Traffic Shaping for Frame Relay over ATM Network

by

Miss Sujitra Anathep

Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Science
in Computer Science
Assumption University

May, 2001

Traffic Shaping for Frame Relay over ATM Network

by

Miss Sujitra Anathep

Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science Assumption University

May, 2001

The Faculty of Science and Technology

Thesis Approval

Thesis Title	Traffic Shaping for Frame Relay over ATM Network
By	Ms. Sujitra Anathep
Thesis Advisor	Asst.Prof.Dr. Chanintorn Jittawiriyanukoon
Academic Year	2/2000

The Department of Computer Science, Faculty of Science and Technology of Assumption University has approved this final report of the **twelve** credits course. **SC7000 Master Thesis**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Approval Committee:

(Asst.Prof.Dr. Chanintorn Jittawiriyanukoon)
Advisor

(Asst.Prof.Dr. Dobri Batovski)
Committee

(Dr. Pham Manh Lam)
Committee

(Asst.Prof.Dr. Surapong Auwatanamonkol)
Representative of Ministry of
University Affairs

Faculty Approval:

(Asst.Prof.Dr. Tang Van To)
Director

(Asst.Prof.Dr. Pratit Santiprabhob)
Dean

May / 2000

TABLE OF CONTENTS

ABSTRACT

To provide frame relay over ATM backbone, frames to cells conversion makes the cell generation process for FR VC's very busy. Thus, in order to achieve high bandwidth utilization, fair allocation of bandwidth resources and guaranteed QoS, traffic shaping after frames to cells conversion is suggested.

This research focuses on the issue of traffic shaping for frame relay over ATM platform. We evaluate the performance of shaping when various size of shaper is used. We also study the criteria for selecting a cell from shapers. And the conclusions are: 1) if the buffer size is large, the number of shaper does not affect the performance of traffic shaping. 2) the performance of shaping between using one large shaper and parallel small shapers with the same total size is somewhat not different. 3) if the buffer size is large enough, the performance of shaping is somewhat independent of scheduler and 4) also the traffic, if the traffic is not busy, the criteria of scheduler does not affect the performance of traffic shaping.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Motivation

Quality of Service (QoS) is an important issue for ATM networks. Since broadband network based on ATM are being designed for integration of various types of network services and end-user applications ranging from voice and interactive data to image, bulk data, and video. And these applications vary greatly in their bandwidth. The key problem is to allocate bandwidth in such a fashion so as to meet the performance objective of all traffic classes while maximizing the utilization of network resources, in particular, the transmission bandwidth. When a virtual circuit is established, both the transport layer and the ATM network layer must agree on a contract defining the service. But in a packet network that implements resource sharing, admission control, and scheduling schemes by themselves are not sufficient to provide guarantees. This is due to the fact that users may, inadvertently or otherwise, attempt to exceed the rates specified at the time of connection establishment. The QoS is usually negotiated for average rate, but bursts occur during transmission, hence resource allocation become unbalanced. We need traffic shaping at the entry into the network, and/or within the network. The paper entitled " Traffic Descriptor Mapping and Traffic Control for Frame Relay over ATM Network " written by Sudhir S. Dixit and Sharad Kumar [1]. Focuses on the issue of FR traffic descriptor mapping, the issue of traffic policing in the frame mode and the cell mode, and the implementation on the QoS when the same bandwidth resources are shared by the ATM and FR virtual connections. And one of major conclusions is that it becomes necessary to provide traffic shaping function in order to achieve high bandwidth utilization, fair allocation of bandwidth resources and guaranteed QoS.

Shaping can be implemented by using a simple buffer. A deterministic server serves the buffer/queue. The cell interarrival time is bounded by the service rate of the queue. Since shaping function can introduce some cell delay and possibly some cell loss. If buffer size is small, cell loss becomes high. And the advantages gained from shaping may not outweigh its disadvantages. However, if we use a large buffer, it will be costly. In Sudhir S. Dixit and Sharad Kumar's paper, they did not include an appropriate size of buffer. Thus, we will propose how to find an appropriate buffer size to maintain the advantages of shaping.

1.2 Statement of The Problem

This research deals with traffic shaping for frame relay over ATM. Thus, we have 2 sources, source one subscribes for service through a frame relay interface and source two subscribes for service through an ATM interface. The two sources belong to the same traffic class and can be multiplexed in the same capacity. Each source has its own shapers. We have 3 criteria to shape traffic:

1) **Sequential scheduler:** a job is selected from the shaper in order. First time a job is selected from the first shaper, second time a job is select from the second shaper and so on.

2) **Queue length based scheduler:** a job is selected from the shaper based on queue length. A node with longer queue has higher priority.

3) **Randomly scheduler:** a job is selected randomly.

And for all criteria, each shaper is serving as first come first serve (FCFS).

Since this study also deals with buffer size of shaper, thus we adjust the size of each shaper and also adjust the number of shapers for each source.

1.3 Objectives of The Problem

The main objectives of this research are:

- To study the performance of each scheduler which is mentioned in section 1.2.
- To study the affect of number of shapers to traffic shaping.
- To investigate how buffer size of shaper affects the performance of traffic shaping in frame relay over ATM network.

CHAPTER 2 LITERATURE REVIEW

In chapter 1, we mention about a research entitled "Traffic Descriptor Mapping and Traffic Control for Frame Relay Over ATM Network" written by Sudhir S. Dixit and Sharad Kumar [1]. Thus, this related research should be reviewed to make clear why we do our research.

This paper identified and proposed solutions to some of the traffic parameter mapping and policing problems when frame relay service is offered over an ATM platform via a frame relay interface. The paper discussed some of the advantages of offering frame relay service over an ATM network and compared the simulation results when the traffic control procedures were implemented in the frame mode and the cell mode. The simulation results suggested that the performance is somewhat independent of the mode in which traffic control is implemented. For frame relay over ATM, the implications of the frame-to-cell conversion process on bandwidth allocation were also studied. It was shown that the allocation of bandwidth in the multiplexer and the switching fabric is greatly dependent on the service interface. Further more, they also studied about traffic shaping by running the following simulation. A channel capacity of 5 Mb/s was assumed, and buffer sizes of 150 were assumed both at the mutiplexer and at the shapers for the FR VC's. the service rates of the shaping queues were also set at 5 Mb/s. the conversion of frames to cells occurred at an infinite internal rate. And the results clearly show that traffic shaping on the converted cells becomes a necessity in order to achieve high bandwidth utilization, fair allocation of bandwidth resources, and guaranteed QoS.

CHAPTER 3 KNOWLEDGE BACKGROUND

3.1 ATM

Asynchronous Transfer Mode (ATM) has been accepted as the basis for the transfer of information within broadband network. ATM is fundamental for B-ISDN because it provides higher bandwidth, low-delay, packet-like switching, and multiplex for various service types [7][8]. The basic idea behind ATM is to transmit all information in small, fixed-size packets called cells. The ATM cell structure consists of 53 bytes: 48 bytes for the information field and 5 bytes for the preceding header. The header field contains information about the virtual channel (VCI: Virtual Channel Identifier) and virtual path (VPI: Virtual Path Identifier) in use, payload type (PT) and cell loss priority (CLP).

Broadband ISDN using ATM has its own reference model, different from the OSI model and also different from the TCP/IP model. This model is shown in Fig.3-1. It consists of three layers, the physical, ATM, and ATM adaptation layers, plus whatever the users want to put on top of that.

The physical layer deals with the physical medium: voltages, bit timing and various other issues. ATM does not prescribe a particular set of rules, but instead says that ATM cells may be sent on a wire or fiber by themselves, but they may also be packaged inside the payload of other carrier systems. In other words, ATM has been designed to be independent of the transmission medium.

The ATM layer deals with cells and cell transport. It defines layout of a cell and tells what the header fields mean. It also deals with establishment and release of virtual circuits. Congestion control is also located here.

The ATM Adaptation layer (AAL) deals with inserting payload data into the 48-byte information field of the ATM cell. The AAL is what gives ATM the flexibility to carry entirely different types of services within the same format. It is important to understand that the AAL is not a network process but instead is performed by the network terminating equipment. Thus the network's task is only to route the cell from one point to another, depending on its header information.

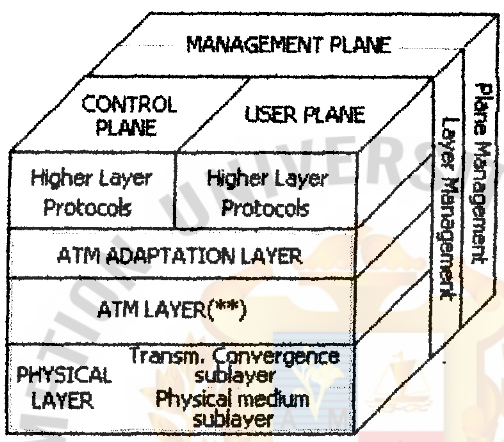

Figure 3-1 The B-ISDN ATM reference model.

Note that up to four bytes might be used by the adaptation process itself with some AAL types, leaving 44 bytes for payload information.

Several adaptation layers have already been standardized. These are:

- **Type 1:** Constant Bit Rate (CBR) services. AAL1 handles traffic where there is a strong timing relation between the source and the destination. Examples include PCM-encoded voice traffic, constant bit rate video and the emulation of public network circuits (e.g. the transport for E1 links).
- **Type 2:** Variable Bit Rate (VBR) timing-sensitive services. AAL2 handles traffic where a strong timing relation between the source and the destination is required, but the bit rate may vary. Examples include variable bit rate voice and compressed, for instant MPEG-coded, video.

- **Type 3/4:** Connection-oriented and connection-less VBR data transfer. AAL3/4 is a fairly complex layer that can handle VBR (i.e. bursty) data both with and without pre-establishing an ATM link. Examples for the connection-oriented type include large file transfer like CAD files or data back-up. The connectionless type is intended for short, higher bursty transfers as might be generated by LANs.
- **Type 5:** Simple and Efficient Adaptation Layer (SEAL). AAL5 may be looked upon as a simplified version of AAL3/4 that is designed to meet the requirements of local, high-speed LAN implementations. AAL5 is intended for connectionless or connection-oriented VBR services.

The ATM model is defined as being three-dimensional, as shown in Fig.3-1. The user plane deals with data transport, flow control, error correction, and other user functions. In contrast, the control plane is concerned with connection management. The layer and plane management functions relate to resource management and interlayer coordination.

ATM network offers several advantages because of their underlying principle:

- Effective usage of network capability through bandwidth on demand and shared bandwidth between parallel applications.
- Low transfer delay and support of both non-real-time and real-time applications through the provision of large peak bandwidth of up to 155 Mbit/s to the telecommunications end-user.
- Support of multimedia applications and mixed traffic through VPs and VCs.
- An easy to manage network infrastructure.

ATM technology is the flexible and powerful common platform for local area networks (LAN) and wide area networks (WAN) to increase productivity, to reduce costs and to implement new applications and services.

3.2 Frame Relay

Frame Relay is a high-speed communications technology. It is a way of sending information over a wide area network (WAN) that divides the information into frames or packets. Each frame has as address that the network uses to determine the destination of the frame.

Frame Relay technology is based on the concept of using virtual circuits (VCs). VCs are two-way, software-defined data paths between two ports that act as private line replacements in the network. While today there are two types of frame relay connections, switched virtual circuits (SVCs) and permanent virtual circuits (PVCs), PVCs were the original service offering. As a result, PVCs were more commonly used, but SVCs products and services are growing in popularity.

Using PVCs

PVCs are set up by a network operator-whether a private network or a service provider- via a network management system. PVCs are initially defined as a connection between two sites or endpoints. New PVCs may be added when there is a demand for new sites, additional bandwidth, alternative routing, or when new applications require existing port to talk to one another.

PVCs are fixed paths, not available on demand or not a call-by-call basis. Although the actual path taken through the network may vary from time to time, such as when automatic rerouting takes place, the beginning and end of their circuit will not change. In this way, the PVC is like a dedicated point-to-point circuit.

PVCs are popular because they provide a cost-effective alternative to leased lines. Provisioning PVCs requires thorough planning, knowledge of traffic patterns, and bandwidth utilization. There are fixed lead times for installation which limit the flexibility of adding service when required for short usage periods.

Using SVCs

Switched virtual circuits are available on a call-by-call basis. Establishing a call by using the SVC signaling protocol is comparable to normal telephone use. Users specify a destination address similar to a phone number.

Implementing SVCs in the network is more complex than using PVCs, but is transparent to end-users. First, the network must dynamically establish connection based on requests by many users (as opposed to PVCs where a central network operator configures the network). The network must quickly establish the connection and allocate bandwidth based on the user's requests. Finally, the network must track the calls and bill according to the amount of service provided.

The basic flow of data in a frame relay network can best be described in a series of key points:

- Data is sent through a frame relay network using a data link connection identifier (DLCI), which specifies the frame's destination.
- If the network has a problem handling a frame due to line errors or congestion, it simply discards the frame.
- The frame relay network does no error correction; instead, it relies on the higher layer protocols in the intelligent user devices to recover by retransmitting the lost frames.

- Error recovery by the higher layer protocols, although automatic and reliable, is costly in terms of delay, processing and bandwidth; thus, it is imperative that the network minimizes the occurrence of discards.
- Frame relay requires lines with low error rates to achieve good performance.
- On clean line, congestion is by far the most frequent cause of discards; thus, the network's ability to avoid or react to congestion is extremely important in determining network performance.

Frame relay provides a number of benefits over alternative technologies:

- Lower cost of ownership.
  - Well-establish and widely adopted standards that allow open architecture and plug and play service implementation.
  - Low overhead combined with high reliability.
  - Network scalability, flexibility and disaster recovery.
  - Interworking with other new services and applications, such as ATM.

Frame relay offers users the ability to improve performance (response time) and reduce transmission costs dramatically for a number of important network applications. In order to be effective, frame relay requires that two conditions be met:

1. the end devices must be running an intelligent higher-layer protocol
2. the transmission lines must be virtually error-free

3.3 Frame Relay over ATM

Both frame relay and ATM are connection-oriented services that are using virtual channels to transport information. Frame relay can be introduced into multiple service networks where ATM provides the high-speed backbone. Interworking between frame relay and ATM is done via the ATM switch.

There are two ways of providing frame relay services over ATM:

Frame relay / ATM Network Interworking

Frame relay / ATM Network Interworking allows Frame relay end-user or networking devices such as FRADs or routers to communicate with each other via an ATM network employed as the backbone. The frame relay devices interact as if they are using frame relay for the entire connection without knowing that an ATM is in the middle.

Traffic enters the ATM network over a frame relay user network interface. The ATM network maps frame relay frames into ATM cells. The cells are transported through the ATM network to the destination node where the cells are reassembled into frames and handed over the users via the frame relay user network interface. An interworking function (IWF) provides all mapping and encapsulation function that are required.

Figure 3-2 Frame Relay / ATM Network Interworking

Frame Relay / ATM Service Interworking

Frame Relay / ATM Service Interworking enables communication between an ATM and frame relay network or end user devices. With service interworking, the traffic enters the network via the frame relay user network interface and exits over an ATM user network interface or vice versa. Transparent interoperation between frame relay and ATM users is provided. The interworking function (IWF) translates frame relay and ATM specific parameters that are used for traffic management and for quality of service. The way of encapsulation of higher level protocols (e.g. IP) is being converted by IWF.

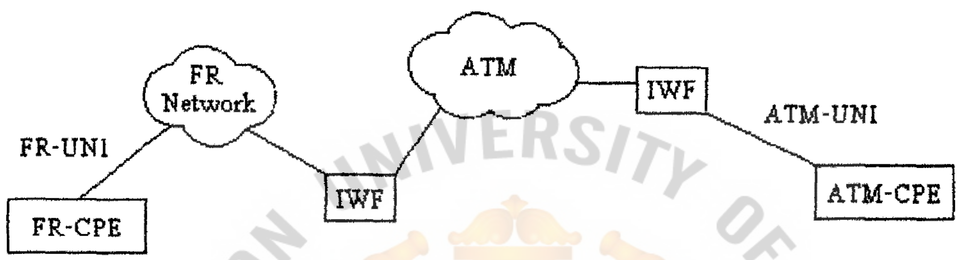

Figure 3-3 Frame Relay / ATM Service Interworking

An important advantage of Frame Relay / ATM Interworking is that it provides solutions to support communications between frame relay and ATM environments without modifications to end-user devices. Since this advantage, users are interested in interworking frame relay and ATM networks to protect their capital investment in existing frame relay networks and to support planned migrations from frame relay to ATM.

3.4 Leaky Bucket Algorithm

Traffic and congestion control is an important thing. Most networks implement traffic control and congestion control mechanisms to preserve network performance objectives [9][10]. There are many functions provide a framework for managing and controlling traffic and congestion in ATM networks [2] and traffic shaping is one of them.

Traffic shaping is necessary for improving services in ATM networks such as VBR video signal transmission [3] and as we said earlier about high bandwidth utilization, fair allocation of bandwidth resources and guaranteed QoS [1].

Traffic shaping is a function that alters the traffic characteristics of a cell stream on a VCC or a VPC to achieve a desired modification of those traffic characteristics [6]. There are a number of algorithms to provide traffic shaping function but leaky bucket algorithm is the simplest one [4]. Imagine a bucket with a hole in it (Fig. 3-4). The bucket "leaks" at a steady rate, no matter what rate water enters the bucket. If the bucket is full, some water may be lost. The same idea can be applied to packets. Each host is connected to the network by an interface containing a finite interface queue. If a packet arrives at the queue when it is full, the packet is discarded.

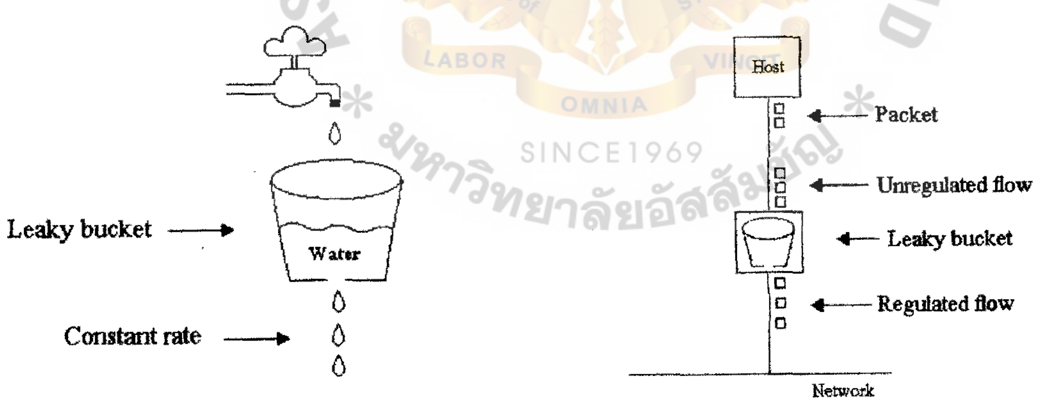

Figure 3-4 Leaky Bucket Algorithm

CHAPTER 4 SIMULATION BACKGROUND

4.1 Discrete System Simulation

Systems may be classified by the way in which their variables (which are chose to represent the system state) change over time. Discrete system is a system that its variables change in a discontinuous manner over time. Available simulation tools for discrete systems are now classified into three categories:

1) **General purpose event-oriented languages:** These languages provide the simulation engine and other standard modules. The user has to provide the individual event-processing routines, so he or she must have some programming skill. Because of the user's control of the contents of individual event routines, these languages are generally more flexible than process-oriented languages in representing complex and specialized processes.

2) **General purpose process-oriented languages:** these languages attempt to provide a set of modules, each representing a commonly occurring process (usually in queuing systems). Since not all possible processes may be predicted and incorporated in a simulation language, some process-oriented languages may also provide for event-oriented modeling. The advantage of event orientation is the flexibility that it provides. The combined event/process-oriented language allows the user to benefit from the advantage of each type of orientation. When modeling a system using the combined language, the process orientation of the language is used to the maximum extent possible. If there are certain sections of the system that may not be represented by the available modules, specialized routines may be written and linked to the model using the event orientation capability of the language.

3) Special purpose simulation environments: these are a special class of process-oriented software that are dedicated to certain application domain, such as manufacturing, communication networks, traffic flow, and scheduling. Since they generally do not require programming, they are called "environments" rather than languages.

4.2 Common Processes in Discrete Systems

Processes-oriented simulation modeling tools attempt to provide prewritten and precompiled program modules that represent commonly occurring phenomena (processes) that take place in various realistic scenarios, thereby relieving the model builders of the tedious task of frequently reconstructing these program segment for use in their various simulation projects. The task of simulation programming then reduces to appropriately selecting and logically linking these ready-made modules, and specifying related parametric values.

Not all general purpose process-oriented simulation languages present the same set of modules as a unique repertoire of common processes. Nevertheless, there is a relatively large overlap in the offerings of all of these languages that includes a number of nearly similar processes. The basis for this similarity is the movement/accumulation concept of the physical commodity, which is universal applies to all application domains in which entity motions are present.

Before going to our discussion of common processes in discrete systems, let us list some brief definitions of the important terms:

System. A section of reality in the form of a set of components connected such that they can perform a function not performable by the individual components.

Parameters. Quantities in the system that do not change unless the analyst commands them to.

Variables. Quantities in the system that are determined by functional relationships and that change over time in dynamic systems.

System state. A snapshot of the system at any point in time characterized by the values of some selected variables in the system.

Events. Changes in the state of the system.

Entities. Objects in the dynamic system whose motion within the system may result in the occurrence of events.

Attributes. Characteristics and properties that describe entities.

Relationships. Expressions of dependency between elements such as variables, parameters, and attributes of a system.

Activities. Time-consuming elements of a system whose starting and ending coincide with event occurrences.

Resources. Limited commodities that are used, consumed, or replenished by the entities.

Control. A mechanism that directs a dynamic system toward a set goal.

Transient state. Atypical conditions that impost time-dependent, radical *perturbations on the system state.*

Steady state. Typical conditions in which changes in the system state within a fixed range and are independent of time.

The organization of the following sections will be based on the entity (commodity) movement pattern through various processes and selection of alternate routes, and on the conditions for entity accumulation in queues. A commonly used form of representation of entity flow by process-oriented languages is a network representation. A network is a collection of nodes that are interconnected by branches. In simulation models, a node may represent a common process that some entities go through.

4.2.1 Entity Movement

Dynamism in the discrete systems is caused by the movement of entities which results in the occurrence of events that change the system state over time.

4.2.1.1 Entity Creation

To create entities as inputs to the system common process acting, as an entity source should be considered. The module representing this process is called source, create, generate, and so on by different simulation languages.

4.2.1.2 Entity Termination

In most system entities that enter the system also leave it. The module that represents this entity departure or disappearance is called terminate, sink, depart, out, and so on by various simulation modeling languages. Although the entities that leave the system are of no further concern to the analyst, the entity termination process is essential for clearing these entities and their attribute information from the computer memory. Otherwise a memory overflow may be experienced during the course of the simulation. A desirable property of the module representing the termination process is the ability to specify the number of entity terminations required to end the simulation process.

The choice of controlling the entity creations in the source module by number or by time, and the number of entity terminations needed at the termination module to end the simulation, provides the analyst with the option of ending the simulation when the system is in either the empty or nonempty state. If the number of entities specified in the source module is greater than that specified in the termination module, then at the end of the simulation there will be some entities left in the system.

4.2.1.3 Entity Traversal

Entities traverse between system components through branches that connect system components. The entity motion through branches may take time, in which case the entity will be displayed in arriving at its destination point.

A desirable property of the entity traversal module is the ability to specify the *delay time* as a constant, a random variable, a user variable, or an expression combining several of those elements. Also, since various entities may traverse through the same branch at different speeds (which may be represented by one of the entity attributes), the traversal module should allow for making the delay time a function of entity attribute.

Another desirable capacity of the module is the ability to *signal the end of the traversal time* with the occurrence of an event. This should allow for those types of entity traversals whose completion may not be schedule in advance.

4.2.1.4 Branch Selection

In many practical situations entities have alternative routes that they may take. The module should allow for the *probabilistic selection* of branches. That is, it should be possible to assign each alternative branch a probability of selection by the arriving entity.

The module should allow for the *conditional selection* of alternative branches; that is, it should be possible to assign a condition to each alternative branch. A branch is then taken by the arriving entity if the associated condition is met. The condition may be based on the values of the global variables, the entity attribute values, or a complex expression.

Since *selection from different queues* (waiting lines) is a commonly occurring process, it is desirable to have provision to some specialized conditional branching on the basis of the status of the alternative queues that an arriving entity may join. Another common process is the selection of which queue to serve if the server has the option of serving more than one queue. Criteria such as selecting from the longest queue, cyclic selection, or random selection are typical.

Another common process is the *selection of a server* from among multiple servers with varying characteristics. Possible criteria here may be the selection of the server with the longest idle time, cyclic selection, or random selection.

4.2.2 Entity Use of Resources

There are 2 types of resources, many simulation languages prefer to handle them differently. We will refer to the first type of resource with the special name of *facility* and the second type *resource*. Following are the descriptions of common properties of each resource type:

Facilities usually represent servers, operations, workstations, or machines with which certain *service times* are associated. The duration of service could have properties similar to that of the entity traversal time. Facilities may have several homogeneous servers working in parallel. A common property of a facility is its possible breakdown or idle time allowance, which may be random or scheduled. Also as mentioned before, certain selection criteria may apply when selecting from among several nonhomogeneous facilities or when selecting an entity to serve from multiple parallel queues before a facility that just became idle. Finally, entities may concurrently need other types of resources while receiving service at the facility (such as tools or machinists for a machining operation)

Resources generally have a *base stock* that initially contains a known number of resource units. Entities can request a certain number of resource units. There may be a delay involved in receiving the resource. The resources seized by the entity may be kept while the entity goes through various processes. They may be returned to base stock after use, in which case there may be a return time involved. Entities may have different priorities in accessing the resources. At times a higher priority entity may preempt a lower priority entity that is in possession of the needed resource. The time during which an entity uses a resource generally need not be explicitly identified since the entities possessing the resources can go through several time-consuming activities of the system.

4.2.3 Entity Accumulation

Throughout the course of their movements in the system, entities may wait at certain accumulation points for a variety of reasons. In discrete systems these accumulation points are generally called *queues*.

The most common situation for queue formation occurs *when entities require the use of facilities*. Facilities may be occupied or temporarily unavailable for reasons other than that they are busy processing entities. The arriving entities that find the needed facility unavailable or occupied by other entities have to wait in a queue. There may be multiple (parallel) queues before a facility.

Queue formation may take place when arriving entities *request a certain resource* (other than facility). If the resource base stock becomes empty, the entities have to queue up until the requested resource becomes available. Each entity may demand one or more resource units. Therefore, an entity that has only a portion of its needed resource available must wait in the queue until its desired number of resources is provided.

Entity grouping is another condition that necessitates queue formation. When a certain number of entities is needed to form a group (in either batched or combined forms) all entities that arrive earlier than the last one (which makes the required number) have to wait in a queue before departing in the form of a single, grouped entity. Also, when carriers are used to move a group of entities, the arriving entities need to queue up and await the carrier arrival.

In certain situations entities have to *await the arrival of other entities* with the same value of a specified attribute at some points in the system before they can proceed. Once the *matching* entities are all available at their assigned queue, all matching entities leave simultaneously, each taking its own route.

Finally, entities at certain points in the system may need to *await permission* to proceed. The permission may be issued by other entities at remote locations in the system.

4.2.4 Auxiliary Operations

Thus far, our discussion of common processes in discrete systems has considered those occurrences that correspond to physical phenomena readily observable in real-world scenarios. For each of the processes described, one may construct or use a modeling module that represents the process. When one constructs a model of a system there are, however, certain auxiliary operations that may be needed, which may or may not correspond to any physical phenomenon. Following is a description of some of the most common auxiliary operations.

4.2.4.1 Assignment Operations

In many situations a modeler may need to define some global variables or entity attributes and assign, or occasionally change, the values of these variables and attributes. The values that are affected may be changed by constant or random amounts or be derived from complex expressions. It is often desirable to use if-then-else conditions at a single assignment module to avoid excessive conditional branches. This capability would be useful in situations when an entity must be directed to one of several alternative assignment points on the basis of a given set of conditions. Generally, the more capabilities built into the assignment module of a proccess-oriented simulation tool, the more flexible the tool becomes for modeling a variety of situations.

4.2.4.2 File Operations

In certain cases needs arise for transferring, deleting, or copying entities already in queues. We refer to those transactions as *file operations*. The choice of the term *file* is due to the fact that queues are complex data structures in which entities with various attributes are filed in a certain order for future processing. The indicator for selection of entities in files is usually the entity rank in the file (first, last, or *n*th). A specified entity attribute having a specific value is another common indicator for selection of entities from files.

4.2.4.3 Handling and Utilization of Variables

There are 2 kinds of global variables that are frequently used in systems simulation studies. These are *user variables* and *system variables*. User variables, as implied by name, are explicitly defined by the user for specific purposes. System variables are those global variables that are automatically created by the simulation tool, and their number and type depend on the specific model used. System variables are frequently used when specifying conditions, durations, and various forms of expressions, and at various assignment stages for setting user variables and entity attributes. A simulation tool should provide access to these variables when requested by the user.

4.2.4.4 System Initialization

User variables, resource levels, status of gates, and the possible number of existing entities in queues and their corresponding attribute values should be identified at the beginning of a simulation. A simulation tool should allow for initialization of these quantities prior to the actual model execution.

4.2.4.5 Statistics Specification

Another auxiliary operation in a simulation model is the specification of the desired statistics. Certain statistics that are commonly needed by users deal with information concerning the length of queues, the entity waiting time in queues, utilization of facilities, and utilization of resources. Because of the high frequency of user needs for these statistics, it is desirable for a simulation tool to automatically generate them. There are, however, other kinds of case-specific statistics that should be left for user to specify. These are generally one of the following types:

- Statistics on the arrival times of entities to a point in the system
- Statistics on the number of entities passing through a certain point in the system (entity count)
- Statistics on traversal time of entities between two points in the system, the most common type being the system time, or the total time spent in the system (from creation to termination)
- Statistics on the time between the passage of successive entities from a certain point in the system
- Statistics on global variables

4.2.4.6 Tracing Entities in The Model

Simulation models are usually hard to debug for error correction, model verification, and some basic aspects of model validation. Consequently, to help facilitate the debugging process a tracing capability that shows the position of some specified entities at various stages of model execution is usually provided by simulation languages. A trace capability that prints the name of the position of the specified entities at various event times, or preferably an animation capability that shows the motion of entities in the model network or flow diagram itself, is an essential feature of a good simulation tool.

4.2.4.7 Scene Animation

Scene animation is a capability that can enhance the presentation of a simulation model. A scene animation generally shows a background with some fixed components and some moving icons representing entities. Animation usually shows a short period of system operation, so it is not a proper means for performance analysis of the simulation system. Many events usually take place in a simulation run of a typical length that may not be entirely captured in a few animation frames. In addition, creation of scene animation usually takes a considerable effort, so scene animation is rarely used for debugging and trace purposes. Because of the difficulty involved in building scene animation and the limitation of its applications, most simulation studies do not use scene animation.

The desirable properties of a scene animation package include ease of use (preferably without programming), high-resolution graphics support, a library of animation icons, high-speed operation, smooth motion of moving images, and true three-dimensional support.

4.3 EZSIM - A General Purpose Discrete System Simulation Tool

EZSIM is a general purpose process-oriented simulation modeling tool for discrete systems involving entity flow. It was developed by Behrokh Khoshnevis from University of Southern California. The software design allows for maximum modeling flexibility within the process-orientation frame. Models in EZSIM are represented in network form. Each node represents a process and branches show the entity path from one node to another.

4.3.1 Procedure for Using EZSIM

Figure 4-6 shows the flowchart of the procedure for using EZSIM. The following stages are involved in a complete EZSIM.

- **Model network construction:** the network construction component of EZSIM submits a global representation of the system structure to a rule-based procedure that interactively asks for complementary information needed to progressively construct a complete system representation and form an internally complete model.

- **Nodal parameter specification:** generally, there are some parameters or other declarations that need to be specified for each node of the network. Fig. 4-1 shows nodal parameter specification screen.

Figure 4-1 Nodal parameter specification screen

- **Desired statistics specification:** after completing the nodal parameter specification stage, the statistics screen is displayed. Here we may specify the desired statistics by any identifier name that is to appear in the simulation output. Fig. 4-2 shows the statistics screen.
- **Execution in batch or in real-time with animation:** the model may be executed in either batch mode or animation mode. If the batch mode is

selected, the desired length of simulation in time units must be identified. The simulation process then starts and the current simulation time, as it progresses, is displayed. If the selected execution is animation. It may be observed in either the continuous mode or the step mode. Then the control returns to the model network and the motion of entities through the branches and nodes of the network model is displayed. Fig. 4-3 and Fig. 4-4 show the execution screen for batch mode and animation mode.

Figure 4-2 The statistics screen.

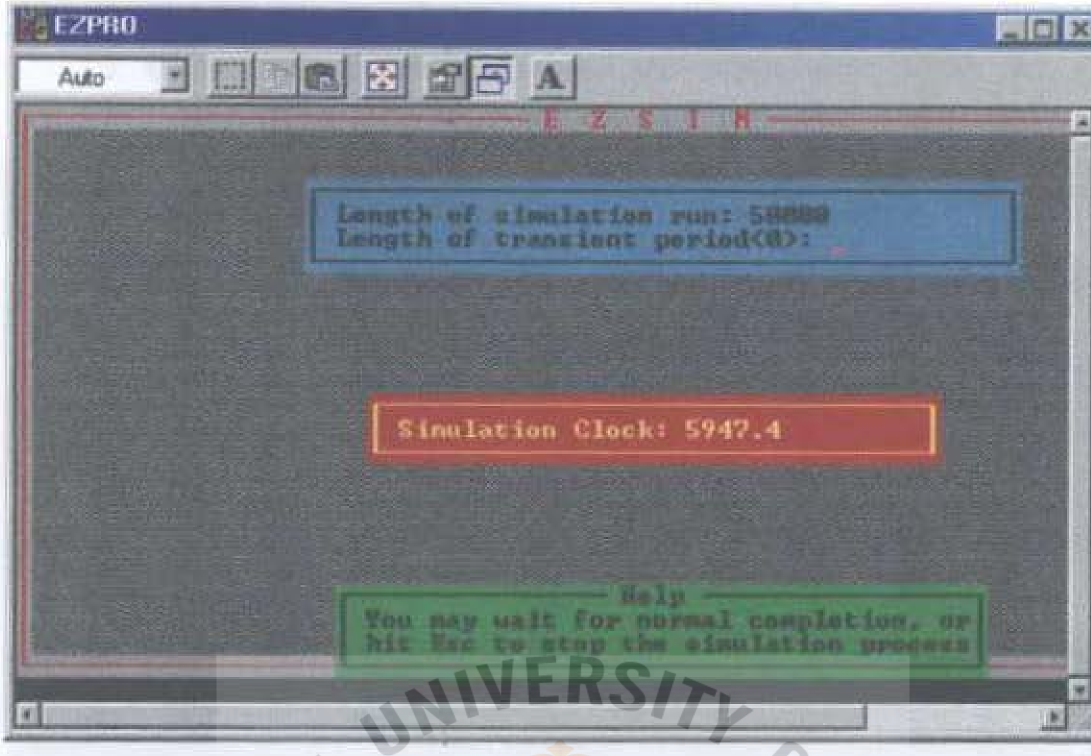

Figure 4-3 The execution screen for batch mode

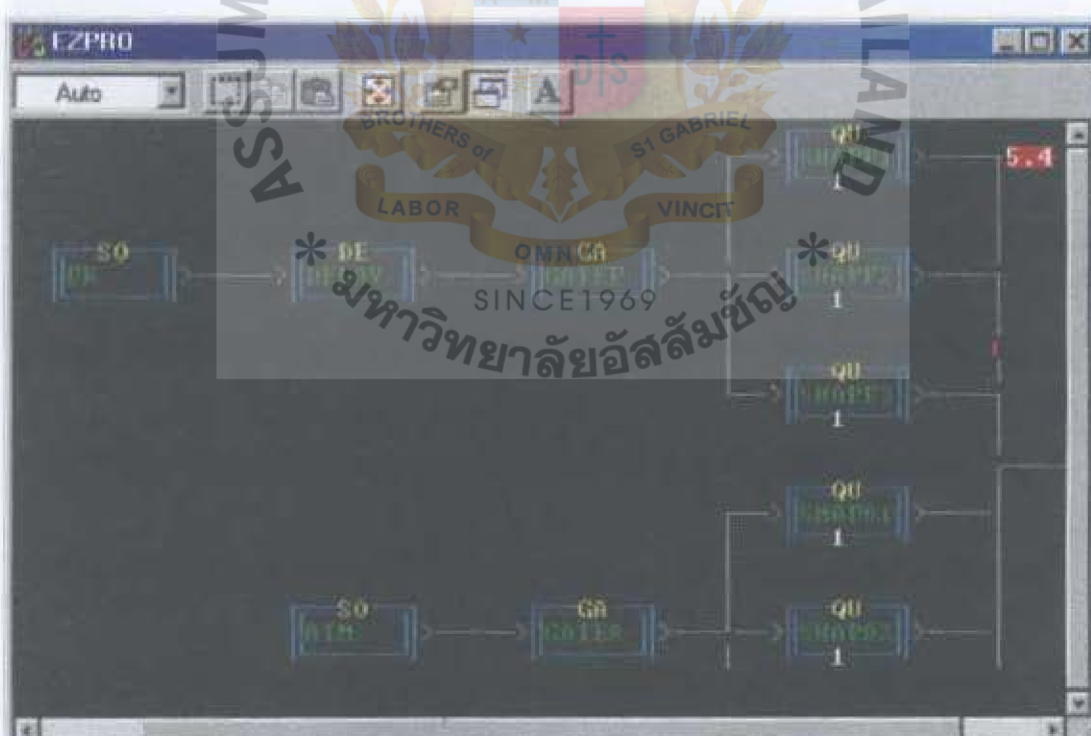

Figure 4-4 The execution screen for animation mode

- **Output observation on screen:** the output menu allows for the observation of the selected statistics and graphs on the screen, during or at the end of either the batch or the animation execution mode.
- **Model and output disk file and/or hardcopy generation:** the entire simulation output and the network model with its related information may be saved on the disk or sent to the printer by means of the related selection output in the Output menu.

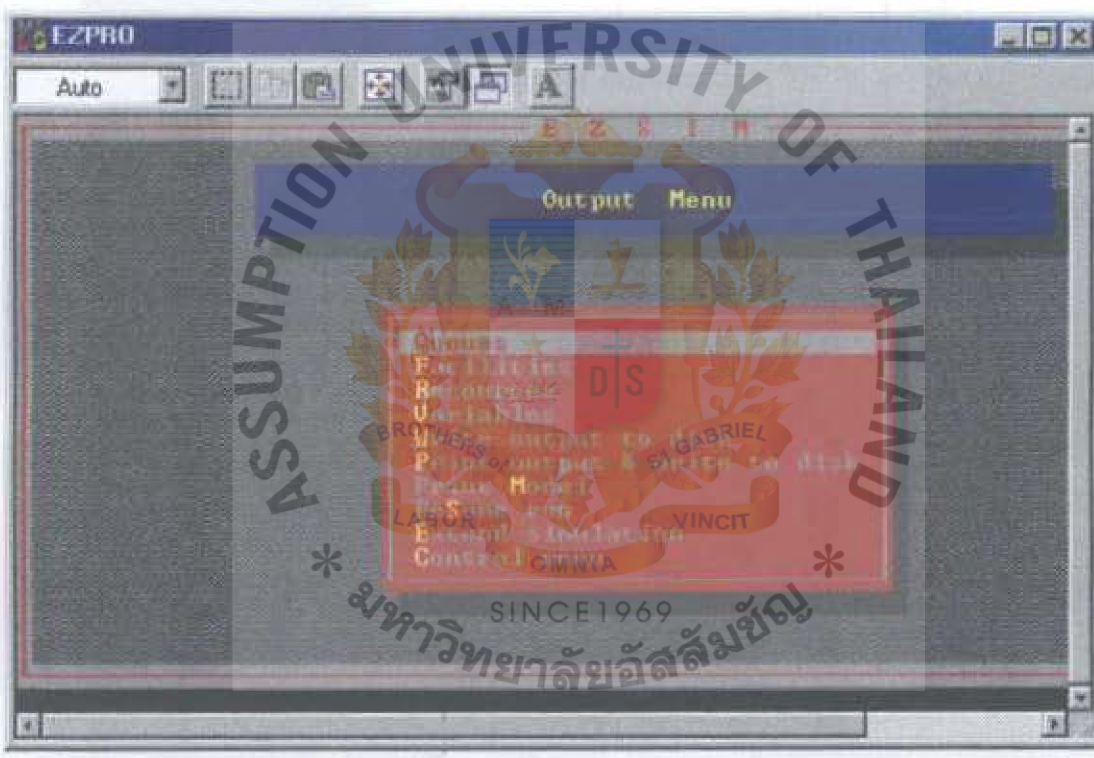

Figure 4-5 The output menu

- **Possible model modification and reexecution:** one of the major capacities of EZSIM is its ease of modifying the model network and related modeling information. Modifications may be made at any point in the model building process or after its completion. Nodes may be added or deleted and related parametric or declarative information may be modified as desired. And the execution may be resumed after the output is observed.

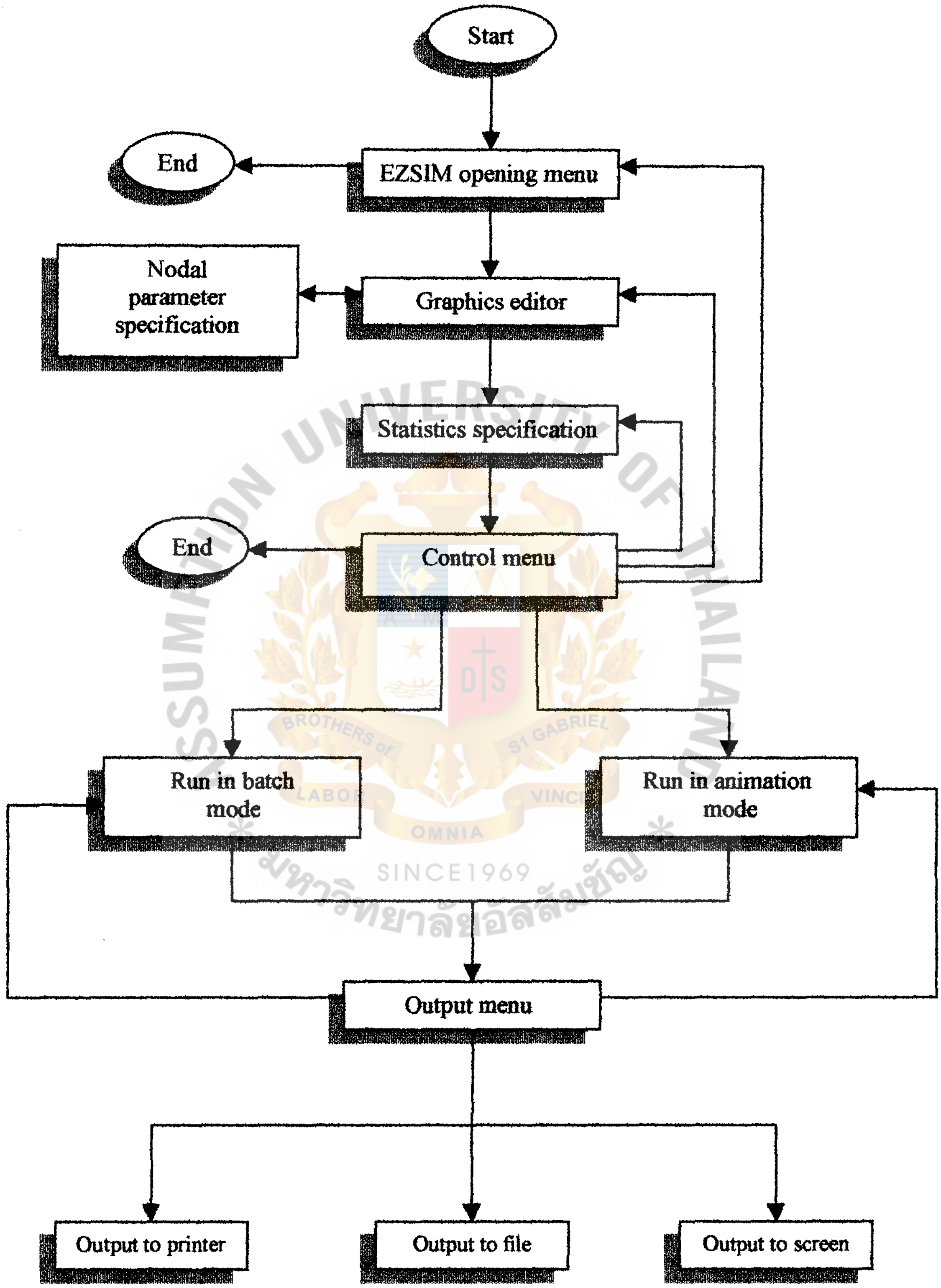

Figure 4-6 Flowchart of the procedure for using the EZSIM.

4.3.2 Simulation Output

It should be mentioned that EZSIM automatically collects and reports some frequently needed statistics. These include the statistics on all types of queues, facilities, and resources specified in the model. EZSIM also allows the user to choose the desired statistics from the following list of options:

- Arrival time of first entity at a node
- All times of entity arrivals at a node
- Traversal time between two nodes
- Time between arrivals
- Entity count at a node
- Any user-defined variable
- An expression

The statistics in the output are identified under the associated node or resource names given at the modeling stage.

4.3.3 EZSIM Modeling Objects and Capabilities

In this section, each EZSIM nodes that will be used in our model and related capacities are discussed.

- **Source Node:** The source node creates entities. It is possible to control the number of creations at the source node. And we can also specify various time elements such as time of first entity creation, time between creations.
- **Delay Node:** The delay node is used for creating a delay that corresponds to the traversal time of the entity from one node to another.
- **Gate-Q Node:** The gate-q node contains a buffer and a gate behind the buffer. When the gate is open, the entities in the buffer can leave the node;

otherwise, they have to wait in the buffer until the gate is open. Gates may be opened and closed by switch nodes that may be activated by some remote entities.

- **Switch Node:** The switch node can be used to open or close the gates from any remote point in the network. It allows for specification of the gates that are opened or closed every time an entity passes through the switch node.
- **Facility Node:** The facility node acts as a server. Entities remain in the node for the duration of their service. When the node is occupied, the arriving entities have to wait until the node is free. Facility node must be preceded by a queue node.
- **Queue Node:** The queue node represents buffers before facility nodes and is always succeeded by a facility node. A queue may have capacity limitation, in which case the arriving entities may balk to a specified node, block the facility behind (if any), or terminate at the node.
- **Terminate Node:** The terminate node ends the path of entities. When an entity enters a terminate node it is considered to be out of the network, and the computer memory allocated to its attributes id freed. Each terminate node can end the entire simulation if the entity count specified in it is reached.

CHAPTER 5 EXPERIMENTS AND DISCUSSION

5.1 System Model

Since we consider about traffic shaping for frame relay over ATM network. Thus, we have 2 identical sources which have the same QoS requirement (Fig. 5-1). Source 1 subscribes for service through a frame relay interface and source 2 subscribes for service through an ATM interface. These 2 sources alternate between burst and silent state (2-state Markov process). Burst and silent duration for both sources are assumed to be exponentially distributed with the same average values. Since these 2 sources behave identically, thus they belong to the same traffic class and can be multiplexed in the same subcapacity.

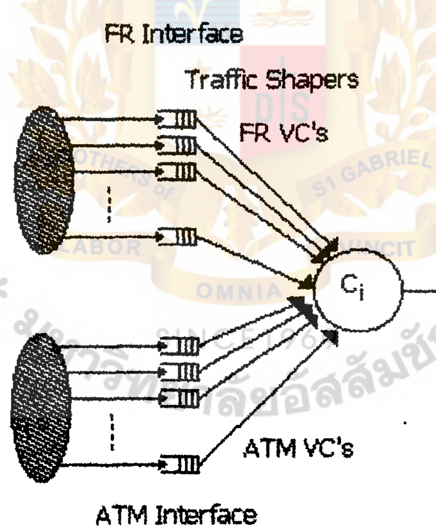

Figure 5-1 System Model

5.2 Simulation Model Construction

The first stage, we have to create a model for our network. Fig. 5-2 shows an example of model that is used in this research.

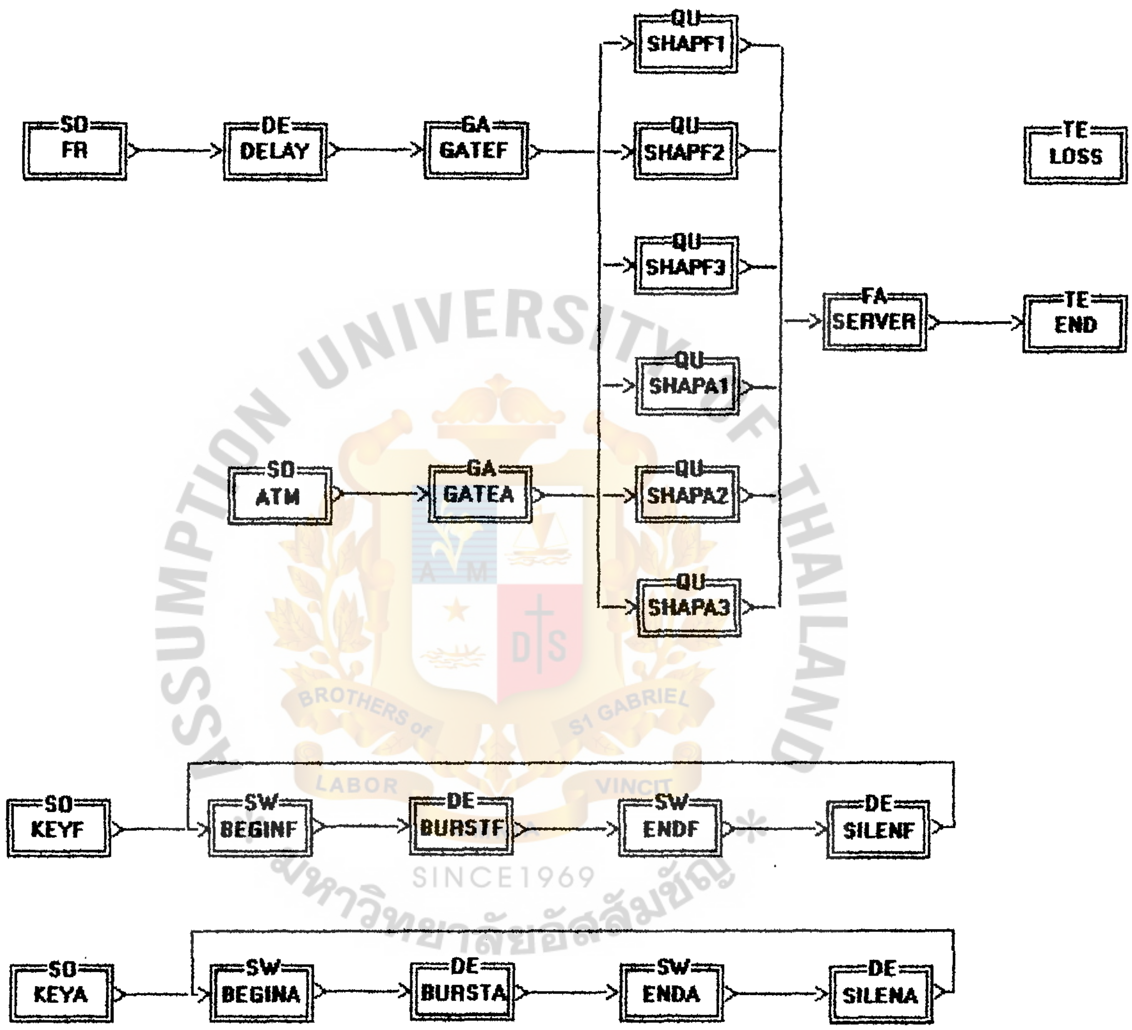

Figure 5-2 An example of simulation model

We study about traffic shaping for frame relay over ATM network. Thus, in the model, we have 2 source nodes. Source 1 generates traffic for frame relay and source 2 generates traffic for ATM. Since frame relay frames have to convert into ATM cells, a delay node is added after the source node that generates traffic for frame relay to increase time for conversion. We use queue node as a shaper. The number of shapers can be adjusted by reducing or increasing queue node in the model. Buffer size can be adjusted by changing queue node parameter. The model has a facility node and 2 terminate nodes. The facility node represents as a channel capacity. The 2 terminate nodes end the path of traffic cells. If a cell arrives at the queue when it is full, the cell will go to the terminate node named LOSS. All cells which ends at each terminate node are counted for calculating cell loss probability. We can also set burst duration and silent duration by using gate nodes. These gates are opened and closed by switch nodes. The delay nodes specify the length of times during which the gates are to remain opened or closed.

5.3 Experiments and Discussion

Several cases were simulated to study the performance of traffic shaping. This section presents the results obtained from the simulations and evaluates the performance of shaping by comparing values of cell loss probability. For all cases, the following traffic parameters were used:

- Average burst duration = 200 ms
- Average silent duration = 250 ms
- Average rate = 69 kb/s
- Channel capacity = 1.544 Mb/s

The buffer size is varied from 530 bytes to 7950 bytes. The simulation results as cell loss probabilities are shown in Table 5-1. These values come from the simulation output file that automatically collect all statistics. For an example of all statistics, see *Appendix*.

Buffer size (bytes)	EXP1	EXP2	EXP3	EXP4	EXP5	EXP6	EXP7	EXP8	EXP9
530	47	43.6	34.8	26	25.1	20.9	19.4	19.3	14.3
1590	35.8	21.5	15.7	14.1	6.8	3.7	4.3	3.2	1.5
2650	26.1	12.1	5.2	3.2	2	1	0.9	0.7	0.3
3710	18.8	7.2	2.1	3.7	1.2	0.5	0.2	0	0
4770	13.8	3.4	1.6	0.6	0.3	0.1	0	0	0
5830	10.3	2.1	1.4	0.3	0	0	0	0	0
6890	6.6	1.9	0.7	0.1	0	0	0	0	0
7950	8	1.6	0.3	0	0	0	0	0	0

Table 5-1 Cell loss probabilities when average rate of traffic is 69 Kb/s and buffer size are varied from 530 bytes to 7950 bytes.

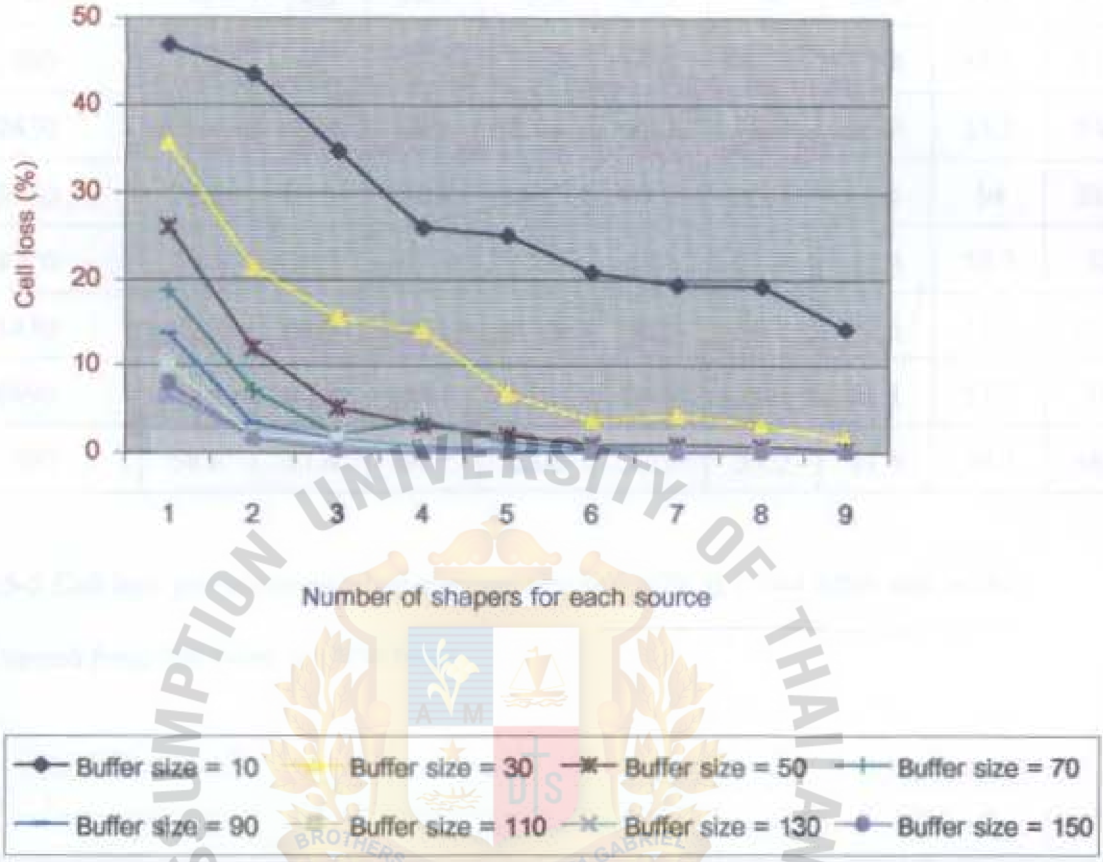

Figure 5-3 Compare performance of shaper when average rate of traffic is 69 Kb/s and buffer size varied from 530 bytes to 7950 bytes.

Fig. 5-3 compares results obtained for all cases. The results show that if the buffer size is more than 5830 bytes, cell loss probabilities are very close. This means, for the case in which traffic is not busy, if the buffer size is large, the number of shapers will almost not affect the performance of shaping. And then we change the average rate to 1.544 Mb/s. Results are shown in Table 5-2.

Buffer size (bytes)	EXP1	EXP2	EXP3	EXP4	EXP5	EXP6	EXP7	EXP8	EXP9
530	75	72.1	66.7	70.4	65.5	74	68.8	71.5	67.6
1590	73.5	68	74.7	44.7	61.5	67.1	71.6	58.1	57.6
2650	71.1	69.5	69.2	55.4	68.7	62.7	56.3	55.5	54.8
3710	74.6	68.8	52.7	69.3	67.2	65.7	64.1	54	53.4
4770	70.8	69.2	63.3	67.8	63.6	53.2	53.4	52.7	52
5830	72.7	69.5	67.1	65.5	60.9	53.2	52.3	51.4	52.2
6890	73.5	66.8	64.1	63.5	64.5	52.2	51.1	51.7	50.5
7950	58.6	67.4	64.7	63.6	52.5	51.2	51.6	50.2	48.7

Table 5-2 Cell loss probabilities when average rate of traffic is 1.544 Mb/s and buffer size is varied from 530 bytes to 7950 bytes

Buffer size (bytes)	EXP1	EXP2	EXP3	EXP4	EXP5	EXP6	EXP7	EXP8	EXP9
10600	65.4	66.3	64.7	52.1	50.4	50.2	48.2	47.8	45.7
21200	68.4	52.1	50.2	47.8	42.8	34.5	25.4	1.1	0
31800	60.8	50.2	45.5	34.5	21.9	0	0	0	0
42400	63.7	47.8	34.4	4.8	0	0	0	0	0
53000	50.3	42.8	21.8	0	0	0	0	0	0
63600	48.5	34.4	0	0	0	0	0	0	0

Table 5-3 Cell loss probabilities when average rate of traffic is 1.544 Mb/s and buffer size is varied from 10600 bytes to 63600 bytes.

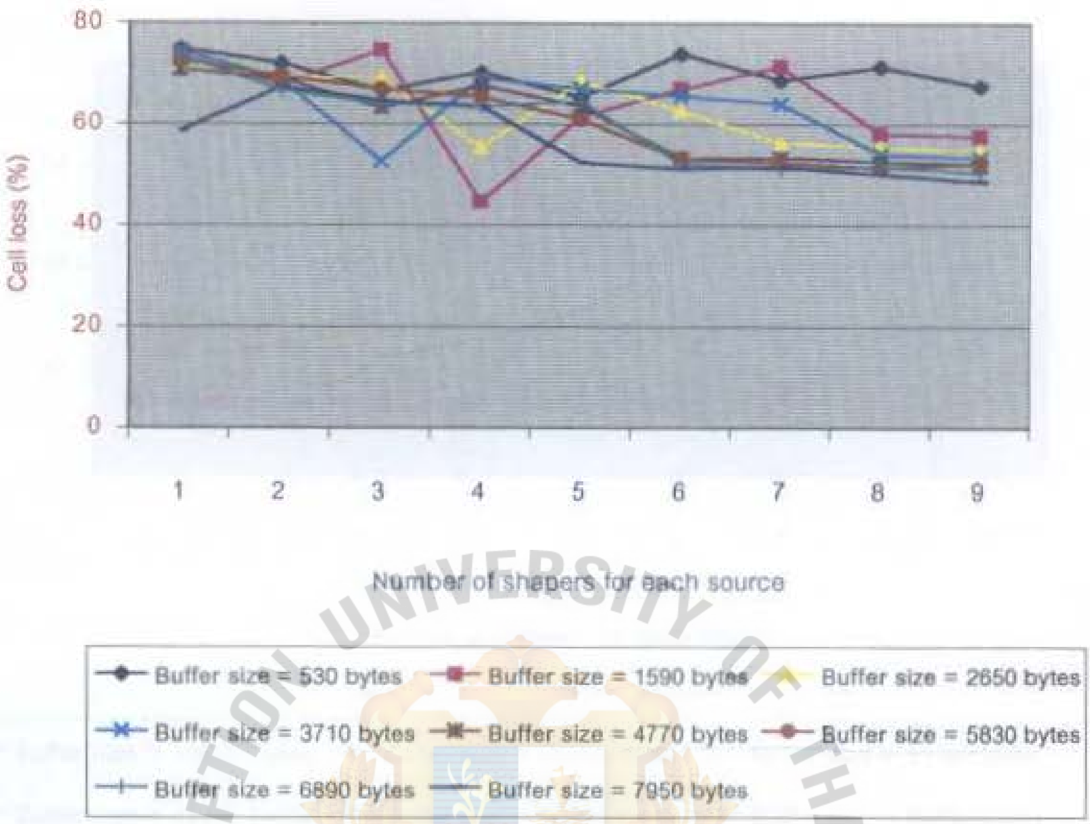

Figure 5-4 Compare performance of shaper when average rate of traffic is 1.544 Mb/s and buffer size varied from 530 bytes to 7950 bytes.

Fig. 5-4 shows the results when we use the same range of buffer size as previous experiment which are 530 bytes to 7950 bytes. We can see that cell loss probabilities are not different in this range of buffer size, although we increase the number of shapers for each source. This means we need larger size of buffer, thus we do another experiment by increasing the buffer size. For this experiment, we vary buffer size from 10600 bytes to 63600 bytes. The results are shown in Table 5-3.

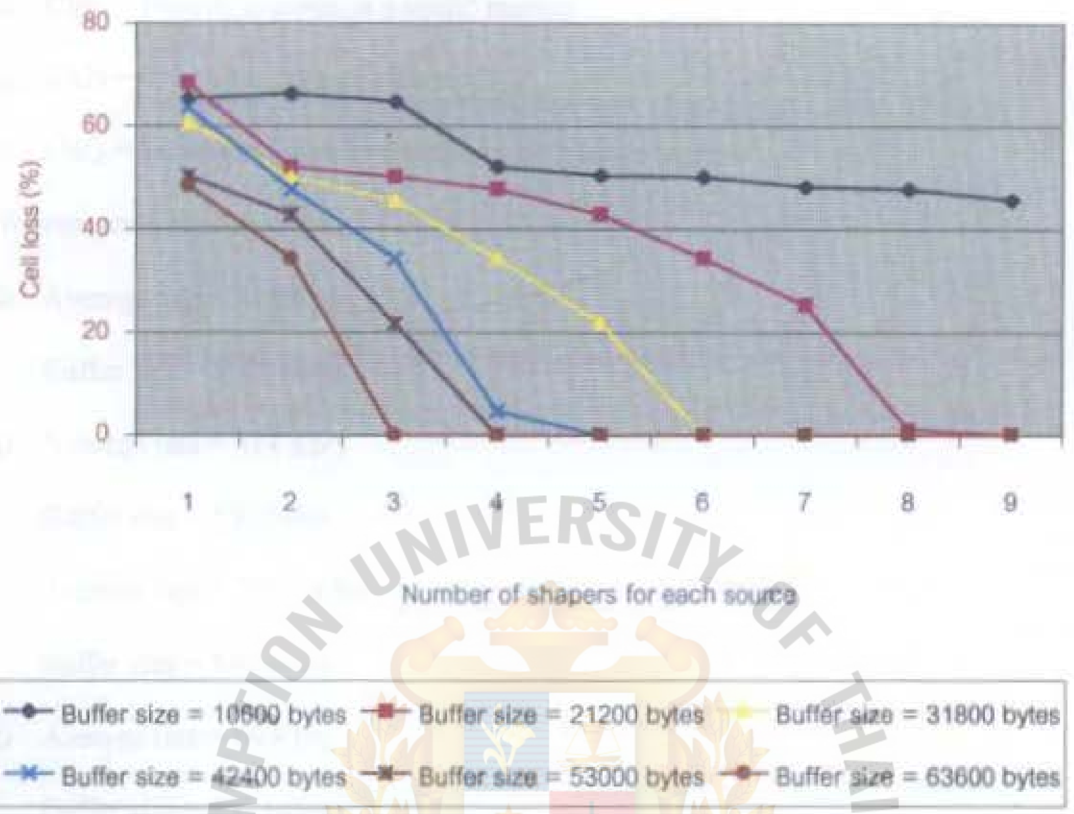

Figure 5-5 Compare performance of shaper when average rate of traffic is 1.544 Mb/s and buffer size varied from 10600 bytes to 63600 bytes.

From Fig. 5-5, we can see that, in this range of buffer size, the effect of buffer size to cell loss probabilities is more than the previous experiment (Fig 5-4). And from Fig. 5-4, If we consider 7950 bytes buffer with 8 shapers for each source, this means total capacity of buffer is 63600 bytes for each source, compare with one 63600 bytes-shaper for each source in Fig. 5-5. For the first case, cell loss probability is 50.2 % and for the second case, cell loss probability is 48.5 %. However, since only two-percent difference between both cases and using parallel shapers is complicated, thus it may be better if we use one large shaper instead of multiple small shapers.

There are many criteria to select cell from shapers. We studied 3 criteria:

- CYC = Priority is given in a cyclic manner
- RAN = Priority is given randomly
- LNQ = Priority is given to the shaper with longer queue

We run simulation in 4 cases:

- Average rate = 414 Kb/s

  Buffer size = 5300 bytes
- Average rate = 414 Kb/s

  Buffer size = 530 bytes
- Average rate = 103.5 Kb/s

  Buffer size = 530 bytes
- Average rate = 69 Kb/s

  Buffer size = 530 bytes

For all cases, the following parameter were used:

- Average burst duration = 200 ms
- Average silent duration = 250 ms
- Channel capacity = 1.544 Mb/s

Each case was run by using all above parameters.

The results as cell loss probabilities are shown in Table 5-4 to Table 5-7.

Number of shaper for each source	CYC	RAN	LNQ
1	45.6	43.6	42.8
2	26.6	32.3	29.4
3	22.9	31.3	18.6
4	15.5	20.3	27.4
5	19.7	24.7	11.1
6	15.8	15.8	12.9
7	9.7	14.9	15.1
8	13	16.4	9.3
9	8.4	12.3	13.4

Table 5-4 Cell loss probabilities when Average rate = 414 Kb/s and buffer size = 5300 bytes

Number of shaper for each source	CYC	RAN	LNQ
1	67.1	54.8	65.7
2	76.5	61.5	56.2
3	62.2	46.7	58.4
4	48.9	53.7	51.7
5	62.6	39.6	55.5
6	46.2	49.9	39.7
7	18.9	46.6	37.8
8	43.2	33.3	48
9	10.1	34	36.7

Table 5-5 Cell loss probabilities when Average rate = 414 Kb/s and buffer size = 530 bytes

Number of shaper for each source	CYC	RAN	ENQ
1	48.1	47.7	47.7
2	41	45.9	41.4
3	41.6	37.8	40.2
4	34	35.6	39
5	30.8	32.8	31.8
6	36	27.4	29.5
7	26.3	29.7	31.7
8	20.9	21.9	24.9
9	20.6	21	27.2

Table 5-6 Cell loss probabilities when Average rate = 103.5 Kb/s and buffer size = 530 bytes

Number of shaper for each source	CYC	RAN	ENQ
1	47.5	47	47
2	40	38.2	43.6
3	35.3	36.4	34.8
4	30.8	30.6	26
5	23.6	26.9	25.1
6	23.6	22.8	20.9
7	19.8	20.4	19.4
8	22.1	17.9	19.3
9	14.3	15.9	14.3

Table 5-7 Cell loss probabilities when Average rate = 69 Kb/s and buffer size = 530 bytes

Figure 5-6 Compare performance of shaping for scheduler when average rate = 414 Kb/s and buffer size is 5300 bytes.

Average rate = 414 Kb/s   Buffer = 530 bytes

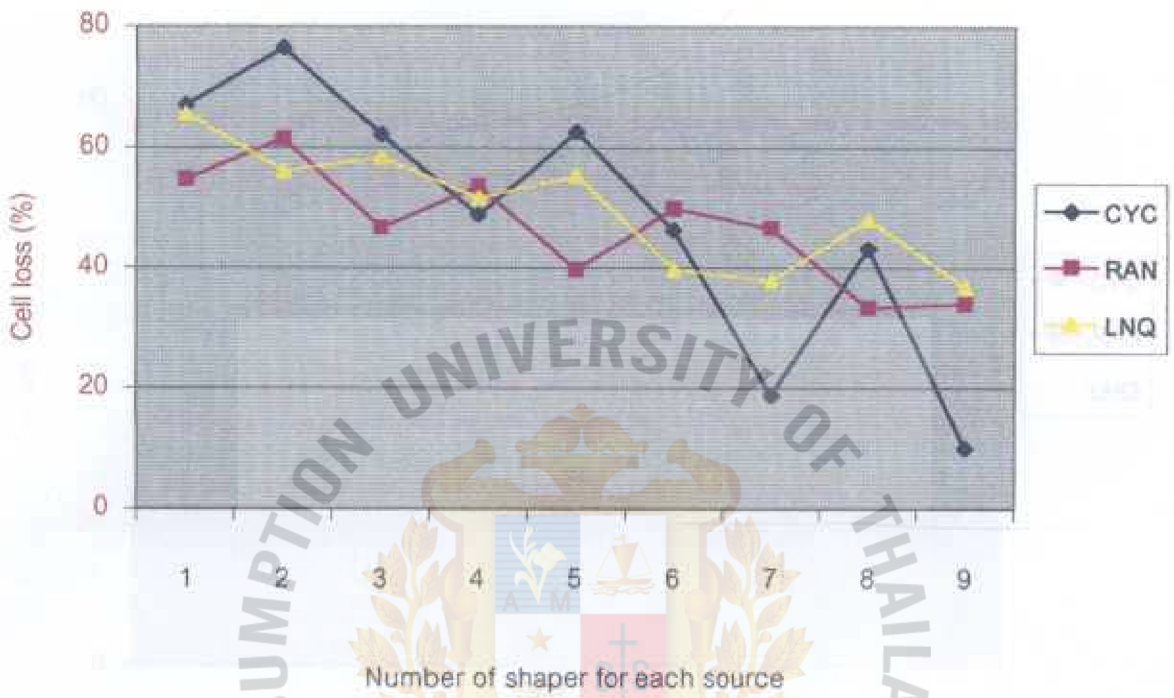

Figure 5-7 Compare performance of shaping for scheduler when average rate = 414 Kb/s and buffer size is 530 bytes.

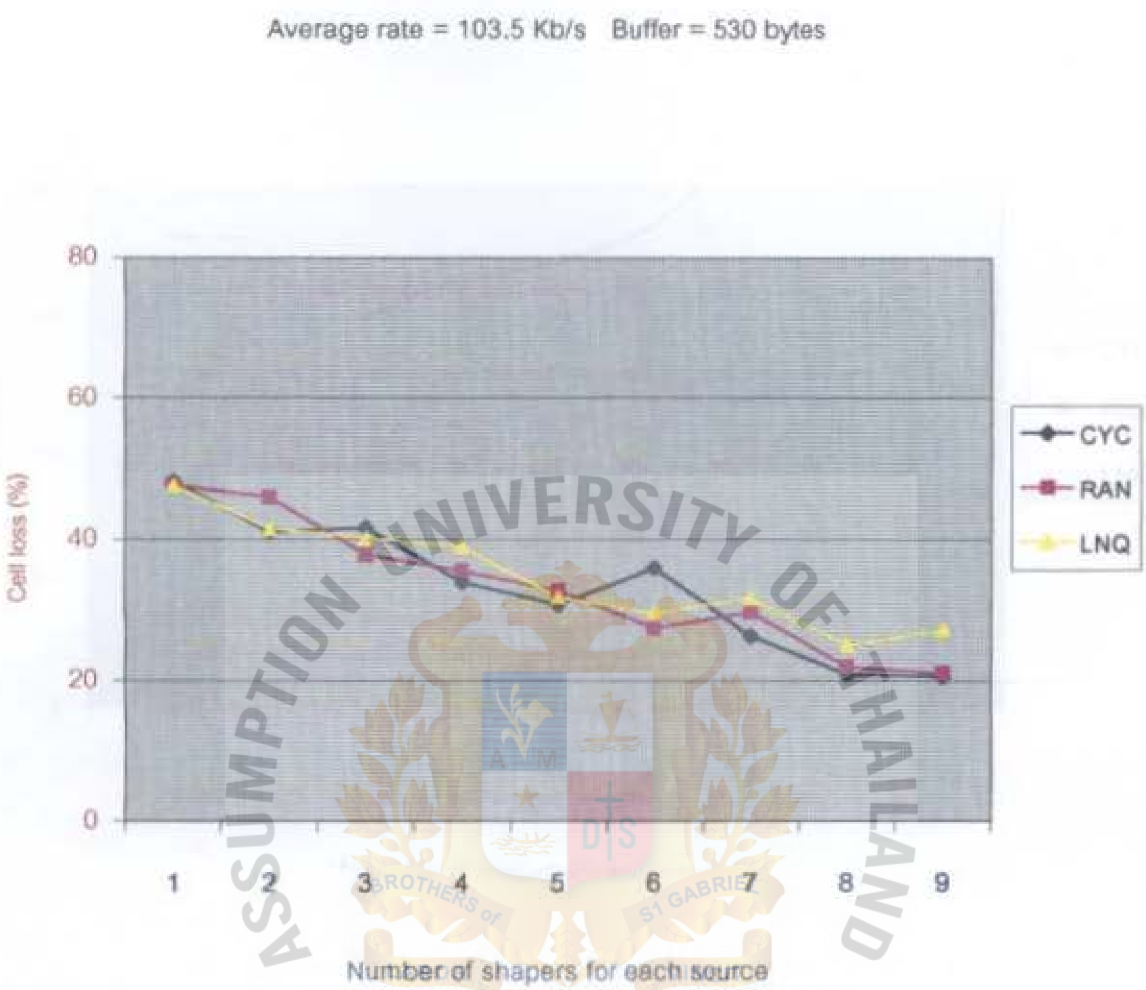

Figure 5-8 Compare performance of shaping for scheduler when average rate = 103.5 Kb/s and buffer size is 530 bytes.

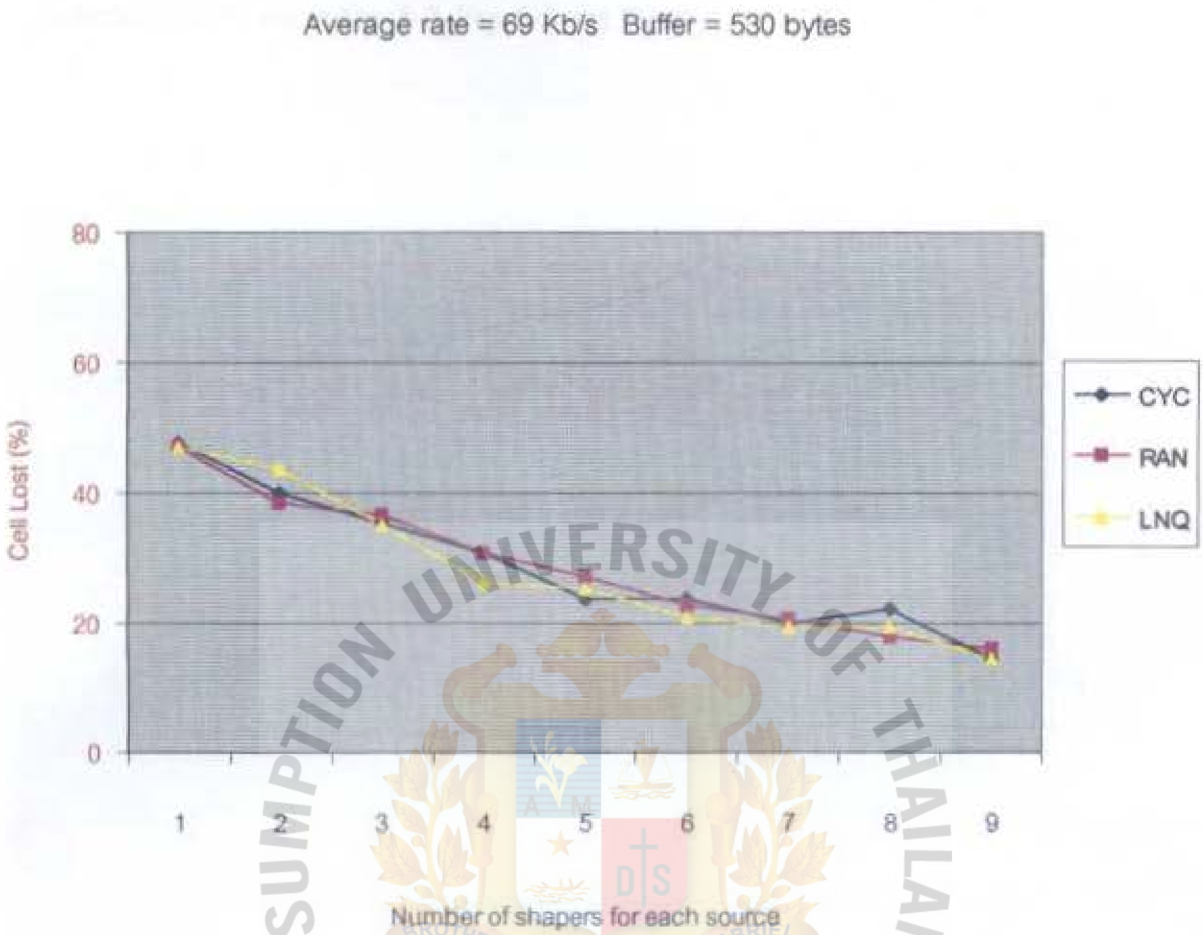

Figure 5-9 Compare performance of shaping for scheduler when average rate = 69 Kb/s and buffer size is 530 bytes.

Fig. 5-6 to Fig. 5-9 show all graphs which are plotted from the results we obtained. From Fig. 5-6 and Fig. 5-7 the results show that if buffer size is 5300 bytes, cell loss probabilities of each scheduler are close. But if buffer size is 530 bytes, cell loss probabilities of each scheduler are varied. This means the performance of shaping for each scheduler is closer, if buffer size is larger. And after interarrival time is increased, The results show that cell loss probabilities of each scheduler are closer.

This means when traffic is not busy, the performance of shaping does not depend on scheduler as shown in Fig. 5-7, Fig. 5-8 and Fig. 5-9.

CHAPTER 6 CONCLUSION

In this research, we evaluated the performance of traffic shaping for frame relay over ATM network. A simulation tool that used in this research is called EZSIM. We run the simulation by varying both of buffer size and number of shaper for each source. We compared the simulation results that we obtained. The results are shown that if the buffer size is large, the number of shaper does not affect the performance of traffic shaping. However, a certain size of buffer depends on characteristic of traffic. We also run the simulation to compare the performance of shaping between using one large shaper and parallel small shapers with the same total size. From the results, we can see that the performance of shaping for both cases in term of cell loss probability is somewhat not different.

We also studied about the criteria of scheduler for selecting cell from shapers. In this work, we studied 3 criteria, which are cyclic, random, and queue length based manner. First time we run the simulation by using 2 different size of buffer but the same average rate of traffic. And then we used the same buffer but different average rate of traffic. The results suggested that if the buffer size is large enough, the performance of shaping is somewhat independent of scheduler. And also the traffic, if the traffic is not busy, the criteria of scheduler does not affect the performance of traffic shaping.

REFERENCES

1. Sudhir S. Dixit and Sharad Kumar, "Traffic Descriptor Mapping and Traffic Control for Frame Relay Over ATM Network", *IEEE/ACM Transactions on Networking*, Vol. 6, No. 1, February 1998.
2. Dominique Gaiti and Guy Pujolle, "Performance Management Issues in ATM Networks: Traffic and Congestion Control", *IEEE/ACM Transactions on Networking*, Vol. 4, No. 2, April 1996.
3. Kyeong Soo Kim and Byeong Gi Lee,"Three-Level Traffic Shaper and Its Application to Source Clock Frequency Recovery for VBR Video Services in ATM Networks", *IEEE/ACM Transactions on Networking*, Vol. 3 No. 4, August 1995.
4. Walter J. Goralski, "Introduction to ATM network", McGraw-Hill Inc., 1995.
5. S. Dixit and S. Elby, "frame relay and ATM interworking", *IEEE Commun. Mag.*, vol. 34, pp.64-82, June 1996.
6. ITU-T Recommendation I.137, Traffic control and congestion control in B-ISDN, Mar.1993.
7. *Broadband Aspects of ISDN*, ITU-T Recommendation I.121, Blue Book, Nov.1988.
8. *B-ISDN User-Network Interface*, ITU-T Recommendation I.413, Dec.1990.
9. *Traffic Management Specifications Version 4.0*, ATM Forum, Apr.1996.
10. *Traffic Control and Congestion Control in B-ISDN*, ITU-T Recommendation I.371, June 1992.

APPENDIX

Disk file name: EXP1-1.OUT Date: 04/03/01

Current Time: 120000.00 Transient Period: 0.00

FR = (3,10) ATM = EXPON (6,10) SERVER = (0.27,10)

BURST = EXPON (200,10) SILENT = EXPON (250,10)

TERMINATE NODE

Termination count: INF (using run without animation)

Enter selection criterion: LNQ

At this node the entities are taken from nodes listed below

SHAPE 1 SHAPE 2

Q U E U E S (530 bytes):

NAME	MIN/MAX/LAST LENGTH	MEAN LENGTH	STD LENGTH	MEAN DELAY	STD DELAY
GATEF	0/421/ 0	42.02	65.87		
GATEA	0/513/ 0	31.12	59.79	185.20	354.03
SHAPA1	0/10/ 0	0.03	0.43	0.18	0.58
SHAPF1	0/10/ 0	0.04	0.49		

GATE STATISTICS:

NAME	TIME OPEN	TIME CLOSED	PERCENT OPEN	PERCENT CLOSE
GATEF	5.44E+04	6.56E+04	45.37	54.63
GATEA	5.03E+04	6.97E+04	41.95	58.05

F A C I L I T I E S:

NAME	NBR SRVRS	MIN/MAX/LAST UTILIZATION	MEAN UTLZ	STD UTLZ	MEAN IDLE	MEAN BUSY
SERVER	1	0/ 1/ 0	0.07	0.26	3.51	0.27

VARIABLES:

NAME	MEAN	STD	MIN	MAX	No.OBSRVD
COUNT1	3.18E+04	0.00E+00	3.18E+04	3.18E+04	31753
COUNT2	2.82E+04	0.00E+00	2.82E+04	2.82E+04	28162

Cell lost = 47%

Disk file name: EXP1-2.OUT Date: 04/03/01

Current Time: 120000.00 Transient Period: 0.00

FR = (3,10) ATM = EXPON (6,10) SERVER = (0.27,10)

BURST = EXPON (200,10) SILENT = EXPON (250,10)

TERMINATE NODE

Termination count: INF (using run without animation)

Enter selection criterion: LNQ

At this node the entities are taken from nodes listed below

SHAPE 1 SHAPE 2

Q U E U E S (1590 bytes):

NAME	MIN/MAX/LAST LENGTH	MEAN LENGTH	STD LENGTH	MEAN DELAY	STD DELAY
GATEF	0/337/59	44.26	63.94		
GATEA	0/175/ 4	21.50	31.84	130.66	193.05
SHAPA1	0/30/ 0	0.21	1.98	1.29	2.40
SHAPF1	0/30/ 0	0.25	2.16		

GATE STATISTICS:

NAME	TIME OPEN	TIME CLOSED	PERCENT OPEN	PERCENT CLOSE
GATEF	4.21E+04	5.79E+04	42.11	57.89
GATEA	4.24E+04	5.76E+04	42.39	57.61

F A C I L I T I E S:

NAME	NBR SRVRS	MIN/MAX/LAST UTILIZATION	MEAN UTLZ	STD UTLZ	MEAN IDLE	MEAN BUSY
SERVER	1	0/ 1/ 0	0.09	0.28	2.85	0.27

VARIABLES:

NAME	MEAN	STD	MIN	MAX	No.OBSRVD
COUNT1	3.20E+04	0.00E+00	3.20E+04	3.20E+04	32011
COUNT2	1.78E+04	0.00E+00	1.78E+04	1.78E+04	17817

Cell lost = 35.8%

Disk file name: EXP1-3.OUT Date: 04/03/01

Current Time: 120000.00 Transient Period: 0.00

FR = (3,10) ATM = EXPON (6,10) SERVER = (0.27,10)

BURST = EXPON (200,10) SILENT = EXPON (250,10)

TERMINATE NODE

Termination count: INF (using run without animation)

Enter selection criterion: LNQ

At this node the entities are taken from nodes listed below

SHAPE 1 SHAPE 2

QUEUES (2650 bytes):

NAME	MIN/MAX/LAST LENGTH	MEAN LENGTH	STD LENGTH	MEAN DELAY	STD DELAY
GATEF	0/632/55	51.26	88.32	153.05	262.64
GATEA	0/195/13	22.38	33.38	135.68	200.54
SHAPA1	0/50/ 0	0.37	3.26	2.25	3.78
SHAPF1	0/50/ 0	0.55	4.02	1.64	3.47

GATE STATISTICS:

NAME	TIME OPEN	TIME CLOSED	PERCENT OPEN	PERCENT CLOSE
GATEF	4.06E+04	4.74E+04	46.17	53.83
GATEA	3.89E+04	4.91E+04	44.20	55.80

FACILITIES:

NAME	NBR SRVRS	MIN/MAX/LAST UTILIZATION	MEAN UTLZ	STD UTLZ	MEAN IDLE	MEAN BUSY
SERVER	1	0/ 1/ 0	0.10	0.30	2.44	0.27

VARIABLES:

NAME	MEAN	STD	MIN	MAX	No.OBSRVD
COUNT1	3.25E+04	0.00E+00	3.25E+04	3.25E+04	32452
COUNT2	1.14E+04	0.00E+00	1.14E+04	1.14E+04	11436

Cell lost = 26.1%

Disk file name: EXP1-4.OUT Date: 04/03/01

Current Time: 120000.00 Transient Period: 0.00

FR = (3,10) ATM = EXPON (6,10) SERVER = (0.27,10)

BURST = EXPON (200,10) SILENT = EXPON (250,10)

TERMINATE NODE

Termination count: INF (using run without animation)

Enter selection criterion: LNQ

At this node the entities are taken from nodes listed below

SHAPE 1 SHAPE 2

Q U E U E S (3710 bytes):

NAME	MIN/MAX/LAST LENGTH	MEAN LENGTH	STD LENGTH	MEAN DELAY	STD DELAY
GATEF	0/430/221	44.49	69.55	130.75	207.52
GATEA	0/244/77	22.27	35.12	132.77	212.31
SHAPA1	0/70/ 0	0.58	4.60	3.50	5.37
SHAPF1	0/70/ 0	0.97	6.35	2.92	5.27

GATE STATISTICS:

NAME	TIME OPEN	TIME CLOSED	PERCENT OPEN	PERCENT CLOSE
GATEF	3.65E+04	4.45E+04	45.01	54.99
GATEA	3.52E+04	4.58E+04	43.48	56.52

F A C I L I T I E S:

NAME	NBR SRVRS	MIN/MAX/LAST UTILIZATION	MEAN UTLZ	STD UTLZ	MEAN IDLE	MEAN BUSY
SERVER	1	0/ 1/ 0	0.11	0.31	2.21	0.27

VARIABLES:

NAME	MEAN	STD	MIN	MAX	No.OBSRVD
COUNT1	3.26E+04	0.00E+00	3.26E+04	3.26E+04	32616
COUNT2	7.53E+03	0.00E+00	7.53E+03	7.53E+03	7534

Cell lost = 18.8%

Disk file name: EXP1-5.OUT Date: 04/03/01

Current Time: 120000.00 Transient Period: 0.00

FR = (3,10) ATM = EXPON (6,10) SERVER = (0.27,10)

BURST = EXPON (200,10) SILENT = EXPON (250,10)

TERMINATE NODE

Termination count: INF (using run without animation)

Enter selection criterion: LNQ

At this node the entities are taken from nodes listed below

SHAPE 1 SHAPE 2

Q U E U E S (4770 bytes):

NAME	MIN/MAX/LAST LENGTH	MEAN LENGTH	STD LENGTH	MEAN DELAY	STD DELAY
GATEF	0/472/ 0	45.36	75.53	136.81	223.60
GATEA	0/186/98	21.18	31.79	123.91	188.01
SHAPA1	0/90/ 0	0.83	6.11	4.93	7.21
SHAPF1	0/90/ 1	1.41	8.42	4.25	6.94

GATE STATISTICS:

NAME	TIME OPEN	TIME CLOSED	PERCENT OPEN	PERCENT CLOSE
GATEF	3.43E+04	4.17E+04	45.16	54.84
GATEA	3.44E+04	4.16E+04	45.21	54.79

F A C I L I T I E S:

NAME	NBR SRVRS	MIN/MAX/LAST UTILIZATION	MEAN UTLZ	STD UTLZ	MEAN IDLE	MEAN BUSY
SERVER	1	0/ 1/ 1	0.12	0.32	2.05	0.27

VARIABLES:

NAME	MEAN	STD	MIN	MAX	No.OBSRVD
COUNT1	3.27E+04	0.00E+00	3.27E+04	3.27E+04	32696
COUNT2	5.24E+03	0.00E+00	5.24E+03	5.24E+03	5236

Cell lost = 13.8%

Disk file name: EXP1-6.OUT Date: 04/03/01

Current Time: 120000.00 Transient Period: 0.00

FR = (3,10) ATM = EXPON (6,10) SERVER = (0.27,10)

BURST = EXPON (200,10) SILENT = EXPON (250,10)

TERMINATE NODE

Termination count: INF (using run without animation)

Enter selection criterion: LNQ

At this node the entities are taken from nodes listed below

SHAPE 1 SHAPE 2

Q U E U E S (5830 bytes):

NAME	MIN/MAX/LAST LENGTH	MEAN LENGTH	STD LENGTH	MEAN DELAY	STD DELAY
GATEF	0/454/14	43.90	71.59	131.47	218.71
GATEA	0/273/ 0	20.15	35.89	123.63	223.00
SHAPA1	0/110/ 0	0.74	5.98	4.55	7.15
SHAPF1	0/110/ 0	1.74	10.47	5.22	8.33

GATE STATISTICS:

NAME	TIME OPEN	TIME CLOSED	PERCENT OPEN	PERCENT CLOSE
GATEF	3.51E+04	3.79E+04	48.11	51.89
GATEA	3.49E+04	3.81E+04	47.75	52.25

F A C I L I T I E S:

NAME	NBR SRVRS	MIN/MAX/LAST UTILIZATION	MEAN UTLZ	STD UTLZ	MEAN IDLE	MEAN BUSY
SERVER	1	0/ 1/ 1	0.12	0.32	1.98	0.27

VARIABLES:

NAME	MEAN	STD	MIN	MAX	No.OBSRVD
COUNT1	3.25E+04	0.00E+00	3.25E+04	3.25E+04	32532
COUNT2	3.72E+03	0.00E+00	3.72E+03	3.72E+03	3725

Cell lost = 10.3%

Disk file name: EXP1-7.OUT Date: 04/03/01

Current Time: 120000.00 Transient Period: 0.00

FR = (3,10) ATM = EXPON (6,10) SERVER = (0.27,10)

BURST = EXPON (200,10) SILENT = EXPON (250,10)

TERMINATE NODE

Termination count: INF (using run without animation)

Enter selection criterion: LNQ

At this node the entities are taken from nodes listed below

SHAPE 1 SHAPE 2

Q U E U E S (6890 bytes):

NAME	MIN/MAX/LAST LENGTH	MEAN LENGTH	STD LENGTH	MEAN DELAY	STD DELAY
GATEF	0/374/47	36.89	63.60	111.27	192.32
GATEA	0/261/ 0	22.64	39.96	136.91	240.24
SHAPA1	0/130/ 0	0.91	7.10	5.51	8.90
SHAPF1	0/130/ 0	1.99	11.77	6.01	9.60

GATE STATISTICS:

NAME	TIME OPEN	TIME CLOSED	PERCENT OPEN	PERCENT CLOSE
GATEF	3.57E+04	3.43E+04	51.05	48.95
GATEA	3.28E+04	3.72E+04	46.91	53.09

F A C I L I T I E S:

NAME	NBR SRVRS	MIN/MAX/LAST UTILIZATION	MEAN UTLZ	STD UTLZ	MEAN IDLE	MEAN BUSY
SERVER	1	0/ 1/ 0	0.12	0.33	1.89	0.27

VARIABLES:

NAME	MEAN	STD	MIN	MAX	No.OBSRVD
COUNT1	3.24E+04	0.00E+00	3.24E+04	3.24E+04	32408
COUNT2	2.30E+03	0.00E+00	2.30E+03	2.30E+03	2302

Cell lost = 6.6%

Disk file name: EXP1-8.OUT Date: 04/03/01

Current Time: 120000.00 Transient Period: 0.00

FR = (3,10) ATM = EXPON (6,10) SERVER = (0.27,10)

BURST = EXPON (200,10) SILENT = EXPON (250,10)

TERMINATE NODE

Termination count: INF (using run without animation)

Enter selection criterion: LNQ

At this node the entities are taken from nodes listed below

SHAPE 1 SHAPE 2

Q U E U E S (7950 bytes):

NAME	MIN/MAX/LAST LENGTH	MEAN LENGTH	STD LENGTH	MEAN DELAY	STD DELAY
GATEF	0/515/37	54.93	84.84	164.79	254.88
GATEA	0/267/36	24.25	36.85	145.40	217.67
SHAPA1	0/150/ 0	1.11	8.32	6.70	9.25
SHAPF1	0/150/ 0	2.71	14.70	8.16	11.30

GATE STATISTICS:

NAME	TIME OPEN	TIME CLOSED	PERCENT OPEN	PERCENT CLOSE
GATEF	2.82E+04	4.28E+04	39.72	60.28
GATEA	2.93E+04	4.17E+04	41.22	58.78

F A C I L I T I E S:

NAME	NBR SRVRS	MIN/MAX/LAST UTILIZATION	MEAN UTLZ	STD UTLZ	MEAN IDLE	MEAN BUSY
SERVER	1	0/ 1/ 0	0.12	0.33	1.91	0.27

VARIABLES:

NAME	MEAN	STD	MIN	MAX	No.OBSRVD
COUNT1	3.26E+04	0.00E+00	3.26E+04	3.26E+04	32563
COUNT2	2.84E+03	0.00E+00	2.84E+03	2.84E+03	2841

Cell lost = 8%