe$) with two saturated edges is detected. Search on this path stops until all the other paths encounter two saturated edges each.

3. On the fourth hop another path ($Scefg$) with two saturated edges is found. Search continues on other paths for discovering two saturated edges each or to reach $Q$.

4. On the fifth hop, we have the following paths: $Sadbed$, $Sadbef$, $Sadbeh$, $SadgiQ$, $Sabedg$, $Sabedb$, $Sabefd$, $Sabefg$, $Sabefh$, $Sabehi$, and $ScehiQ$. Out of these, $SadgiQ$ and $ScehiQ$ are the two shortest paths from source $S$ to sink $Q$.

5. Both the paths have two saturated edges, i.e., $Sa$ and $iQ$ in case of $SadgiQ$; $Sc$ and $iQ$ in case of $ScehiQ$.

6. Since the number of saturated edges in each path is the same, we compute the enhancement in flow that will be achieved when all the saturated edges in one path are enhanced.

7. $Sa$ and $iQ$ enhancement results in a flow increase of 6 units. $Sc$ and $iQ$ enhancement results in 4 unit increase.

8. The subset of edge(s) containing $iQ$ alone provides 4 units of flow increase.

9. Although enhancement along the $SadgiQ$ path provides better enhancement for the overall flow, it requires two enhancements compared to enhancing $iQ$ alone. Since we are looking for the least number of edges to enhance to realize an increase in the overall flow, $C = \{iQ\}$ which yields 4 units of flow increase.

Figure 6.3 shows the two paths that are chosen for consideration in Step 6 of the algorithm execution described above, in darker lines.

## 6.4 Identifying Fixed Number of Edges

The algorithm we presented identifies the least number of saturated edges that need to be enhanced to realize an increase in the maximal flow. There can be budgetary constraints that may warrant a search for specific number of edges that deserve an upgrade. For example, a designer might have the resources to enhance just one edge in the network and so may be interested in identifying one edge in the graph that will provide maximum increase in flow upon enhancement. A direct approach to the problem will be converting each saturated edge into an unsaturated edge, calculating the new flow and reverting it back to its original capacity. The edge that yields the maximum increase in flow would be the ideal one to be upgraded, i.e., upgrading the corresponding link in the LAN to an ATM link will result in the maximum increase in flow. In order to do this efficiently, we can use the concept of *minimum cutsets*.

**Max-flow Min-cut Theorem:** The maximal flow value from a source $S$ to a sink $Q$ is equal to the minimum of the capacities of the cut separating $S$ from $Q$.

Now the problem can be posed as follows:

**Problem:** Given a graph G, with capacities on the edges, select an edge $e \in E$, and replace it with an edge of infinite capacity such that the flow between all pairs of vertices is maximized.

*Solution:* To solve the problem, we can use the *Min-cutset* of the graph.

- Find the *Min-cut* edge set.
- For each edge in the set, compute the resulting enhancement in overall flow if that edge is chosen for upgrade.
- The edge that provides the maximum increase in the overall flow is the winning candidate.

Since the solution for single a edge case can be derived thus, let us consider a more complicated scenario of a series of chains. If we need to pick only one link for upgrading it to ATM, we use the *Min-cutset* concept. But if we need two links, we have to start afresh. So if we are trying to replace $x$ links:

1. Identify the link that directly connects source to destination, if one such link exists. If there is one, replace that link and stop.
2. If there is no such link and the number of links that need to be replaced is just one, carry out min cut-set, replace the links one by one by a higher bandwidth link and compute the total flow. The link that results in the maximum increase of network bandwidth is the one that deserves an upgrade. Stop.

3. If the number of links that need to be replaced is two, look for two hop chains from source to destination. If one exists, replace that two link chain and stop.

4. If there is no such chain, we can identify every two link combination possible. Sort all the combinations and replace the lowermost two link chain.

5. In the present situation, the serial links hold the key. We identify all serial links, sort the capacities and compute resulting flow for replacing two low capacity links in each serial link. The two low capacity edges can be in two different chains. This procedure is computationally expensive when there are three or more links to be replaced.

Since there can always be an $m + 1$ length chain that are saturated while we are looking for $m$ saturated edges for replacement, we may never be able to chose a given set number of edges (rather than an unknown minimum number).

## 6.5 Summary

In this chapter we considered the problem of upgrading parts of a large digraph network in the most efficient, cost effective way in order to enhance the overall flow. The algorithm we provided looked at the question of identifying the minimum possible list of edges that need to be upgraded, so that the network is exploited for its maximum possible potential. Considering a slightly different question where we have to pick exactly $n$ edges for capacity enhancement so that it increases the overall maximal flow can be much more complicated. Considering an efficiency factor

$\eta = \textit{increase in flow/Number of links chosen for enhancement}$

may give a clearer image of the cost and benefit scenario. If we contrast a path where

the successive links from source to sink have monotonically increasing capacities against another path of same length where the successive links from source to sink have monotonically decreasing capacities, need for such an efficiency factor becomes more important.

# Chapter 7

# Intelligent Negotiation

Transmission of data in ATM network begins with bandwidth negotiation between the source of the transmission and the network. The network examines a set of traffic parameters submitted by the source and sets up a virtual circuit between the source and the destination so that the requirements are accommodated. At the end of the transmission, the virtual circuit set up is torn down and the bandwidth is released for other transmission requests to use. This process is repeated each time a source and a destination have to communicate. The source and destination do not participate in this circuit set up/teardown process. We notice two areas that possess potential for improvement. The first one is the repeated virtual circuit build-up and teardown process that takes resources. Second is the fact that the two most important entities in a communication, the source and the destination, whose judgement on the quality of the transmission matters most, are not closely involved in the negotiation process. In this chapter, we propose two negotiation paradigms that ameliorate these conditions.

## 7.1 Adaptive Burn-in

We introduce the concept of *adaptive burn-in* for the virtual path set up and teardown process. When a call is initiated, a pre-defined set of call parameters are communicated to the network by the terminal initiating the call. The network then proceeds to set up a virtual path from the source to the receiver. Figure 7.1 presents

a schematic representation of this connection and transmission process. Setting up such virtual paths and then tearing them down at the end of each session takes computational resources. Studies are on to find techniques that would obviate the need for such repeated set up/teardown routines. One method suggested [6, 9] is defining virtual paths between every possible source and receiver ahead of time and storing that information in a lookup table. Empirical studies based on actual internet traffic [9] found that several wide area conversations are short ones. Such an understanding justifies the use of permanent virtual circuits (PVCs) in order to avoid the latency brought in by VC establishment. But this idea works only if the topology of the network (more specifically the set of nodes that are going to communicate frequently) is small and static. It has severe scaling problems.

What we propose is adaptive and so scales well and works better than rigid and inflexible precalculated PVCs. The scheme is presented in a flow chart form in Figure 7.2. Our system allows dynamic construction and deconstruction of VCs. But each VC created is recorded in the routers providing the path in a lookup table. The lookup table is of a fixed size and will be empty when the routers are first turned on. As new VCs are set up they also get recorded in the table on a FIFO basis. The time for which the detail of one VC remains in the table is determined by three factors. One is the physical time and/or availability of space on the table. The second one is a QoS rating provided by the source and the receiver at the completion of the call. Thus, at the end of each call, the source and receiver determine the quality of the call that was just completed and communicate a QoS rating to the routers enroute. If the QoS rating is very high, the corresponding VC is retained in the table for a longer period of time. In addition to these two factors, the frequency with which the source and sink communicate will also enhance the time for which the VCs remain

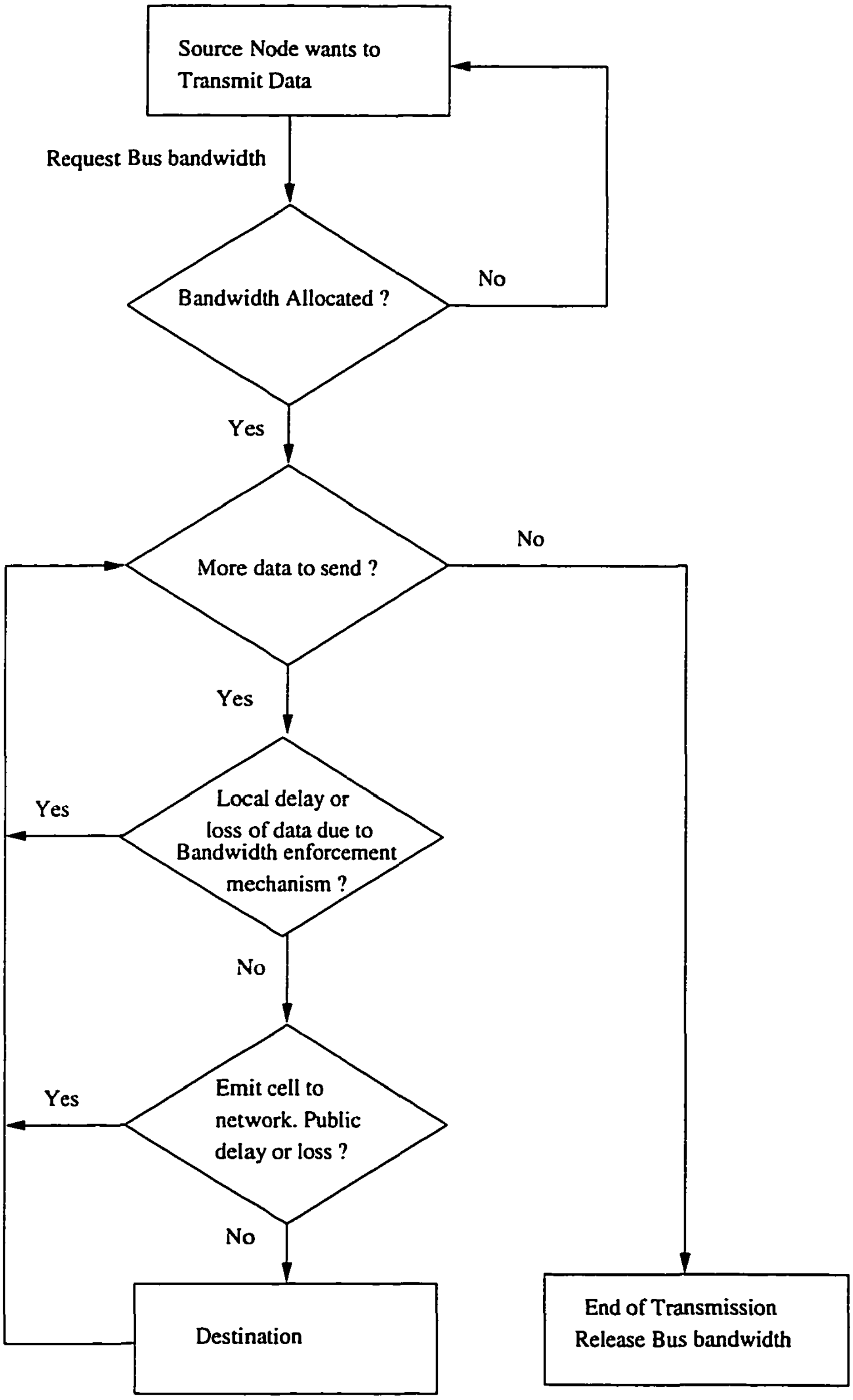


Figure 7.1: The Connection Process

etched in the table. Thus, if a VC set up provides good QoS and is used frequently, that path will be *burnt-in* into the table eliminating the need for recomputation of the same path again and again.

Due to changes in the traffic pattern, if the same VC starts to provide poorer QoS over a period of time, the poor QoS rating provided by the end points will reduce the said VCs *rating*, eventually eliminating it from the table. Since there is no resource intensive precomputation of VCs involved in this method, it is superior to the PVC method found in [6, 9]. Since VCs can be generated between any two nodes dynamically, this method scales well and is robust and adaptive to dynamic changes in the network.

### 7.1.1 Origin of the Concept

The concept of Adaptive Burn-in is based loosely on the idea of cache memory used in computer architecture design. In case of program execution, the inherent sequential nature of programs help in prefetching the next possible sequence of instruction. This advantage maintains cache hit ratio above 90% in reality. Although call requests coming into an ATM network can not be predicted that easily, the inherent adaptive qualities of the technique is bound to improve the performance with very little possibility for any degradation.

The term burn-in is inspired by the burn-in experienced in CRT screens that constantly or frequently display a same set of data. After a while, that display gets etched into the screen and remains visible even after the CRT is powered down. We felt VCs used frequently can be similarly etched into the routing tables of the ATM network, obviating the need for repeated recomputation. But since the entry can be

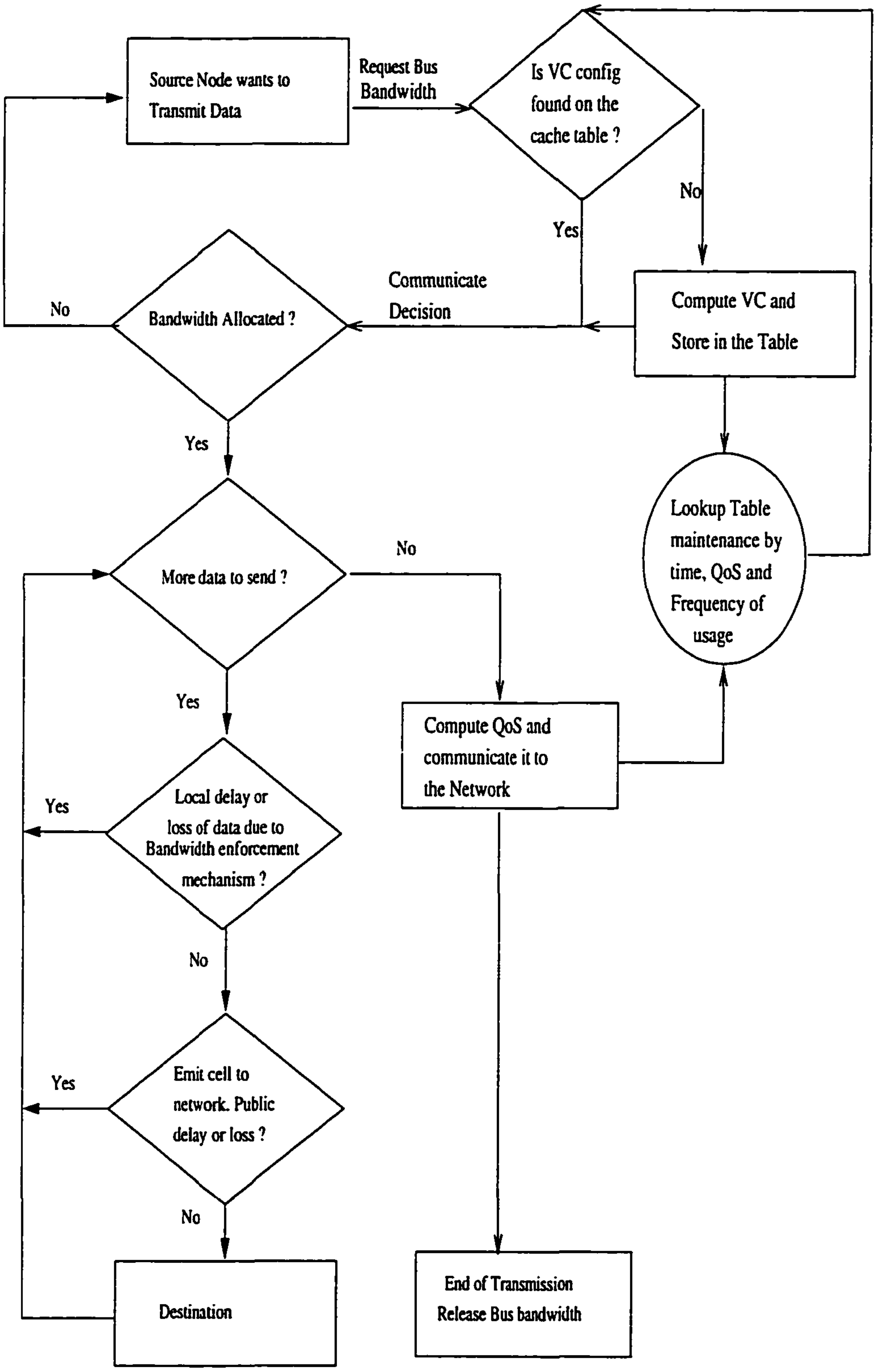


Figure 7.2: Modified Connection Process

removed from the table if it fails to provide good QoS in future, it remains adaptive (and not permanent).

## 7.2 Reinforced Learning

As we discussed briefly in the introduction, the two most important entities involved in the communication process are the traffic source and the destination. They are at the vantage point to judge the QoS of a call still in progress or just completed. Their satisfaction and judgment of the transmission should be an important factor defining the QoS standards. But this advantage is not used in improving the negotiation process.

We propose a model in which a learning mechanism is added to each source and destination node. This mechanism observes the negotiation and receives the virtual path and related details at the end of the negotiation. This can be achieved using *End-to-End Signaling* paradigm widely used in ISDN connections [34, pages 343 – 345]. It also monitors the cell delay or loss encountered during the transmission and computes Quality of Service (QoS) tightly based on the requirements of the source and destination. The learning mechanism is designed to draw inferences from the path used, the achieved QoS and other related factors. Based upon these inferences, this mechanism advises the source (as well as the network) and mediates the negotiation during subsequent transmission attempts to improve the QoS. The overhead due to the learning mechanism does not impair the transmission as its time intensive computations (analyzing the QoS of the completed calls and drawing simple inferences) can be carried out off-line. This approach provides an opportunity for the source to ask for channels that have higher probability of possessing free

bandwidth (rather than accepting whatever channel it is handed out by the network). The advantages of implementing such a mechanism would be better QoS and faster virtual circuit set up.

Since as a whole the network will be involved in processing too many call requests in a given period of time compared to individual sources and sinks, it will be easier for sources and destinations to store some of the virtual path and QoS related records of the past that can be used for setting up VCs in the future. This paradigm is based on the concept of reinforced learning [19].

While suggesting such an approach, we need to make sure that several entities that may participate in the negotiation process will not land the entire system in chaos. We can ensure that the system will not be lead into such chaos by limiting the extent to which the sources and destinations are allowed to manipulate the VC set up process. In a simple set up, the sources may be designed to submit a preferred VC to a specific destination (that provided good QoS during previous transmissions) while submitting the traffic parameters during the call initiation process. Depending upon the availability of such a VC, the network may or may not entertain the specific VC request by the source. If the request is not granted, the source will not have any means of acquiring that specific VC. It may simply refuse to start the transmission on the new VC or accept the new VC, carry out the transmission, compute the new QoS, and add the newly acquired knowledge to its data bank. If the new VC turned out to be a better one than the one the source requested, it may switch its preference to the new VC henceforth. Alternatively, if it determines that the new VC is worse than the one requested, the source may refuse to carry out the transmission in future when it is offered the same VC once again.

From the very unobtrusive model described above, to a much more invasive one in which the source is always granted the specific VC it requests, different negotiation models can be defined from the same principle. For more invasive models a premium in the form of a higher network access fee can be imposed so that the privilege is justified for calls that deserve such treatment. Since several models of different invasive levels can coexist in the same network, the overall network functionality can still be sustained.

## 7.3 Summary

We observed that there is possible resource wastage in the VC set up and teardown process. We also observed that the sources and destinations may have better knowledge of their own QoS requirement that can be taken advantage of in VC set up process. So we introduced a concept called *adaptive burn-in* to reduce unnecessary VC buildup/teardown processes. We also proposed a new negotiation paradigm that can help sources and destination tailor the VCs used in the communication to better suit their needs. The first concept approaches the problem from the network end; the second from the source and destination end. When implemented together, these two concepts should help improve the negotiation process.

# Chapter 8

# Conclusions

This research effort focused on ways and means to improve the traffic management and congestion control techniques in the ATM network model. Specifically it focussed on four areas of paramount interest.

1. The Bandwidth Allocation Problem
2. Simulation of ATM traffic on slower speed networks
3. Migration Planning
4. Better Negotiation Models

Chapters 4, 5, 6 and 7, respectively, discussed these problems in detail and provided new solutions to handle the problems with better ease.

## 8.1 Traffic Management Issues

A brief quote from the book **ATM: Theory and Application** [27] should summarize unresolved issues in the ATM traffic management arena facing the industry. "A large number of published technical articles describe the complexities, unsolved (or unsolvable) problems, issues, and proposed solutions on the general topic of traffic management. The fact that this book dedicated five chapters to this subject is evidence of the complexity and importance of this topic... The problem of achieving LAN-like flow and congestion control over ATM will take longer to solve, and is a

critical issue for the success of ATM. ATM Forum is focusing on this issue as a high priority. If the solution developed by the ATM Forum balances complexity against optimal performance and achieves industry acceptance, then a major step towards the goal of seamless networking using ATM will have been made. The problem of determining Connection Admission Control (CAC) procedures to implement a network to provide multiple QoS classes will be a challenging one. The ability for a network provider to perform this balancing act will be a competitive differentiator." Our work in this research effort, hopefully, contributes some positive ideas towards addressing these issues.

## 8.2 Future Work

As we have been emphasizing all along, ATM is still a fledgling technology. At this time it is very well suited to carry multimedia traffic at very high speed across networks. Although there are very strong contenders in the form of Gigabit networks, fast ethernet, etc. for multimedia traffic hauling networks, such technologies do lag behind ATM in one or more areas. For example, gigabit networks do not possess the QoS guarantees found in ATM. Fast ethernet, though as of now more economical compared to ATM, does not work very well at very high speeds (600 Mbps and more) compared to ATM. So the ATM technology has a lot of potential and is well set to become the dominant network technology in the early twenty-first century.

Work carried out in this research can be carried further in several areas. The genetic algorithm discussed in Chapter 4 can be expanded to analyze the performance taking into account more subtle characteristics of ATM traffic. Effects of

QoS achieved on successive call selection processes is in itself an interesting area of study.

There are several commercially available software/hardware packages in the market that allows very detailed simulation of ATM operations on smaller LANs. It is predicted that due to economic reasons, ATM technology will be used more and more on the backbone networks initially, rather than desktops. Simulation packages will allow the network managers to understand the pros and cons of deploying ATM on their network well in advance. This will allow them to justify the expense or make a decision to postpone the deployment to a later date. More realistic simulations than what we did for this study can be carried out with trace driven traffic that may give better empirical understanding of the traffic characteristics of a network under consideration.

Similarly, better migration planning algorithms and improved negotiation techniques between the end terminals and the network will obviously improve the efficiency of resource utilization. Theoretical standards development work is supposed to restart by 1998 when the industry completes the *catching-up* process. So this might be the ideal time to understand the issues still unresolved and get on board.

# Bibliography


[1] H. Ahmadi and R. Guérin, "Bandwidth Allocation in High Speed Networks Based on the Concept of Equivalent Capacity," ITC Specialist Seminar, NJ, Oct. 1990.

[2] M.E. Anagostou, M.E. Theologou and E.N. Protonotarios, "Cell Insertion Ratio in Asynchronous Transfer Mode Networks," *Computer Networks and ISDN Systems,* 24(4), 15 May 1992, 335–344.

[3] G. J. Armitage, "The Application of Asynchronous Transfer Mode to Multimedia and Local Area Networks," Ph.D. Thesis, Univ. of Melbourne, Australia, Jan. 1994.

[4] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications,* The Elsevier North Holland, Inc., 1976, 191 - 211.

[5] P.E. Boyer and D.P. Tranchier, "A Reservation Principle with Applications to the ATM Traffic Control," *Computer Networks and ISDN Systems,* 24(4), 15 May 1992, 321–334.

[6] R. Cáceres, "Multiplexing Traffic at the Entrance to Wide-Area Networks," Ph.D. Dissertation, University of California, Berkeley, December 1992.

[7] CCITT Temporary Document 43, Com XVIII/8, "On Networking and Resource Management," Matsuyama, Dec. 1990.

[8] S. Chowdhury and K. Sohraby, "Bandwidth Allocation Algorithms for Packet Video in ATM Networks," *Computer Networks and ISDN Systems,* 26, 1994, 1215–1223.

[9] K. Claffy, "Internet Traffic Characterization," Ph.D. Dissertation, University of California, San Diego, 1994.

[10] M. De Prycker, *Asynchronous Transfer Mode - Solution for Broadband ISDN,* Second edition, Ellis Horwood Pub., 1989.

[11] M. Decina and T. Toniatti, "On Bandwidth Allocation to Bursty Virtual Connections in ATM Networks," ICC 90, 1989.

[12] Z. Dziong et al, "Admission Control and Routing in ATM Networks," ITC Specialist Seminar, Adelaide, 1989.

[13] A. Eckberg, "B-ISDN/ATM Traffic and Congestion Control," *IEEE Network,* Sept. 1992.

[14] C. Fayet, A. Jacques and G. Pujolle, "High Speed Switching for ATM: the BSS," *Computer Networks and ISDN Systems,* 26, 1994, 1225–1234.

[15] P.E. Fleischer, R.C. Lau, M.E. Lukacs, "Digital Transport of HDTV on Optical Fiber," *IEEE Communications,* Aug. 1991, pp. 36–41.

[16] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W.H. Freeman Publishing, Inc., 1979.

[17] A. Gersht and K. Lee, "A Congestion Control Framework for ATM Networks," *IEEE J. on Selected Areas in Communications,* Sept. 1991.

[18] F. Guillemin and A. Dupuis, "A Basic Requirement for the Policing Function in ATM Networks," *Computer Networks and ISDN Systems,* 24(4), 15 May 1992, 311–320.

[19] S. Haykin, *Neural Networks: A Comprehensive Foundation,* Macmillan Publishing, Inc., 1994.

[20] A. Hunter, *SUGAL User Manual v2.0,* 1995.

[21] D.A. Junkins, "ATM Switch Simulator," Masters Thesis, University of Washington, April 1996.

[22] J.-Y. Le Boudec, "The Asynchronous Transfer Mode: A Tutorial," *Computer Networks and ISDN Systems,* 24(4), 15 May 1992, 279–310.

[23] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM,* Vol. 34, No. 4, April 1991.

[24] M.L. Liou, "Overview of the px64 kbit/s Video Coding Standard," *Communication of the ACM,* Vol. 34, No.4, Apr. 1991.

[25] M.L. Liou, "Visual Telephony as an ISDN Application," *IEEE Communications,* February 1990, pp. 30–38.

[26] S. Loeb, "Delivering Interactive Multimedia Documents over Networks," *IEEE Communications Magazine,* May 1992, 53.

[27] D.E. McDysan and D.L. Spohn, *ATM Theory and Application,* McGraw-Hill, Inc., 1994.

[28] S. Mehta, "Network Monitoring and Testing," *Communications Week,* May 22, 1995, S4–S8.

[29] A. Miller, "From Here to ATM," *IEEE Spectrum,* 31(6), June 1994, 20–24.

[30] E.P. Rathgeb, "Policing Mechanisms for ATM Networks, Modeling and Performance Comparison," *Proc. 7th ITC Seminar, Morristown, NJ,* Oct. 1990.

[31] W. Roberts, "Traffic Control in B-ISDN," *Computer Networks and ISDN System,* 25 1993, 1065–1064.

[32] J. Rosenberg *et al.* , "Multimedia Communications for Users," *IEEE Communications Magazine,* May 1992, 26.

[33] D. Ruiu, "Testing ATM Systems," *IEEE Spectrum,* 31(6), June 1994, 25–27.

[34] W. Stallings, *ISDN and Broadband ISDN with Frame Relay and ATM,* Third edition, Prentice Hall Pub., 1995.

[35] J. Sutherland and L. Litteral, "Residential Video Services," *IEEE Communication,* February 1992, 36 - 37.

[36] K. Wallace, "The JPEG Still Picture Compression Standard," *Communications of the ACM.* 34(4), Apr. 1991.

[37] H. Walther, *Ten Applications of Graph Theory,* D. Reidel Publishing Company, 1984, 31 - 68.

[38] J.L. Wang and L.T. Lee, "Two-Level Congestion Control Schemes for ATM Networks," *ACM SIGICE Bulletin,* 20(2), October 1994, 13–32.

# Vita

Sundararajan Vedantham received his Bachelor of Engineering degree in Electronics and Instrumentation from Annamalai University, India in 1986. After working for Hindustan Aeronautics Ltd. India, for an year and Oil and Natural Gas Commission of India for two more years, he joined Louisiana State University in Fall 1989 to pursue graduate studies. He received his Master of Science degree in Computer Engineering in 1991. Subsequently he joined the Department of Computer Science at L.S.U. to work towards the Doctor of Philosophy degree. He received his doctorate in 1997. He has been working as a network administrator for the College of Education, L.S.U. since 1993. His research interests include ATM traffic management, high speed networks and study of theoretical models of computers.

## DOCTORAL EXAMINATION AND DISSERTATION REPORT

**Candidate:** Sundararajan Vendantham

**Major Field:** Computer Science

**Title of Dissertation:** Traffic Management and Congestion Control in the ATM Network Model

**Approved:**

Major Professor and Chairman

Dean of the Graduate School

**EXAMINING COMMITTEE:**

**Date of Examination:**

October 9, 1997

# IMAGE EVALUATION TEST TARGET (QA-3)

150mm

6″

APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989