TRUNK SIZING IN PACKET NETWORKS – THE EFFECT ON USER QUALITY OF SERVICE

By

MARK E. ALLEN

Bachelor of Science
University of Kansas
Lawrence, Kansas
1990

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1993

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the degree of
DOCTOR OF PHILOSOPHY
May, 1997

TRUNK SIZING IN PACKET NETWORKS – THE EFFECT ON

USER QUALITY OF SERVICE

Thesis Approved:

____
Thesis Adviser

____

____

____

____
Dean of the Graduate College

ACKNOWLEDGMENTS

This study was possible through the support of WorldCom of Tulsa, Oklahoma. The assistance of WorldCom engineers Loren Lovell, Chris Painter, Robert Eppich, and Dave Barmann over the past two years was greatly appreciated as was the guidance and wisdom of my OSU committee members and teachers Dr. Rao Yarlagadda, Dr. Martin Hagan, and Dr. Blayne Mayfield.

The special teaching abilities, guidance and support of my advisor Dr. George Scheets provided me with the tools, for which I am deeply indebted, to perform this research.

I would especially like to thank my wife Annie for her patience, love, and understanding during this endeavor. Without her encouragement it would not have been possible.

TABLE OF CONTENTS

I. Introduction

Importance of Packet Networks

In the recent decade, data has jumped to the forefront to become one of the most important communications mediums. It is now common for data traffic to exceed voice traffic on the large carrier's networks. This surge has been fueled by rapid advances in the enabling technology for data communications: the computer. Users are finding that advanced computing and storage resources allow them to process and view data more rapidly and in more detail than ever before. The role of the computer in almost every key aspect of the business environment, from the front desk to the warehouse inventory system, has also resulted in corporate networks growing up not just within a single building, but across the country and world. For example, American Airline's SABRE system is accessed by thousands of travel agents around the world. The central reservation system stores 4.4 trillion bytes of flight schedules and reservations that would occupy more than 2 billion pages if printed on paper [1]. Further accelerating this growth is widespread interest in computer communications for the home. In the past ten years, the Internet has evolved from a small network linking critical defense and academic institutions to one where school age children explore, recipes are exchanged, and companies such as Walt Disney promote their latest movies and theme parks on the Web [2].

The fact that data traffic growth is far exceeding that of voice is forcing large communications providers to rethink their approach to designing networks. The statistical nature of data traffic differs from voice, making well established network sizing tools such as Erlang formulas [6] ineffective. Further compounding the problem is the rate at which these networks are growing. In some cases, the growth is literally exponential [33]. Network providers are finding it is more cost-effective to grow their network as quickly as possible rather than trying to carefully plan what resources should be allocated and when.

The time is soon approaching when, as is the case with voice network design, a critical analysis of resource allocation will be necessary to maximize revenues in a market that has both fierce competition and low margins. The purpose of the research presented here is to develop some techniques for sizing Inter-Machine Trunks (IMTs) for Frame Relay networks and predicting performance impacts on the user's Quality of Service (QoS).

Motivation for this Research

Discussions with engineers responsible for designing large Frame Relay networks has made it clear that the number of tools and heuristics on which they can rely is small. As a result, it has been necessary for them to design and size the network by trial and error using ad hoc techniques that are imprecise and require allocating more network resources than are probably required. Currently, no straight-forward mathematically based methodologies exist. Furthermore, it is believed that all providers of Frame Relay are having similar difficulties [33].

Improving this situation will have benefits. First, the cost savings from no longer over-designing the network should be significant. There is a good chance that the current network has excess bandwidth allocated on some of the trunks. Second, the overall performance of the network will be improved - meaning more satisfied customers. Another important benefit is that the network sizing and designing process will become more proactive rather than reactive. This should allow engineers to increase the performance of the network before problems occur rather than waiting to react to a customer complaint. Finally, some parts of this research will be provided to customers so they will have a better understanding of what kind of service they can expect based on their choices of product offerings.

Evolution of Frame Relay

This section gives some background explaining how Frame Relay came about and what type of users require Frame Relay.

Time Division Multiplexing

From its initial inception up until the early 60's the telephone system was a 100% analog system. The central office switches were "step-by-step" or "crossbar" switches which relied on mechanical switching hardware. The communications between central offices and tandem switches consisted of modulating several channels using frequency division modulation (FDM) on a microwave channel. The local loop between the customer and the central office was analog as well [3].

In the mid '90s the only component of the system that remains predominantly analog is the local loop. Digital switches using time slot interchange (TSI) have replaced the mechanical switches. Time division multiplexing (TDM) over fiber-optic cable or digital microwave has replaced the analog microwave FDM system for transporting traffic between switches. Modems are one way that users have found of overcoming the remaining analog local loop. With the relative high quality of the current system, uncompressed modem transfer rates of up to 28.8 kbits/sec are possible, however many users find these rates to be less than adequate.

For data, one alternative to modem communications is dedicated private data lines which provide the customer with a digital local loop. This type of circuit was initially introduced by AT&T under the name Dataphone Digital Service (DDS). These circuits have been used by businesses during recent years and are available in speeds ranging from 56kbits/sec up to 45Mbits/sec. The main drawback is that they are not switched, so the customer generally pays a considerable fixed monthly charge whether he is using the line or not.

Narrowband Integrated Services Data Network (Narrowband ISDN) is a technology designed to offer switched data service directly to the customer. As of yet, it has received a lukewarm reception primarily due to its relatively high cost and limited availability. However, this situation is improving. At the current time, users in need of high bandwidth at affordable prices are finding some attractive solutions in the form of packet based data networks.

X.25

X.25 has been the most commonly used packet data network. An X.25 network consists of packet switches connected together in some type of mesh configuration using 56kb/sec data circuits. A user connects to the network via a packet assembler/disassembler (PAD) which accepts a standard serial data protocol such as RS-232-C [32]. A complete wide area data network can be constructed by connecting terminals, printers, and computers to the network. Normally the dedicated lines, used as trunks, are purchased from a service provider.

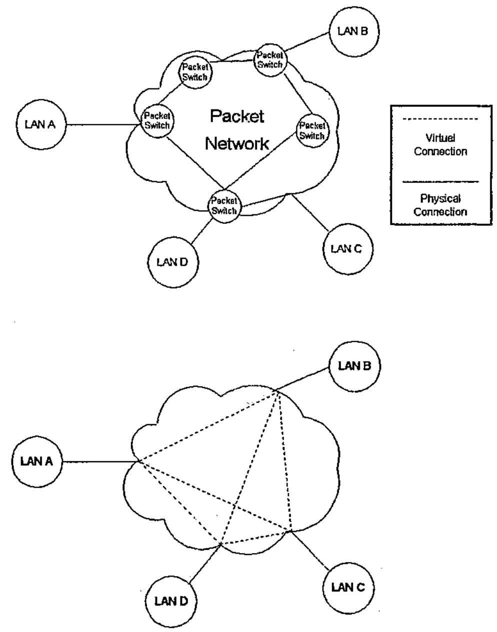

Figure I-1. A packet network connecting four local area networks (LANs). Note the difference between the physical connections and the virtual connections.

The obvious advantage to such a network is that the cost of the private lines is shared between several different devices. This means that while some devices are idle others may be using the network. The switches contain buffers that store incoming packets and flow-control the connected devices until spare capacity is available for transmission out of the switch. This type of resource sharing is called statistical multiplexing.

Figure I-1 shows a simple packet network being used to connect several Local Area Networks (LANs). The connections, as seen by the users, are referred to as virtual connections. The characteristics of such virtual connections will be discussed in Chapter II. The key point here is that these virtual connections can differ from the physical connections, which are the paths the data actually takes over the physical network. Notice that in this example (Figure I-1) there is no packet switch near LAN C. In fact, LAN C enters the packet network at the same switch as LAN D. However, from the standpoint of the virtual connections this is transparent.

X.25 has an important place in data communications, but is beginning to show its age. First, the mechanism that is used for communicating between switches is extremely robust and uses strict error checking and flow control. These were included in the original design because of the poor quality of the lines at that time. On today's high quality digital lines, this robustness is overkill due to the low probability of bit errors occurring. Finally, X.25 is not widely available at speeds greater than 56kb/s, which proves to be insufficient bandwidth for many of today's users.

While waiting for Broadband ISDN or Asynchronous Transfer Mode (ATM) to come and fill this need for higher speed packet networks, standards were developed for a new protocol, patterned after X.25, that could use inter-switch trunks faster than 45 Mbits/s with much less flow control and error checking. The assumption that most of the flow control and error checking could be eliminated was also based on the fact that TCP and other modern-day protocols would handle these things themselves. The low-level (level 2 in the OSI stack) protocol defined by these standards and known as Frame Relay combines the advantages of statistical multiplexing, namely efficient resource sharing, with the advantages of TDM, known for its low delay [4].

The first public offering of Frame Relay was made by WilTel in 1991. Since that time, every major carrier has begun offering it as a service. Between 1993 and 1995, the total worldwide revenues for Frame Relay went from 80 million to 556 million dollars, nearly a 7 fold increase and are expected to double to $1 billion for 1996 [4].

Packet Network Design Issues

In designing packet networks there are several issues that must be considered. Considerable research has gone into all major areas of network design because there is such an opportunity for money savings. A noteworthy example of the gains that can be achieved by research is Karmarkar's optimization method which allowed AT&T to save more than $1 billion on their telephone network [34].

Generally, finding the truly optimum network design is impossible. First, specifying all the parameters that constitute the objective function, or function to be minimized, and associating appropriate costs to each parameter is a formidable task by itself. Second, if such a function could be obtained, the computational resources to solve it for a network of worthwhile size would be prohibitive. As a result, the different sub-problems are generally considered somewhat independently of each other. A high level overview of some of these sub-problems follows ([5] and [6] serve as an excellent starting point for an explanation of these issues):

The first type of network design problems are those that fall into the category of topology problems. The solutions to these problems address: how many switches should be used; how the switches should be interconnected; and how the traffic should be routed between the switches in order to minimize both the average delay and the cost of the network. Examples of such research can be found in [7,8,9,35,36]. This type of research assumes that the engineer has flexibility over the location of trunks and switches and the routing algorithms used in the network. Often the engineer does not have this flexibility. Additionally, in these papers, relatively simple models for the traffic are used because the emphasis is at a higher level.

A second category of problems addressed in the literature is the problem of designing and analyzing the performance of the packet switches themselves. These works usually involve queuing theory and numerical calculations to compute the probabilities of buffer overflow. Such research can be found in [23-26,37-41]. The most detailed models for traffic sources can be found in this category of research.

A third area of research, which falls somewhere between the topology studies and the switch studies in terms of viewing the forest or the trees, investigates the "effective-bandwidth" on the trunks connecting packet switches together. This type of research attempts to describe the characteristics of this large traffic mixture by modeling it as a stochastic process where the first and second order statistics are matched to real measurements. In some cases, this is done in the frequency domain by describing the power spectrum of the process. In theory, the results of these studies should aid the designer in sizing the trunks -- a problem closely related to the one at hand. One main difficulty is obtaining meaningful measurements in order to perform the fitting. Another is that the particular protocol has an influence on the results. None of the papers in this area, such as [42-44], address these specifics.

Overview of this Research

The research that will be presented in the following Chapters focuses on how the size of the trunks between Frame Relay switches affects the quality of service as seen by the user. The definition of quality will be given in the following section.

It will be assumed that the location and number of switches is fixed. Also, it will be assumed the method of routing the virtual circuits through the network cannot be changed and also the design of the switches, in terms of buffer sizes and internal operation, is set. The main variable that can be modified will be the size, or bandwidth allocation, of the trunks.

To reiterate, the two main reasons that results from [42-44] cannot be used directly are: 1) These works leave out the protocol-specific (in this case Frame Relay) details that have a significant impact on the results, and 2) These works assume that adequate measurements of the actual traffic are available in order to perform the analysis.

Once the relationship between trunk sizing and quality is understood, it should be possible to find what degree of quality improvement can be obtained by increasing the size of the trunks.

How is Quality Defined?

The quality, or performance, of an individual virtual circuit can be broadly quantified via three basic parameters which are interrelated. Collectively, these will be referred throughout this paper as the Quality of Service (QoS).

<u>Throughput</u> is the amount of information bandwidth available to the user on his virtual connection. In cases of light load, this will equal the maximum rate the customer can send data onto the network. As delay and blocking increase, throughput can decrease to zero.

Delay is the time duration between a packet entering and leaving the network. Delay is composed of propagation delay, or the time it actually takes the information to propagate through the fiber and wire, and queuing delay which is the total time the packets sit awaiting service by the switches. When throughput is very low, the buffers in the switch can fill up so that frame discards, or blocking, will occur. When this happens, the delay perceived by the customer is very high since the frame must be retransmitted.

Blocking: When queues reach their maximum length, additional frames that arrive are blocked, or in Frame Relay terms, discarded. Blocking makes the perceived delay increase and the effective throughput decrease.

Contributions of this Research

The area of packet network design is a heavily plowed field and several good results exist that will aid in performing this research. The new components in this paper, however, relate to how these results will be applied and the mathematical approximations that can be used for the special case of Frame Relay. The uniqueness of the results here are due, in part, to the specific characteristics of Frame Relay which differ from Asynchronous Transfer Mode (ATM) (a topic heavily investigated in the literature) in some important ways. The main difference is ATM uses a fixed (53 byte) packet size whereas Frame Relay uses a variable length packet, or frame. Another important difference is that ATM networks are slated to use more sophisticated management and monitoring systems that will aid network design engineers somewhat more than Frame Relay monitoring systems. But since ATM is not yet widely available this remains to be seen.

The other subject unique to this research is the method that will be used to estimate, from the statistics provided by the packet switches, the traffic flow into the network. As mentioned above, other research assumes this information is known or can be easily obtained. In fact, obtaining this information is one of the most difficult tasks.

The Role of Simulation in this Research

As packet networks increase in complexity and size, the current trend is towards using discrete event simulation as a means of predicting performance. The research presented here relies on simulation, but an effort is made to approximate and validate results in mathematical form as much as possible.

Several packages exist for simulating communication networks [10]. One drawback to using such off-the-shelf packages is that they often lack the implementation of specific details critical to the particular type of network being considered. As a result, most simulations for this research were performed by programs written in C++, but they could have been implemented in any of the special simulation languages such as SLAM or SIMSCRIPT II. Another drawback to using simulation packages is that for large complex networks, the computational requirements to perform the simulation are prohibitive requiring that small pieces of the network be analyzed separately. The process of dissecting the network in a meaningful way is no small task.

Averill and McComas [10] offer some advice on simulation that is worth repeating here :

> ... perhaps only 30 to 40 percent of the total work in a sound simulation study is actually model "programming." Even if a simulation product could be developed with unlimited flexibility and essentially zero model programming time, there would still be a number of significant methodological issues that the simulation analyst would have to face. These include problem formulation, collection of the appropriate information and data, model validation, modeling of message traffic, and design and analysis of simulation runs.

The Basic Approach

The approach taken in performing this research can be seen in the arrangement of the Chapters. First, an investigation of Frame Relay will be done to gain an understanding the details of the protocol that are important to performing this research. This is described in Chapter II.

Next, a discussion of traffic models and their important characteristics will be contained in Chapter III. This discussion of traffic models is necessary because they will be used to represent the actual traffic going into the network. A simple traffic model will be proposed which, as it turns out, will need to be modified slightly as more information is obtained about the actual traffic.

In Chapter IV, a description of the system used by Frame Relay to collect traffic statistics will be provided. These statistics will be used to fit the model to the actual traffic. This fitting process, as well as any necessary modifications to the model, will be described in Chapter V. The modified model, a three-state model with hyper-exponentially distributed off-times, is shown to fit the measured statistics. The problem of fitting the model rapidly rather than by trial and error is addressed.

Chapter VI will discuss the details of a simulation program that was developed to simulate Frame Relay's method of collecting statistics. This program, and some approximations to the simulation program will be examined in detail. These approximations allow the three-state traffic model to be rapidly fit to the measured statistics. The properties of a mixture of source model traffic will be investigated in Chapter VII. In particular, the mean, variance, and autocorrelation of the aggregate of several three-state traffic models is investigated.

Chapter VIII will describe some equations that can be used to predict the performance on a trunk given the buffer sizes, trunk speed, and three-state traffic models for the sources traversing the trunk.

Finally, the conclusion will summarize the developments made during the research and how they can be applied to aid in the network design problem. Some areas where further research is warranted will also be described.

II. Fundamentals of Frame Relay

This chapter provides an overview of the implementation details of Frame Relay which pertain to this research. For more wide ranging information see books on the subject such as [4] or [11].

The Frame Format

The frame format for Frame Relay is simple (See Fig II-1). The first byte is a starting delimiter. The following two bytes contain the header composed of the Data Link Connection Identifier (DLCI), ten bits which identify the endpoints of the virtual connection, and six other bits that are used as status flags.

The user's data follows and can be of variable size. As discussed in chapter I, the variable size of a frame makes Frame Relay different than "cell" based systems like ATM and makes it best suited for data. Complete packets from Ethernet and Token Ring networks can be placed inside the user data portion of the frame.

Finally, the last two bytes are the frame check sequence which contain information necessary to see if the frame has become corrupt in some way.

0 1 1 1 1 1 1 0					1 Byte
DLCI			C/R	EA	2 Bytes*
DLCI	FECN	BECN	DE	EA	
User Data					1 - 8192 Bytes
Frame Check Sequence					2 Bytes
Frame Check Sequence					
0 1 1 1 1 1 1 0					1 Byte

* 2 Bytes is the default although 3 and 4 are allowed.

Source - Network Services Technology Overview, Cascade Communications Company, 1994

Figure II-1. Frame structure for Frame Relay.

The CIR, Bc and Be

The customer connects to the Frame Relay network via an access line. The speed of the access line determines the rate at which information is offered to the network and is known as the "port speed." When the customer is offering traffic, it always arrives at the port speed -- in essence the port has only two states: on or off. But, the average rate, determined by considering the amount of time the port is in either of these two states, is also important.

Three parameters important to Frame Relay are the Committed Information Rate (CIR), the Committed Burst Size (B_C) and the Excess Burst Size (B_e). These parameters are defined below:

1) CIR - Committed Information Rate - The transmission rate the service provider arranges to give the customer. Because of the nature of packet networks, this cannot be truly guaranteed, but the service provider makes every effort to provide this rate to the customer. This parameter is chosen mainly by estimating the average throughput the customer needs to see even in times of high network utilization. In some cases, customers may choose a low CIR when the port speed connection is high. Two primary instances are: 1) the customer expects the need for the connection to grow over time and thus wants to reserve a small amount of throughput initially with plans of increasing it later as necessary and 2) when many virtual connections with low CIRs share a common port connection.

2) Bc - The committed burst size - The number of bits a customer can send during a specified time interval without the data being marked "discard-eligible". To mark the frame discard eligible, the "DE" bit in the frame header is set to a logic 1.

3) Be - The excess burst parameter - This is the number of bits, exceeding Bc, that can be sent during the time interval before the data is subject to being discarded.

Generally, when a customer orders a virtual connection on a Frame Relay network, he specifies the CIR, the Port Connection speed, Bc and Be. Using these settings, T_C is the measurement period for determining the discard status of the frames and is computed as follows:

$$T_C = \frac{Bc}{CIR} \qquad \text{(Eq. II-1)}$$

In some special circumstances, customers may specify $B_c = 0$ in which case T_C is set to be,

$$T_C = \frac{Be}{PortSpeed}$$

It can be seen from (Fig. II-2) that (Eq. II-1) describes the time required to transmit Bc bits if the customer transmits exactly at the CIR. In other words, Bc/T_C is the slope of the dashed line labeled "CIR" in (Fig. II-2). If the number of bits transmitted during T_C is less than or equal to Bc, the corresponding frames are not marked Discard Eligible (DE). When the number of bits is greater than Bc but less than Bc+Be, the frames are marked discard eligible but not discarded. And finally, when the number exceeds Bc+Be frames are marked discard eligible and, depending on the implementation, the frame may be discarded at this time.

A counter (denoted C) is used to keep track of how many bits have passed through the virtual connection during the measurement interval. As frames pass, the value of C is compared to Bc and Bc+Be to determine how these packets should be marked. At the end of each measurement period, the counter is decremented by the value of C or Bc, whichever is less, such that,

$$C = C\text{-}Min(C,Bc)$$

where Min denotes the minimum of C or B_C. This guarantees that in cases where C is less than Bc the counter won't be assigned a negative value. Also, this means that if the customer transmitted more than B_C bits during the interval, C will be non-zero at the beginning of the next interval so he will actually be allowed to transmit less than B_C bits until the frames are marked DE.

This overall system described above is called the "Leaky Bucket" algorithm and is a type of traffic policing mechanism. Such policing mechanisms are common in systems, including ATM, where it is necessary to penalize users when they exceed their agreed-to rate for extended periods of time. The relationship to a leaky bucket should be apparent: B_C is the leak rate of a bucket, C. The capacity of the bucket is B_C+B_e because this is the maximum value C can assume. The value of C is what is used to determine if the customer is exceeding his agreed-to rate and should be penalized by having his frames marked discard eligible.

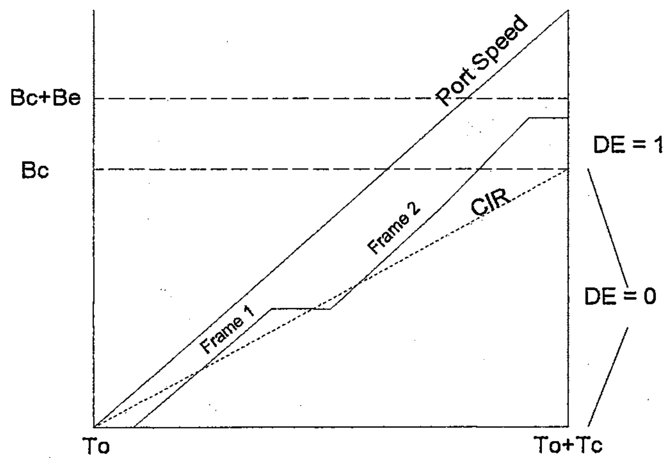

Figure II-2. The operation of the Frame Relay leaky bucket. Here, two frames are transmitted during the T_C interval. The first frame will not be tagged discard eligible, the second will.

PVCs and SVCs

Permanent Virtual Circuits (PVCs) and Switched Virtual Calls (SVCs) are used to represent the virtual connections through the network. Currently, PVCs are more commonly used than SVCs.

With PVCs, when a customer desires Frame Relay Service, he must contact the service provider and indicate what source and destination locations he wants connected together. He must also have an estimate of what his average bandwidth requirements are between each location. These bandwidth requirements are given in terms of the port speed, which is the bit rate of the access line between the customer's location and the Frame Relay network, and the committed information rate (CIR) which corresponds to the customer's estimate of how much bandwidth is needed between the two locations. The CIR can be any value from zero to the port speed and, depending on the billing arrangements, is a major determinant in the price of the circuit. Bc and Be are also specified by the customer at this time.

These parameters for the various connections are loaded into a central network administration computer and a route for each connection is chosen. From this point in time until a reroute of the traffic occurs, the information traveling over each connection will take the same path through the network. Rerouting may be performed by the service provider on demand or can occur automatically when equipment failure occurs or new circuits are added.

The less prevalent SVC is similar to the PVC except that the path setup and selection occurs in real-time. Most Frame Relay network providers don't yet support SVCs. SVCs allow the customer to connect to the network, specify the parameters and endpoints, and have the connection made on demand in a manner similar to how the telephone system works.

SVCs add more uncertainty to network sizing because it is necessary to predict how many customers will want to connect and what their bandwidth requirements will be at any given time. By contrast, with PVCs, the customer pool is a known quantity, but whether or not they are using their connections at any given time remains unknown.

Congestion control

Within the framework of the Open Systems Interconnections (OSI) 7 layer stack (See Figure II-3) Frame Relay is largely a level 2 protocol. The OSI stack is a standard which allows providers of different communications protocols to define these protocols while limiting the specification to within some bounds. The higher levels of the stack are mainly software, the lower levels define hardware and physical connections (e.g. connectors and cabling).

Congestion can occur at various levels in the OSI stack. Likewise, congestion control can occur at various levels. As mentioned previously, Frame Relay assumes, by design, that flow control occurs at higher levels. Nonetheless, both leaky bucket traffic policing and some crude congestion control are included in Frame Relay.

7. Application	
6. Presentation	
5. Session	
4.Transport	Transmission Control Protocol (TCP)
3. Network	Internet Protocol (IP)
2. Data Link	Frame Relay
1. Physical	

Figure II-3. Frame Relay's Location in the OSI Stack. TCP resides above and has its own flow control mechanisms.

The reason traffic policing and congestion control must be included is that Frame Relay cannot totally rely on the user's protocol to implement them at the higher levels. Doing so would result in situations where the users who followed the rules would be unfairly penalized because other users could abuse the system and "blast away". Furthermore, when traffic offered to a network exceeds a certain level, throughput can actually decrease. This fact arises from a combination of things:

First, the Frame Relay packet switches have limited buffer sizes and must begin discarding packets when these buffers approach the full state. As these packets are discarded, the receiving end of the customers connection must re-request the information it didn't get, further increasing the traffic offered to the network. Soon, rather than transferring new information between users, the network traffic becomes bogged down attempting to retransmit old data that keeps getting discarded. Finally, since the buffers are at a full state, the delay in the network is high which further aggravates the problem. These problems can be avoided with congestion control.

Frame Relay's congestion control is implemented by way of two additional bits included in the frame header. These bits are manipulated by the switches which collect information about how the network is performing.

1) FECN - is the Forward Explicit Congestion Notification bit. This bit is present in the header field in each Frame and is set to a one to notify the receiving end that congestion has occurred on the path.

2) BECN - is the Backward Explicit Congestion Notification bit, also present in the header field. This bit is set to notify the receiving end that it may see congestion when it transmits data.

Ideally, when the receiving end sees the FECN bit set, it should in turn tell the end transmitting to it to slow down. For this mechanism to work properly, it is assumed that the receiving end has a mechanism of controlling the rate of the transmit end.

When the receiving end sees the BECN bit set, it should slow down its transmit rate as to not further aggravate the congestion that is occurring where it is transmitting.

When Frame Relay was designed, there were some differing opinions on whether congestion actually occurred quickly or gradually. The differing opinions are reflected in the FECN and BECN bits. For FECN to work properly, congestion must occur slowly because the receiving end needs to be able to communicate with other locations to make them slow down. BECN, on the other hand, assumes that when a receiver sees BECN it slows itself and doesn't try to use the network to control other sites.

In some cases, the customer's Frame Relay equipment is not capable of passing the status of FECN and BECN up to the higher level protocol (or they don't want it to) which means the higher level protocols only slow down after frames get discarded or delayed.

Cascade's Implementation [12]

One particular vendor, Cascade, uses a scheme called "graceful discard" where the frames are not discarded unless necessary. The philosophy behind such a scheme is that in cases when the switch is not congested, there is no apparent reason why the customer shouldn't be allowed to transmit at his port speed even though he is exceeding $B_c + B_e$.

Cascade also describes their methodology of discarding these frames, depending on overall node congestion. They specify "green" frames as being those that arrive under the Bc threshold, amber as under $B_c + B_e$ and "red" as being above $B_c + B_e$. Furthermore, three states of congestion are defined depending on how full the queue is on the output buffer of the Frame

Relay switch. Their default configuration is 64 small buffers, each with a size of 512 bytes. Taken together, these 64 buffers can hold 32 kbytes which is approximately four maximum sized frames of 8192 bytes. The number of these small buffers that are full determines the congestion level of the switch:

1) Mild Congestion - 16 buffers full. Red frames are discarded. Frames transmitted are marked by setting the FECN bit. Frames received are marked by setting the BECN bit.

2) Severe Congestion - 32 buffers full. Red and amber frames are discarded. Frames are transmitted with the FECN bit set and frames received are marked by setting the BECN bit.

3) Absolute Congestion - 64 buffers full. All frames are discarded until the buffer size is down to a second threshold , "the high water mark". Then, frames are discarded based on a priority of 1 through 3 until the buffer is down further to the "low water mark". (With their system it is possible to assign priorities to PVCs.) Cascade's default low-water mark is ten buffers in the transmit queue.

Frame Relay Performance Monitoring Systems

Most Frame Relay network management systems include the capabilities to collect and store performance statistics on the switches, trunks, and PVCs. They are used to advise the network manager of the performance patterns of the network. Discrete alarms are also handled by the system to advise of service affecting failures of the equipment.

When measuring this type of performance information, the most desirable form would be to have measurements on every single frame generated by every single customer. Unfortunately, because of the sheer volume and the load put on the measurement system, this amount of detail cannot be obtained in practice.

Typically, such systems are implemented as event counters [4][12]. Separate counters are set up to measure different events on different trunks and PVCs. Examples of counter values could be: Number of Frames, Number of DE Frames, Number of Lost Frames, etc. These counter values can be automatically downloaded and stored at periodic intervals by the management system. The frequency at which these counters can be downloaded determines the time resolution of the statistics.

A Frame Relay Switch

In the figure below (Fig. II-4), the switch is connected to other switches via trunks. At this switch, there will be some PVCs both terminating and originating. Other PVCs will not terminate or originate at the switch; rather they will pass through on their way to the next switch. The main thing is that the trunks are shared and it will be necessary to find the aggregate of the several PVCs that constitute the trunk traffic. A knowledge of the routing and the network topology will be required as well as the traffic characteristics of each of the PVCs.

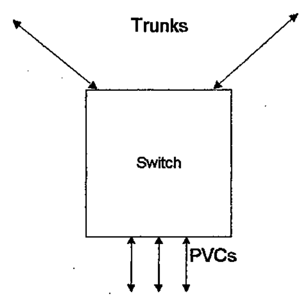

Figure II-4. A model of one switch. The switch will have incoming and outgoing trunks, and incoming and outgoing PVCs.

Initially, (Chapters III-VII) the characteristics of the traffic will be considered. In later sections, more detail will be included to represent the internal buffering performed in the switch. The next chapter will discuss traffic models for individual PVCs.

III. Traffic Models for Individual PVCs

Fig. III-1 shows the main hardware components the PVC passes through as it traverses the section of the circuit between the traffic source and output trunk of the switch. The traffic source could represent, for example, data from a LAN. First, the data is routed through a Frame Relay router, which places the information in frames. The router is connected, via an access line, to an input port on the switch. It is at the input port where the leaky bucket algorithm is applied. Finally, the frames pass on to the switching matrix where buffering and switching occur before the frames are transferred out on the trunk. This chapter focuses on the traffic source. Specifically, this chapter describes the background research and other factors that were used to determine a model for this traffic source.

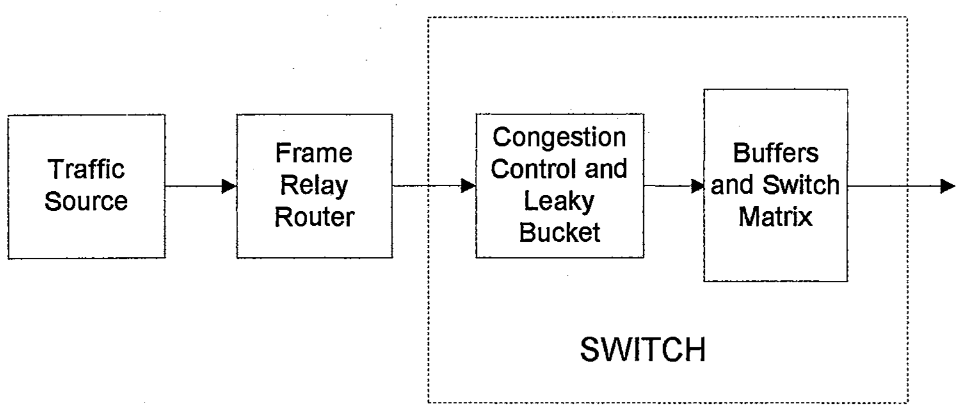

Figure III-1. The path of a PVC between the traffic source and the switch output trunk.

Overview of Some Traffic Models

Creating the model of a telecommunications network requires representing the traffic sources that provide input to the network. As will be discussed, the traffic model chosen should accurately capture the most important traits of the true message source and, at the same time, lend itself to mathematical analysis and simulation. In this research, the traffic model will represent the actual traffic on a PVC connected through the network.

One simple model for describing traffic arrivals is the Poisson model. With the Poisson model, the message interarrival time comes from an exponential distribution. This gives the model the characteristic of being "memoryless," which means the expected time until the next message arrives is unrelated to when the last message arrived. When used in conjunction with a queue where the service time for each message is also exponentially distributed, some simple mathematical results for delay can be obtained, assuming the queue has infinite capacity. These results are known as the classic M/M/1 queuing model described in [6] and [20-22].

It has generally been accepted in the literature that capturing the "burstiness" of sources is of utmost importance when modeling a source. Burstiness relates to the second order statistics of the process and is related to the autocorrelation function defined as:

$$R_{XX}(\tau) = E\{X(t)X(t+\tau)\} \qquad \text{(Eq. III-1)}$$

where X is the output of the traffic source as a function of time, τ is the lag time, and E{} denotes the expected value operator. In essence, what is important in a source model is that if a traffic source is currently generating a message, the probability that a message immediately follows must be accurately represented by the model. Failure to do so can ultimately result in under-designing the network and is a key weakness of the Poisson model. In a paper entitled "Wide Area Traffic: The Failure of Poisson Modeling" [15], and in [16] this issue is explored.

Laddada[17] used exponentially distributed interarrival times with the number of packets in an arrival coming from a geometric distribution in order to simulate the burstiness of a Frame Relay source. Models of this type fall into the category of "bulk arrival queues" named because the arrivals can occur in bulk. Such queues have been studied extensively, see [29]. The results, however, are not as nice as the M/M/1 model, particularly when the queue is of finite length.

Another technique to represent bursty arrivals is to use a Markov-modulated Poisson-process (MMPP) [45]. An MMPP consists of an underlying Markov process that jumps among various states over time. Associated with each state is an arrival rate parameter that specifies the rate for a Poisson process. A lot of the work in the MMPP field involves finding the behavior of queues when the arrivals come from an MMPP. Most of the more advanced models require numerical techniques for solutions.

Some of the work in the literature involves finding traffic models for particular types of wide-area network traffic. For example, Paxson[13] presents some empirically gathered statistical information describing the characteristics of TELNET (remote login), NNTP (news), SMTP (mail) and FTP (file transfer), all of which are protocols in the TCP suite. Another paper by Petr et al. [14] describes some broadband network simulations where the data source was chosen to mimic one of four possible classes of traffic: voice, video, image, and data. In their work, voice was modeled as a constant bit rate source, and video was modeled as a fixed length frame with exponentially distributed silence times between each frame. The image traffic was modeled as a shifted, truncated, and reversed exponential distribution with silence intervals, exponentially distributed, between each. The data traffic packet size was modeled as "bimodal" by choosing a value from two exponential distributions with different parameters. The choice of which exponential distribution supplied the size was also made randomly. The silence time between packets was again chosen to be exponentially distributed.

Two survey papers ([18] and [19]) provide an overview of some different traffic models that have been used. In these papers, traffic sources tend to be grouped by what kind of sources are being modeled: voice, video, or data.

The problem with taking one particular traffic model out of these papers and using it directly is that the exact traffic type on the Frame Relay network is unknown. Also, the model must have parameters that can be set based on the specific measurements available.

In [19], Stamoulis et al. provide a set of selection criteria that highlights the important considerations in selecting a source. Stamoulis specifically addresses ATM traffic sources, but the criteria should apply to Frame Relay as well, because Frame Relay traffic is a subset of the type of traffic slated to be carried by ATM. Using this criteria as a guide, the following are the selection criteria used in this research to select the traffic model.

1) The Type of Traffic - The traffic source should be chosen based on the type of data known to be on the network. It is known that Frame Relay carries primarily data, however, the customer base for the Frame Relay network considered in this research is a varied one. It includes: interconnected LANs, Internet providers, data terminals connected to remote mainframes, a small amount of video conferencing, and other applications such as voice over Frame Relay [30]. Knowing the exact breakdown would require a survey of each and every customer - a process that would be expensive and subject to inaccuracies. The network of the sponsor currently supports more than 20,000 PVCs, and is increasing daily. Hence, the source selected must be flexible enough to capture the important characteristics of a wide range of traffic.

2) Ease of Implementation - The source selected should be as simple as possible, but no simpler.

3) Mathematical Tractability - To be consistent with the goal of validating the results of simulations with mathematics and vice versa, the model should lend itself to formulas and approximations. This rules out some models used only in simulations.

4) Appropriateness to modeling aggregate and output traffic - When several of the sources are blended together to form the aggregate stream, does the resulting stream have characteristics similar to its real-life analogue? In this research, the leaky-bucket algorithm will be applied to the data source. The model also needs to be appropriate for this.

5) Ease of fitting to real sources - The traffic model should have parameters that can be set based on measurements of the actual traffic and the procedure for doing so should be straightforward. Furthermore, any extra parameters in the model that have no relevance to

measurements of the actual source are of no value. Hence, if the information gathered while measuring the real source is limited, the model will be inherently limited thereby suggesting the use of a simple model.

The On-Off Model

Many of the source traffic models in the literature fall into the broad category called "on-off" sources. That is, these models alternate between an "on" state and an "off" state. The variation in the different types of on-off sources comes about through the mechanisms which cause the source to switch states. For example, the work by Petr et al. [14], discussed above, uses on-off models where the on-time duration came from different distributions depending on the type of traffic the authors were interested in modeling.

It seems the on-off model, applied to telecommunications traffic modeling, first appeared in a frequently cited paper by Anick, Mitra and Sohndi [23] which appeared in the *Bell System Technical Journal* in 1982. The model they introduced is a simple two-state source that is silent when it is "off" and transmits at a fixed rate when it is "on". The source alternates between the two states and the time spent in each state, or the dwell time, comes from two independent exponential distributions which are not necessarily statistically identical. In their work, they were interested in finding the buffer length distribution for a buffer which was servicing traffic from some arbitrary number of these on-off sources with identical parameters.

In more recent works, such as [24] and [25], on-off sources with more parameters are analyzed. Having more parameters allows the sources to be more accurately adjusted to model voice traffic. These works assume that detailed measurements of the voice traffic streams are available.

The on-off model in its various levels of complexity is now well established in the telecommunications field. For this research, the on-off source with exponentially distributed on and off time will be considered first. This decision is based partially on the comments of Igelnik et. al [26]:

> The sources with exponentially distributed, on-off characteristics .... represent the simplest, most basic models of sources of

> bursty traffic. Their limitations are well known; on the other hand, such models proved extremely insightful in early studies on homogenous, bursty sources and constitute the natural starting point for new approaches for heterogeneous sources.

and similar comments in references such as [18] and [19].

Analysis of the On-Off Model with Exponentially Distributed Dwell Times

This section gives the mathematical background behind the operation of the on-off model. The bulk of the material in this section is from [27] and [28] which discuss Markov processes, a category of processes that includes the on-off model.

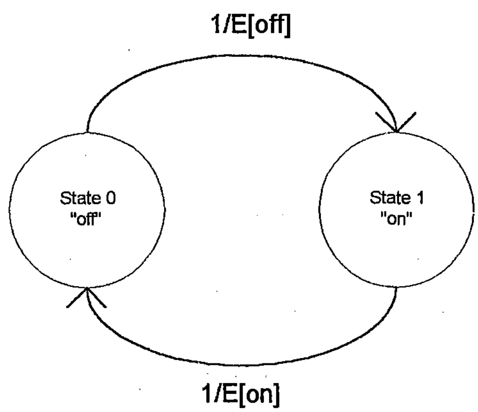

Figure III-2. The state diagram of the on-off traffic model. This model falls into a category of processes called Markov processes.

When the system is in state 0, the message source is off. It remains off for a period of time t_{off} which is a random variable from an exponential distribution with a mean value of E[off]. After t_{off} has elapsed, the message enters the "on" state where it remains for a period of t_{on} which is a random variable from another exponential distribution with mean value E[on]. Figure III-3 shows the output from an on-off traffic model.

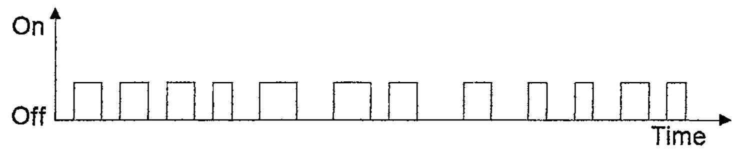

Figure III-3. The output from an on-off process. In this research, the on times are chosen from an exponential distribution, the off-times are chosen from another exponential distribution.

First, define the two additional parameters μ = 1/E[on] and λ = 1/E[off]. These are rate parameters from the exponential distributions. Also note that μdt is the probability the source will turn off when in the on state and λdt is the probability the source will turn on from the off state. The balance equations can now be written as [27],

$$\begin{aligned} P_0(t+dt) &= P_1(t)\mu dt + P_0(t)(1-\lambda dt) \\ P_1(t+dt) &= P_0(t)\lambda dt + P_1(t)(1-\mu dt) \end{aligned} \qquad \text{(Eq. III-2)}$$

which are equivalent to,

$$\begin{aligned} P_0(t+dt) - P_0(t) &= P_1(t)\mu dt - P_0(t)\lambda dt \\ P_1(t+dt) - P_1(t) &= P_0(t)\lambda dt - P_1(t)\mu dt \end{aligned} \qquad \text{(Eq. III-3)}$$

The first part of (Eq. III-2) states that the probability, as a function of time, denoted $P_0(t)$, of the source being in state 0 at time t+dt is equal to the probability that it goes from state 1 to state 0 (in time dt) plus the probability that it is in state 0 and stays there. In a similar fashion, the second equation describes the probability of being in state 1 at t+dt.

After dividing both sides of the (Eq. III-3) by dt, these differential equations can be written in matrix form in the following manner,

$$\begin{bmatrix} P'_0(t) \\ P'_1(t) \end{bmatrix} = \begin{bmatrix} -\lambda & \mu \\ \lambda & -\mu \end{bmatrix} \begin{bmatrix} P_0(t) \\ P_1(t) \end{bmatrix}$$

which is of the form,

$$\dot{P} = AP$$

having the solution of the form,

$$P(t) = e^{At} P(t_0)$$

In this model, there are only two possible states to start from: when **P**(t_0) = [0 1]' or **P**(t_0) = [1 0]'. Finding the solution for each starting case by using standard methods (such as the expansion or Laplace transform method [31]) yields,

$$\begin{aligned} P_0(t) &= \tfrac{\mu}{\mu+\lambda} + \tfrac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)(t-t_0)} \\ P_1(t) &= \tfrac{\lambda}{\lambda+\mu}\left(1 - e^{-(\lambda+\mu)(t-t_0)}\right) \end{aligned} \qquad \text{(Eq. III-4)}$$

for **P**(t_0) = [1 0]' and ,

$$\begin{aligned} P_0(t) &= \tfrac{\mu}{\lambda+\mu}\left(1 - e^{-(\lambda+\mu)(t-t_0)}\right) \\ P_1(t) &= \tfrac{\lambda}{\mu+\lambda} + \tfrac{\mu}{\lambda+\mu} e^{-(\lambda+\mu)(t-t_0)} \end{aligned} \qquad \text{(Eq. III-5)}$$

for **P**(t_0) = [0 1]'.

If n_0 is designated as the starting state at t_0, then n_0 can assume either 0 or 1. This allows (eq. III-4 and III-5) to be combined as,

$$P_0(t|n_0,t_0) = \tfrac{\mu}{\mu+\lambda} + (\delta(n_0) - \tfrac{\lambda}{\mu+\lambda})e^{-(\lambda+\mu)(t-t_0)}$$
$$P_1(t|n_0,t_0) = \tfrac{\lambda}{\mu+\lambda} + (\delta(n_0 - 1) - \tfrac{\mu}{\mu+\lambda})e^{-(\lambda+\mu)(t-t_0)}$$

where,

$\delta(n_0) = 1$ for $n_0=0$, and 0 for $n_0 \neq 0$.

Now U, defined to be the steady state probability of finding the source in the on state, can be written,

$$U = \lim_{(t-t_0)\to\infty} P_1(t|n_0,t_0) = \frac{\lambda}{\lambda+\mu}$$

or equivalently,

$$U = \frac{E[on]}{E[on]+E[off]} \qquad \text{(Eq. III-6)}$$

U is also the "utilization" of the source, because it is the percentage of time the source is in the on state.

In the next chapter, the type of measurements that are available from the switch will be discussed. The possibility of using these measurements to set λ and μ will be explored in Chapter V and more details on the mathematical behavior of this on-off process will be introduced as necessary, using the equations shown here.

IV. Obtaining the Actual Measurements

This chapter describes the process used to obtain the actual PVC traffic measurements from the network. These measurements will be used to set the parameters in the PVC traffic model. Also, included in this chapter will be a discussion of how the measurements can be used to make insights into the behavior of individual customers. Finally, other notable trends that appear in the data will be addressed.

Background

As briefly discussed in Chapter II, most Frame Relay systems include a Performance Monitoring (PM) platform as part of the management system. The switches collect the measurements, and the central management system periodically downloads, interprets, and displays the measurements. This type of operation is not unique to Frame Relay equipment, but applies to routers, channel banks, phone systems and other equipment as well. In fact, standards exist [4] to guide both the equipment manufacturers and the management platform suppliers in an effort to increase interoperability between different types of equipment and management systems.

The Management Information Base (MIB) for network equipment is a collection of parameters held within the equipment itself. Some of the parameters can be modified by the management system. This might occur, for example, when a new circuit is being installed. Other parameters, can be downloaded by the management system. These parameters include information about how the equipment is performing and statistics describing the traffic flowing through the equipment. The MIB for Frame Relay has been specified by committees and working groups and is defined in "Request for Comments" (RFC) 1604.

The design of management platforms, responsible for interacting with the MIB, is also guided by standards committees. One of the commonly used protocols, which came out of these committees, is the Simple Network Management Protocol (SNMP) [4]. SNMP is the basis for the management system used to collect performance information in this research.

The MIB Values

There are fourteen parameters (seven in each direction - "in" and "out") from the Frame Relay MIB which are of interest for this research. A description of these parameters is in Table IV-1.

Parameter Name	Description
frPVCEndptInBc frPVCEndptOutBc	Committed Burst setting on the PVC
frPVCEndptInBe frPVCEndptOutBe	Excess Burst setting on the PVC
frPVCEndptInCIR frPVCEndptOutCIR	Committed Information Rate on the PVC
frPVCEndptInFrames frPVCEndptOutFrames	The number of frames on the PVC since the last initialization
frPVCEndptInOctets frPVCEndptOutOctets	The number of Octets (bytes) on the PVC since the last initialization
frPVCEndptInDEFrames frPVCEndptOutDEFrames	The number of Discard Eligible Frames since the last initialization
frPVCEndptInDEOctets frPVCEndptOutDEOctets	The number of Discard Eligible Octets (bytes) on the PVC since the last initialization

Table IV-1. Parameters from the Frame Relay MIB used in this research.

The MIB parameters that measure the number of bytes and frames are actually counter values. They reflect the number of bytes or frames that have passed through the circuit since the last initialization. In the equipment used in the research these counters are 32-bit unsigned integers which can assume a maximum value of approximately 4 billion before they "roll-over."

When the management program extracts MIB information from the switches, it must do so at fixed intervals and then subtract the current values from the previously extracted readings in order to find the number that occurred during the interval. If the polling interval is too large, the counters can roll over multiple times making it impossible to figure out the true count during the interval.

The polling interval should be small enough to capture the fluctuations, over time, of the measured value of interest. Ideally, the counters would be polled as fast as possible, but this is not practical for two reasons. First, the amount of storage space required for this data becomes unrealizable. For example, in this research, each PVC (there are roughly 20,000) is polled 96 times per day (every 15 minutes). This requires roughly 400 Mbytes to store in ASCII format. Second, the polling process places an extra load on the Frame Relay switch, degrading its performance. For these reasons, 15 minutes was chosen as the sampling interval and will be used throughout this research.

Interpreting MIB Values

Some basic calculations can be made using the values obtained from the MIB. (These examples assume a 15 minute polling interval.) It is assumed that the port speed, CIR, Bc, and Be are all known before performing these calculations and that the readings from the MIB have been differenced from the previously polled readings to get the number within the 15 minute interval.

Using the number of frames and the number of bytes during the interval, the average frame size, E[FrameSize], can be computed as,

$$E[FrameSize] = \frac{\#bytes \times 8}{\#frames} \qquad \text{(Eq. IV-1)}$$

which gives the average size in bits/frame.

Next, the average size of a discarded frame can be computed as,

$$E[DEFrameSize] = \frac{\#DEbytes \times 8}{\#DEframes} \qquad \text{(Eq. IV-2)}$$

again, in bits/frame.

The average utilization can be computed by knowing the port speed of the circuit and the total number of bytes.

$$E[Utilization] = \frac{\dfrac{bytes \times 8}{15\min \times 60\sec/\min}}{portSpeed} = \frac{bytes \times 8}{portSpeed \times 900} \quad \text{(Eq. IV-3)}$$

where the port speed is given in bits/sec.

Finally, the percentage of total bytes and total frames that are discard eligible can be computed from,

$$\%DEbytes = \frac{DEbytes}{bytes} \times 100 \quad \text{(Eq. IV-4)}$$

$$\%DEframes = \frac{DEframes}{frames} \times 100 \quad \text{(Eq. IV-5)}$$

Some Example Measurements

As an example, consider this actual data collected for a single PVC. This customer had a port speed of 64kbits/s and a value of 32kbits for both B_C and B_e.

The measurements obtained for the 15 minute interval were as follows:

#Frames = 7198
#Bytes = 1,368,214
#DE Frames = 492
#DE Bytes = 154,391

Using equation (IV-1 through IV-5) the following values can be obtained,

E[FrameSize] = 190 bytes
E[DEframeSize] = 313 bytes
E[Utilization] = 19%
%DEbytes = 11%
%DEframes = 7%

The results for this example will also be used in the following chapter where the model will be fit to these measurements.

The results for this example will also be used in the following chapter where the model will be fit to these measurements.

V. Fitting the Model

This chapter will discuss the procedure used to fit the on-off traffic model to the actual statistics obtained from the switch. It will be seen that some modifications will need to be made to the original two-state model, proposed in Chapter III, to obtain an adequate fit.

Overview of Leaky Bucket Simulation

A leaky bucket simulation program was written for this research. This simulation program makes it possible to study the behavior of the leaky bucket in a controlled environment which allows for creating traffic patterns, and then seeing how the Leaky Bucket responds to these traffic patterns. Understanding the relationship between artificially generated traffic patterns and the corresponding output from the leaky bucket simulation provides great insight to what characteristics the real traffic sources must have to produce the observed measurements.

The leaky bucket (LB) simulation consists of three stages: the message generation stage, the framing stage, and the leaky bucket stage. (See Figure V-1)

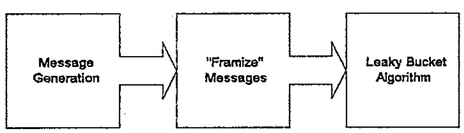

Figure V-1. The three stages of the "leaky bucket simulation."

The message generation stage creates a stream of messages based on the two state Markov process described in Chapter III. The parameters for the two exponential distributions, λ and μ as well as a duration are specified by the user prior to starting. In this research, a duration of 900 seconds was used because this corresponds to the 15 minute polling interval used by the management system.

Once this message stream is created, the framing stage converts these messages into frames by considering the port speed, and the maximum frame size.

Finally, each frame is tagged as being one of three possibilities: discarded, discard eligible (DE), or not DE based on the Bc, Be, and CIR specified by the user.

After running the simulation, the user is presented with the following statistics:

- Number of messages created
- Number of frames created
- Average frame size
- Variance of frame size
- % discarded frames
- % DE frames
- Utilization

Results From Initial On-Off Model

Recall the PVC statistics for the example introduced in Chapter IV. Of those parameters, listed below are the ones that are candidates to be fit to the output of the simulation program.

#Frames = 7198
E[FrameSize] = 190 bytes
Utilization = 19%
%DEframes = 7%

To check the usefulness of the on-off model introduced in Chapter III, an attempt will be made to find the two parameters, λ and μ, for the on-off model which will produce the same results as above.

It should first be observed that the model has only two adjustable parameters, yet there are three independent target measurements which it must fit. At first glance, there appear to be four, but only two out of the three of #Frames, E[FrameSize], or Utilization are independent. Given two of them, you can calculate the third (when the port speed is known). This alone suggests that the model may be overly simple. As an aside, note that the E[FrameSize] value of 190 bytes is relatively small compared to the maximum frame size of 8192 bytes allowed by the Frame Relay protocol. This indicates that most of the messages do not require multiple frames to transport them.

Proposed Modifications

The model must be modified in such a manner that these two parameters, %DEframes and E[FrameSize] still fit, but the average utilization and number of frames over the 15 minute period will also match what is measured.

Noting that the average utilization and number of frames are too large by a similar factor, one way to do this is to occasionally insert silence periods that are, on average, longer than the expected off-time from the exponential distribution originally specified. This can be implemented by using a hyperexponential distribution for the off-time distribution rather than a single exponential distribution, as in the initial model. The hyperexponential distribution is defined as,

$$f_X(x) = p\lambda_{shortOff}e^{-\lambda_{shortOff}x} + (1-p)\lambda_{longOff}e^{-\lambda_{longOff}x}$$

where p is the probability that the random variable X comes from the distribution with parameter $\lambda_{shortOff}$ and (1-p) is the probability that it comes from the distribution with $\lambda_{longOff}$.

Other researchers [52] have studied the three-state model with hyperexponential distribution, but in conjunction with the on state rather than the off state. Although they never collected any real measurements in their research, the authors proposed finding the first three moments of the on-state empirically and then using these to set p, and the two exponential PDF parameters. As mentioned previously in this paper, obtaining useful measurements is often the most difficult part of the application of a traffic model and unfortunately the detail required in [52] is not available from the Frame Relay switches. In this research, real measurements must be used to set the model parameters and it will be shown that, given the capabilities of the measurement system, the model with a hyper-exponentially distributed off-time proposed here makes this process possible. No known method exists for fitting the dual on-state model to the Frame Relay measurements available in this case. Consequently, the hyperexponential

Two of the four measurements must be chosen as primary targets for the fitting. The choice for the first target measurement is straightforward: %DEframes is important because the percentage of frames that are discarded strongly affects the performance seen by the customer as the switch buffers become loaded. The choice made, initially, for the second target measurement was utilization, but this turned out to be a bad choice for reasons that will be pointed out later. With the simple on-off model, the average frame size required for the simulation to produce a %DEframes=7% was much greater than the 190 bytes observed in the measurements. Consequently, the number of frames created during the 15 minute run was much smaller than the 7198 observed in the actual measurements.

The next thing tried was fitting %DEframes and E[frameSize] while letting the utilization and #Frames fall where they may. Running the simulation by trial and error finally yielded, with E[on] = $1/\mu$=0.02375 (s) and E[off] = $1/\lambda$ = 0.0290 (s), the results:

%DEframes = 7% ($\pm$ 3% depending on the seed)
#Frames = 16,863 frames
E[frameSize] = 1521 bits (198 bytes)
Utilization = 46 %

After this fit, the %DEframes and E[frameSize] are relatively close to the actual measurements, but the utilization is higher than measured by a factor of 2.4 and the number of frames (#Frames) is higher by a factor of 2.3. The fact that these two numbers are similar is no surprise: the amount of time the source model is in the "on" state is proportional to the number of frames generated by the source model.

Based on the results illustrated by this and other examples, the two-state source model is unable to generate simulated results that match the observed statistics. Hence, it must be ruled out as the definitive source model for this research.

It should be noted that the example measurements used for illustration are typical of other PVC measurements observed in the data. The common characteristic of all these different PVC measurements is a situation where the simulated two-state source model necessary to produce the measured %DE and E[FrameSize] yields an overall utilization and number of frames higher than the values actually measured.

distribution will be used for the off-time distribution. The state diagram for this model can be seen in (Fig. V-2).

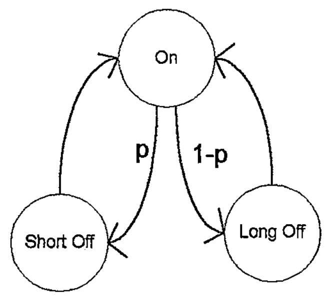

Figure V-2. State diagram of the three-state traffic model.

If $\lambda_{shortOff}$ and $\lambda_{longOff}$ are specified to be the parameters for the short and long off time distributions respectively, then it can be seen that the probability of drawing n consecutive numbers from the $\lambda_{shortOff}$ distribution is proportional to p^n. Seen another way, if p >> 1-p, there will be "bursts" of multiple on - shortOff (shortOff will be used to describe an off time from the distribution with $\lambda_{shortOff}$ parameter and E[shortOff] will be used as the average of these short off times) transitions where the number of these transitions occurring consecutively is geometrically distributed. Separating these bursts will be off times from the $\lambda_{longOff}$ distribution. (See Fig. V-3.)

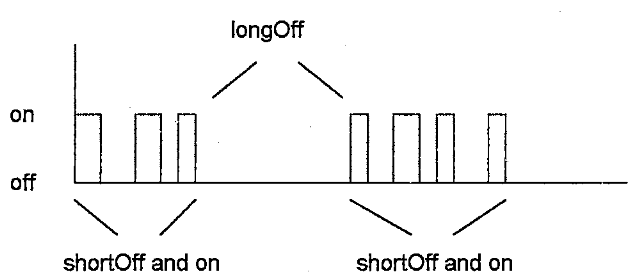

Figure V-3. Output from the three state on-off traffic model. The "bursts" of on-shortOff cycles are separated by pauses from the longOff distribution.

The $\lambda_{shortOff}$ parameter will now be chosen to be equal to the λ parameter in the original on-off model and the value of the μ parameter in the original model will be retained. Assuming this, the utilization *within the bursts* will be the same as for the original on-off model given by (Eq. III-6).

The expected burst length, can be found by using the geometric distribution,

$$G(x) = (p)^{x-1}(1-p) \quad \text{where } x = 1,2,...$$

which describes the PDF of the required number of trials until an event with probability (1-p) occurs. The expectation for the geometric distribution is given as [46],

$$E[x] = \frac{1}{1-p} \qquad \text{(Eq. V-1).}$$

In this case, the event that occurs at the end of the burst is the long off time from the $\lambda_{longOff}$ distribution which occurs with probability 1-p, so the expected number of cycles in the burst is (from Eq. V-1) 1/(1-p) and the expected burst length is,

$$E[burstLength] = (\frac{1}{\mu} + \frac{1}{\lambda_{shortOff}})(\frac{1}{1-p}) \qquad \text{(Eq. V-2)}$$

where the first term in (Eq. V-2) denotes the expected time of one cycle and the second term is the expected number of cycles.

The choice of p and $\lambda_{longOff}$ is still an open issue. Recall that $\lambda_{shortOff}$ has been chosen to be λ in the original on-off model which, with μ, resulted in a utilization which will now be denoted $U_{bursting}$. This utilization was greater than the actual measured utilization (denoted $U_{desired}$ in the following). By introducing three additional new terms: $time_{on}$, $time_{shortOff}$ and $time_{longOff}$, which represent the total time during the 15 minute interval that the model is in each of the three states it is apparent that,

$$time_{on} + time_{shortOff} + time_{longOff} = 15 \text{ minutes}$$

and,

$$time_{bursting} = time_{on} + time_{shortOff} .$$

So clearly,

$$time_{bursting} + time_{longOff} = 15 \text{ minutes} \qquad \text{(Eq. V-3)}$$

which simply states that the model is always in one of the three states: on, longOff, or shortOff. The total amount of time the model spends in the bursting state can be determined by first computing the total time it should be in the on state,

$$time_{on} = (15 \text{ minutes})(U_{desired}) \qquad \text{(Eq. V-4)}$$

and, noting that the only times it can be in the on state is when it is in the bursting state,

$$time_{on} = (time_{bursting})(U_{bursting}) \qquad \text{(Eq. V-5)}$$

Solving (Eqs. V-4 and V-5) yields,

$$time_{bursting} = (15 \text{ minutes})\ (U_{desired}/U_{bursting}) \qquad \text{(Eq. V-6)}$$

and with (Eq. V-3),

$time_{longOff}$ = (15 minutes) ($1-U_{desired}/U_{bursting}$) (Eq. V-7)

The parameters $\lambda_{longOff}$ and p are related: choosing one, dictates the other. To see that this is true, consider (Eq. V-2) which states that as p increases, E[BurstLength] also increases thereby implying there must be fewer bursts to meet the $time_{bursting}$ requirement. Furthermore, fewer bursts during the 15 minutes implies fewer transitions to the longOff state indicating the $time_{longOff}$ requirement must be met by the sum of fewer variates from the $\lambda_{longOff}$ distribution. In effect, a larger p requires a larger E[longOff]. This can be seen mathematically as,

E[number of bursts] = E[number of draws from longOff distribution]

so,

$$\frac{time_{bursting}}{E[burstLength]} = \frac{time_{longOff}}{E[longOff]} \qquad \text{(Eq. V-8)}$$

and by substituting (Eq. V-2) for E[burstLength] it can be seen that,

$$\frac{time_{bursting}}{(\frac{1}{\mu}+\frac{1}{\lambda_{shortOff}})(\frac{1}{1-p})} = time_{longOff}\lambda_{longOff} \qquad \text{(Eq. V-9)}$$

then solving for p yields,

$$p = 1 - \frac{time_{longOff}}{time_{bursting}}\lambda_{longOff}(\frac{1}{\mu}+\frac{1}{\lambda_{shortOff}}). \qquad \text{(Eq. V-10)}$$

The use of these equations will be illustrated via the example below.

Setting the Parameters in the New Model

Now, the model contains four parameters: μ, $\lambda_{shortOff}$, $\lambda_{longOff}$, and p. The goal is to choose these parameters in such a way that *a traffic source model* that matches the measured statistics is produced. Because the amount of information contained in the measurements is limited, there are many possible model configurations and parameter choices that could have been used to match these measurements. Some of the steps that follow may seem somewhat ad-hoc, but are consistent with the goal of finding a possible traffic source (as opposed to the definitive source). In Chapter VII, the characteristics of the model will be compared with more detailed measurements of actual traffic and also with some more information in the literature. The comparison will show that the traffic model developed in this chapter is reasonable and captures important traits of actual PVC traffic.

From the discussion in the previous section, μ and $\lambda_{shortOff}$ are chosen to be the same as the λ and μ parameters in the original two state model after it has been fit to match E[FrameSize] and %DEframes. This leaves p and $\lambda_{longOff}$ to be chosen. It has been determined experimentally that E[burstLength] (a function of p, $\lambda_{shortOff}$ and μ) must be several times larger than T_C, the leaky bucket window size, in order for frames to be marked DE -- a factor of at least 5 times T_C yields a %DE approximately equal to what is obtained by fitting the two state model. Choosing too small of a p results in an E[burstLength] that is small relative to T_C and frames do not get marked DE with the same percentage they did after fitting μ and $\lambda_{shortOff}$ using the two state model. Along with this experimentally determined constraint on E[burstLength], (Eq. V-8) can be used to determine the constraint on $\lambda_{longOff}$ as,

$$\frac{time_{bursting}}{time_{longOff}\lambda_{longOff}} \geq 5T_C$$

so,

$$E[longOff] = \frac{1}{\lambda_{longOff}} \geq 5 \cdot T_C \frac{time_{longOff}}{time_{bursting}} \qquad \text{(Eq. V-11)}$$

The effect of choosing various values for E[longOff] on the probability of dropping traffic is further investigated at the end of this chapter.

Returning to the example measurements first introduced in Chapter IV and used throughout this chapter, $U_{bursting}$ was determined from the two state model to be 46.5%. The $U_{desired}$ from the actual measurements is 19%.

Using (Eq. V-7) for $time_{longOff}$, the total time spent in the longOff state during the 15 minute interval can be obtained as,

$$time_{longOff} = (900\,\text{sec})(1 - \frac{.19}{.46}) = 528\,\text{sec}.$$

The time spent in the bursting state, $time_{bursting}$, will be the remaining time. Using (Eq. V-6) or (Eq. V-3),

$$time_{bursting} = (900\,\text{sec})(\frac{0.19}{0.46}) = 372\,\text{sec}.$$

The relationship between the E[burstLength] and E[longOff] is linear (see Fig. V-4) and so the choice made for E[longOff] directly determines E[burstLength]. In (Fig. V-5) the relationship between p and E[longOff] is shown for the example. In this figure it can be seen that choosing a very large E[longOff] requires a p very close to unity in order to match the $U_{desired}$ parameter. Given the constraining equation for E[longOff], (Eq. V-11), a value for E[longOff] is chosen as 10.0 (sec) which via (Eq. V-8) results in an E[burstLength] of 7.69 (sec), more than 5 times T_C.

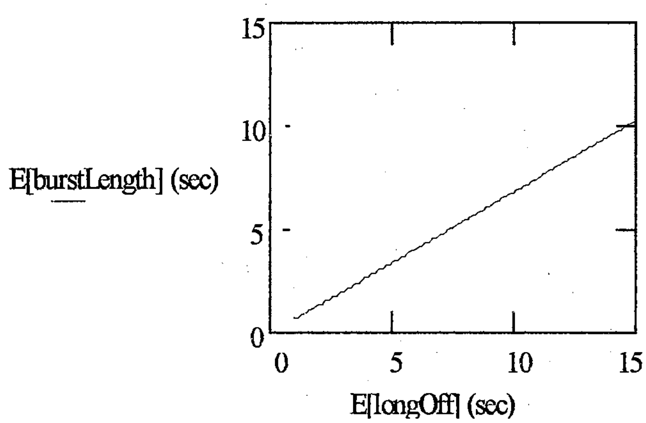

Figure V-4. Relationship between E[longOff] and the E[burstLength] (Eq. V-8) for this example.

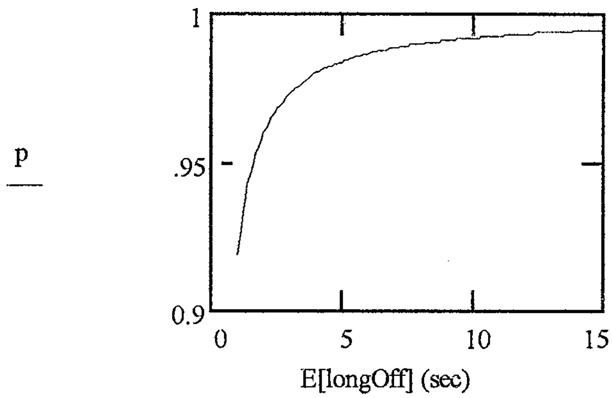

Figure V-5. Relationship between the E[longOff] parameter and p (Eq. V-10) when $U_{desired}$ is fixed.

Via equation (Eq. V-10), p can be found as,

$$p = 1 - \frac{528}{372}(0.1)(0.025 + 0.0285) = 0.993$$

which means the longOff distribution in the hyperexponential distribution should be chosen 0.7% of the time.

Results from three-state On-Off Model

Now that the new PVC model has been developed, it will be checked against the example PVC measurements to see if it fits better than the two state model. This was done by modifying the message generation state in the simulation program to generate messages based on the three-state model.

The four parameters are:

duration: 15 minutes
μ = 1 / 0.02375 = 42.1 (from two-state fit)
$\lambda_{shortOff}$=1 / 0.0290 = 34.5 (from two-state fit)
$\lambda_{longOff}$ = 1/ 10.0 = 0.1 (chosen)
p = 0.993 (calculated)

Results from simulation:

%DEframes = 5%	(vs. 7% measured)
#Frames = 8118 frames	(vs. 7198 measured)
E[frameSize] = 1542 bits	(vs. 1520 measured)
Utilization = 19.2%	(vs. 19% measured)

It is evident that this three-state model does a much better job than the two-state model matching the actual measurements. Since this is a statistical simulation, the results vary slightly depending on the seed, but are generally with 5-10% of the measured values. This method appears to work equally well fitting measurements from other PVCs measurements taken from the data.

Effect of choosing different E[longOff] values on P(dropped packet)

An experiment was performed to further investigate the effect of choosing various values for E[longOff] on the probability of a dropped packet. Previously, it was found experimentally that E[longOff] should be chosen such that E[burstLength] > 5Tc in order for the model to be properly fit to the statistics. There is, however, an open question of how this choice affects the probability of data loss. Does the choice of E[longOff] have a significant effect on the probability of a dropped packet? A simulation was performed to help resolve this question.

In this experiment, 4 sources were sharing a 64 kbit/sec trunk. The E[on] for the sources was set at 0.02375 seconds (190 bytes on a 64kbps circuit) and the trunk utilization was held relatively constant between 50 and 60 percent. The trunk had a buffer size of 64 kbits and the simulation was run for 900 seconds on CACI Comnet. From the values shown in Table V-1 it can be seen that choosing an E[longOff] of more than 5 to 10 times greater than T_C does not have a significant impact on p(Dropped Packet) or p(discard eligible). This should alleviate concerns that the choice of E[longOff] is critical to the probability of a dropped packet.

Further supporting the choice of approximately 5 times T_C for the value of E[longOff] is the fact that a value smaller than this causes the probabilities of both DE and dropped traffic to begin falling to lower values than were observed in the measurements, implying that the model is no longer capturing the traits of the true traffic. As for evidence that the correct choice for E[longOff] is not much greater than 60 time T_C seconds is the fact that the actual peak-hour traffic measurements exhibit fairly consistent behavior from one 15 minute interval to the next. If the correct value for E[longOff] was much longer than 60 times T_C seconds than this wouldn't be true. The traffic measurements during the busy hour would vary considerably between adjacent 15 minute intervals.

In summary, when the range of E[longOff] is between 10 and 60 times T_C, the exact choice made within this range doesn't seem to affect the two critical parameters, p(dropped packet) and p(DE). Once this the value is chosen, the other three model parameters -- p, E[shortOff], and E[on], are fixed.

E[longOff]	Trunk Utilization	prob(DE)	prob(dropped packet)
60	51%	0.1214	0.03629
31	54%	0.1446	0.04303
25	56%	0.1433	0.04380
20	57%	0.1238	0.03215
16	58%	0.1253	0.02654
12	58%	0.1127	0.02633
8	58%	0.0976	0.01331
4	58%	0.0727	0.01002
mixed*	52%	0.1419	0.04260

the mixed case had one each of 12,20,25, and 60 second E[longOff] sources

Table V-1. The effect of varying E[longOff] on the prob(DE) and prob(dropped packet) for T_C = 1.0 s.

Discussion

This chapter illustrated how the three-state model developed here works much better than the two-state model in matching the measured statistics. The procedure of fitting the three-state model involves first fitting the two-state model to the measured E[frameSize] and %DE and then adjusting the two remaining parameters using mathematics. That is, the method requires "trial and error" to initially fit the two state model which gives the appropriate values for $\lambda_{shortOff}$ and μ. Fitting the other two parameters, p and $\lambda_{longOff}$, is done with the equations given here with the constraint that E[burstLength] must be greater than 5 times T_C. This has been performed on a number of circuits and performs satisfactorily.

The three-traffic model developed in this chapter is actually a three state Markov model as shown in Figure V-2. Normal Markov techniques could have been used to analyze the model, but since it was assumed that p >> 1-p and E[longOff] >> E[shortOff] the analysis could be simplified considerably and other parameters such as E[burstLength] could be more easily developed.

The following chapter (Chapter VI.) will contain a more detailed discussion of the simulation program that was used to fit the two-state and three state cases. The primary reason is to develop an approximation to the simulation so that the two-state model doesn't need to be fit by "trial and error." Remaining chapters will discuss the properties of the aggregate stream consisting of several of the three-state traffic models and how this aggregate can be obtained by knowing the routing of the PVCs through the network. Finally, the question of user quality of service will be addressed. During all this, evidence supporting the assumptions made here will be included.

VI. Leaky Bucket Traffic Policer

This chapter gives more details on the leaky bucket traffic policer and the simulation program introduced in Chapter V. Also, an approximation to the simulation will be developed here to allow rapid estimation of the four parameters for the three-state traffic model. It is necessary to develop this approximation because the model parameters for all the PVCs on the network must be obtained -- an operation that would be too time consuming using the simulation. Finally, the performance of this approximation is evaluated against the simulation.

Simulation of Leaky Bucket Algorithm

The traffic measurements from the network were collected in 15 minute intervals. These measurements consisted of %DE, #Frames, E[FrameSize] and utilization.

Recall that in Chapter V the two parameters, $\lambda_{shortOff}$ and μ, were obtained by fitting a two state traffic model to the %DE and E[FrameSize] through the use of the simulation program. This process involved "trial and error." Once this was done, the remaining two parameters, $\lambda_{longOff}$ and p, were set mathematically in conjunction with the measured utilization and #Frames. As a result, this chapter will focus on the two state model because once the two parameters for this model are obtained, the process of finding the remaining two parameters in the three state model is straightforward.

The general steps in the simulation are as follows:

1) Generate 15 minutes worth of on-off messages based on $\lambda_{shortOff}$ and μ.
2) Put these messages into frames.
3) Process these frames using the "leaky bucket" algorithm to tag them as non-DE, DE, or discarded and then display the results.

The details in the steps are as below:

1) A message array was created, each element of the array was a message structure which contained a t_{on}, and a t_{off} parameter. The duration, t_{off} minus t_{on}, for the message was a random

number chosen from a generator with an exponential distribution with a mean of $1/\mu$. The next message in the array had its t_{on} set by taking the t_{off} of the previous message plus a random number chosen from an exponential random number generator with a mean of $1/\lambda_{shortOff}$. In this manner, message structures were added to the array until the last t_{off} was greater than 15 minutes (900 seconds). Because of the randomness of the message sizes, the exact number of elements in the array could not be known beforehand so the array was grown dynamically.

2) A frame array was created, each element of this array was a frame structure. The frame structures have an t_{on} and t_{off} parameter in addition to a size parameter which indicates how many bits the frame contains. They also have flags, which will be set in step three, to indicate whether the frame is non-DE, DE, or discarded. The frame array was filled by processing through the message array and "frameizing" the messages. In the cases where an entire message would not fit into a frame, multiple frames were created at the maximum size followed by one smaller frame holding the remainder of the bits. The number of bits in a frame is the product of the duration of the frame and the port speed. The number of frames created is at least the number of messages plus extra frames in cases where a message fills multiple frames.

3) The frame array was then processed to see which frames exceed B_c or B_c+B_e during the time interval T_c. This procedure iterated through the frame array keeping track of which T_c interval the bits occurred in and flagged frames based on the content of the leaky bucket.

The frame array was then processed to figure out its statistical characteristics. The number of non-DE, DE, and discarded frames was computed and percentages of DE frames and discarded frames were calculated. Also, the average frame size and the utilization were determined.

An Approximation to a Simulation

There are two drawbacks to using this simulation. First, it is a simulation which is computationally expensive. For example, generating 15 minutes worth of messages with an E[on] of 0.01 seconds and E[shortOff] of 0.04 seconds requires generating an average of 900/(0.01+0.04), or 18,000 messages. (This assumes the two-state model.) Second, it provides the measured statistics after being given the E[on] and E[shortOff] which is not what is actually needed. The reverse is actually desired: to obtain the E[on] and E[shortOff] from the measured statistics. But, having this would require placing the simulation program inside a minimization routine and solving iteratively, a procedure that would be extremely slow when applied to measurements from multiple PVCs.

To get around the computational expense of the simulation, it is necessary to have a numerical approximation to the simulation. This numerical approximation will then be placed inside a minimization routine to rapidly obtain the model parameters from the observed measurements. The development of this approximation will now be described.

In the queuing theory literature, there is some interest in the leaky bucket because it can be modeled by a G/D/1/K queue [38,47]. In other words, it can be modeled as a single server queue with maximum queue length K; the arrivals are from a general (arbitrary) distribution and the service rate is a constant, D. Queues of this form are notoriously difficult to solve and are done numerically. The normal approach taken involves writing equations for every state the queue can exist in (there is a finite number, K) and then finding the steady state probabilities of each of these states.

In the case here, solving by modeling as a queue would be very difficult for the following reasons: First, K is Bc+Be which is generally on the order of 64,000 bits or more. Also, the distribution for the message arrival rate is not a straightforward Poisson distribution and the frame arrivals are not independent. They are not independent because when a frame comes in, there is a chance another one will immediately follow if it is part of a larger message. Also, it will be shown that the probability density function (PDF) for the frame size is not straightforward.

An approximation to the leaky bucket that relies on random variables rather than queuing theory and works for the case of Frame Relay with a two-state on-off input will be developed. Its final merits will be judged by comparison with the results of the simulation. The basic approach will be to consider the PDF for the number of bits that arrive during a given time interval. Using this PDF along with consideration of the leaky bucket operation, it will be possible to estimate the PDF for how many bits are in the cumulative bit counter when a frame arrives. This PDF will then describe the probability of the frame being marked either non-DE, DE, or discarded. The PDF for the size of the frame must also be considered because bits in the frame are added to the cumulative bit counter before the frame marking decision is made.

To start the development, Let C(t,i) be a random process that describes the value of the cumulative bit counter during the ith time interval T_c. C(t,i) is a function of the bits carried in from the previous T_c , denoted L(i), plus the bits that have arrived during the current T_c which will be denoted M(t,i). Note that L(i) is not a function of time, however M(t,i) is because its statistics change depending where in the T_c interval it is observed. The i denotes which T_c window is being considered. The basic equation can be written:

$$C(t,i) = M(t,i) + L(i) \qquad \text{(Eq. VI-1)}$$

which says that the value of the cumulative bit counter is a function of the number of bits carried in from the last interval plus the bits that have arrived so far during this interval.

Characteristics of the Random Variables C(t,i), L(i), and FrameSize

Based on the operation of the leaky bucket algorithm, consider the limits imposed on these random variables.

C(t,i) -- Limited to [0,Bc+Be]

L(i) -- Limited to [0,Be]

It can also be observed that,

$L(i) = \max(C(T_c,i-1) - Bc, 0)$ (Eq. VI-2)

which comes from the fact that at the end of a T_c interval, Bc bits are subtracted, or "leaked" from the cumulative bit counter, but it is never allowed to become negative. Since Bc+Be is the maximum value allowed for C(t,i), this explains why L(i) is limited to [0,Be].

An equation for $C(T_c,i-1)$ can also be written as,

$C(T_c,i-1) = \min(Bc+Be, L(i-1) + M(T_c, i-1))$ (Eq. VI-3)

which says that at the end of the T_C interval, the value of $C(T_C,i-1)$ is limited to Bc+Be. Also observe that (Eq. VI-3) can be substituted into (Eq. IV-2) which will produce an equation for L(i) which is a function of both L(i-1) and $M(T_C,i-1)$. In fact, continuing to substitute these two equations while delaying the i's would show that L(i) is a function of what has arrived during all the previous T_C intervals.

Whether a frame is marked DE or discarded is directly related to the probability that the trailing edge of a given frame, arriving at time t, finds one of the following three cases:

1) $C(t,i) < Bc$: The frame will be non-DE,
2) $Bc \leq C(t,i)$: The frame will be marked DE,
3) $Bc+Be \leq C(t,i)$: The frame will typically be discarded.

In order to exactly find the PDF for C(t,i), it would be necessary to find the PDF for L(i) and M(t,i) and then map them through a function with nonlinearities introduced by the max and min operations. This would be very difficult so further simplifications will be explored starting with a closer examination of the PDFs of C(t,i), M(t,i), L(i) and the frame size. This closer examination will reveal that exact PDFs for these four random variables cannot be obtained. However it will be possible to determine estimates for their mean and variance and make some assumptions about the shape of their respective distributions. It will then be possible to combine this information into a form that allows estimating the probabilities of frames being marked discard eligible for a given set of model and PVC parameters.

PDF of M(t,i)

The first thing to be investigated is the distribution for M(t,i). Recall that the input source is a two-state Markov process. Let X(t) denote the output of the on-off model which is either 1 for "on" or 0 for "off". Also, let the random variable S(t) be related to M(t,i) as: M(t,i) = portSpeed x S(t). (S(t) will be defined below) The units of S(t) are seconds, the units of M(t,i) are bits and the port speed units are bits/sec. The PDF for the amount of time the source is on during the interval $[t_0, t]$ can be found by first finding the PDF for the random variable S(t) as defined by,

$$S(t) = \int_{t0}^{t} X(t')dt'$$

but according to the literature [27], the PDF for S(t) is not readily obtainable. However, the moments E[S(t)] and $E[S^2(t)]$ can be computed. To find them, it is first necessary to compute the moments of X(t) [27].

$$\begin{aligned} E[X^k(t)] &= \sum_{n=0}^{1} n^k P_n(t|n_0,t_0) \\ &= 0^k P_0(t|n_0,t_0) + 1^k P_1(t|n_0,t_0) \\ &= P_1(t|n_0,t_0) \end{aligned}$$

So, the mean of X(t) is simply $P_1(t|n_0,t_0)$, defined in Chapter III, and the variance of X(t) can be found as the following [27]:

$$\begin{aligned}\mathrm{var}[X(t)] &= E[X^2(t)] - E[X(t)]^2 \\ &= P_1(t|n_0,t_0) - P_1^2(t|n_0,t_0) \\ &= P_1(t|n_0,t_0)[1 - P_1(t|n_0,t_0)] \\ &= P_1(t|n_0,t_0)P_0(t|n_0,t_0)\end{aligned}$$

Also of interest is the covariance of the process, $\mathrm{cov}[X(t_1)X(t_2)]$ where $t_0 \le t_1 \le t_2$, which can be found in the following manner [27],

$$\begin{aligned}\mathrm{cov}[X(t_1)X(t_2)] &= E[X(t_1)X(t_2)] - E[X(t_1)]E[X(t_2)] \\ &= \sum_{n_1=0}^{1}\sum_{n_2=0}^{1} n_1 n_2 P_{n_1}(t_1|n_0,t_0)P_{n_2}(t_2|n_1,t_1) - P_1(t_1|n_0,t_0)P_1(t_2|n_0,t_0)\end{aligned}$$

evaluating the terms in the summation, this becomes

$$\mathrm{cov}[X(t_1)X(t_2)] = P_1(t_1|n_0,t_0)[P_1(t_2|1,t_1) - P_1(t_2|n_0,t_0)]$$

for $t_0 \le t_1 \le t_2$.

In the three equations above for the mean, variance, and covariance of X(t), it is especially useful to find their long term values by substituting the definition of $P_n(t|n_0,t_0)$ and then evaluating the steady state expected values.

$$\lim_{(t-t_0)\to\infty} E[X(t)] = \frac{\lambda_{shortOff}}{\lambda_{shortOff} + \mu} \qquad \text{(Eq. VI-4)}$$

$$\lim_{(t-t_0)\to\infty} \mathrm{var}[X(t)] = \frac{\lambda_{shortOff}\,\mu}{(\lambda_{shortOff} + \mu)^2} \qquad \text{(Eq. VI-5)}$$

$$\lim_{t_0\to-\infty} \mathrm{cov}[X(t_1)X(t_2)] = \frac{\lambda_{shortOff}\,\mu}{(\lambda_{shortOff} + \mu)^2} e^{-(\lambda_{shortOff}+\mu)(t_2-t_1)} \qquad \text{(Eq. VI-6)}$$

The (Eq. VI-4) is the familiar formula that has been used for utilization. The third equation (Eq. VI-6) shows that the asymptotic decorrelation time for this on-off source is $(\lambda_{shortOff}+\mu)^{-1}$ [27] which will be useful when it is time to investigate the properties of the autocorrelation of the on-off process.

Back to the problem at hand -- it is necessary to compute the mean and variance of the random variable S(t). Results from Markov Model theory show that E[S(t)] can be found by solving,

$$E[S(t)] = \int_{t_0}^{t} E[X(t')]dt' = \int_{t_0}^{t} P_1(t'|n_0,t_0)dt'$$

where $P_1(t|n_0,t_0)$ is defined in Chapter III. Finding the variance of S(t) is a fairly involved process that requires the use of the "Moment Evolution Equations", solving a "somewhat tedious integration" and solving a differential equation [27]. The necessary results can be obtained by finding the long time limit of the mean and variance of S(t) which are,

$$\lim_{(t-t_0)\to\infty} E[S(t)] = \frac{\lambda_{shortOff}}{\lambda_{shortOfff}+\mu}(t-t_0) \qquad \text{(Eq. VI-7)}$$

for the expected value and,

$$\lim_{(t-t_0)\to\infty} \mathrm{var}[S(t)] = \frac{2\lambda_{shortOff}\mu}{(\mu+\lambda_{shortOff})^3}(t-t_0) \qquad \text{(Eq. VI-8)}$$

for the variance. These results are also due to [27]. The exact PDF for S(t), has never been solved [52,53]. Preliminary investigations performed during this research showed that the solution may involve a sum of several differently weighted and truncated Gamma PDFs but, at this point in time, a closed form solution has not been obtained.

The two equations (Eq. VI-7,8) describe the mean and variance of S(t) which is seen to be a function of the amount of time over which the integral is evaluated. As stated above, these equations provide direct insight into the mean and variance for the PDF for M(t,i) because the number of bits generated by the source during a time interval is simply the product of the port speed and the amount of time the source was on during that interval.

So, in developing this approximation to the leaky bucket algorithm, (Eq. VI-7,8) will be used to provide an approximation to the mean and variance of M(t,i) – the distribution for the number of bits generated by the source model during the current T_c interval. It will also be used for the mean and variance of the distribution of $M(T_C,i-1)$ which is the number of bits generated during the last complete T_c interval. (See Fig. VI-1).

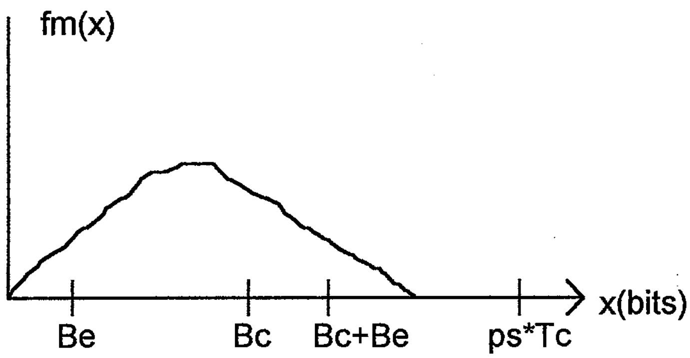

Figure VI-1. The PDF for the random variable $M(T_C,i-1)$ (the number of bits that have arrived during the last complete T_C interval) is limited between 0 and portSpeed x T_C. The exact distribution is unknown, but equations are available for the mean and variance.

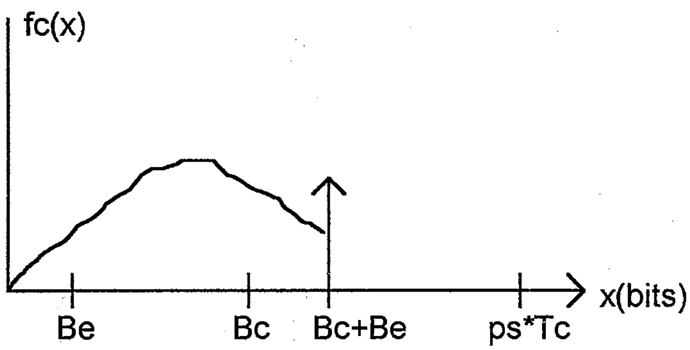

Figure VI-2. The distribution for $C(T_C,i-1)$ (the value of the cumulative bit counter at the end of the last complete T_c interval) is limited to $B_C + B_e$. This figure assumes that C(0,i-1) was zero i.e. no bits were carried in from the interval prior to the last complete one.

PDF of L(i)

The PDF of $C(T_c,i-1)$, which is the contents of the cumulative bit counter at the end of the i-1th T_c interval (assuming for simplicity that L(i-1)=0) can be used to gain insight into the

PDF of L(i) – the number of bits carried in from the last T_c interval. This PDF will have no values greater than B_c+B_e, thus, an impulse will be produced at the value of B_c+B_e. (See Figure VI-2.)

The PDF for L(i) which is the number of bits carried into the current T_c interval is equivalent to C(T_c,i-1) after Bc bits have been leaked out. This PDF is nonzero only in the range from 0 to B_e. The shape is strongly influenced by the two impulses caused by the nonlinearities. (See Figure VI-3.)

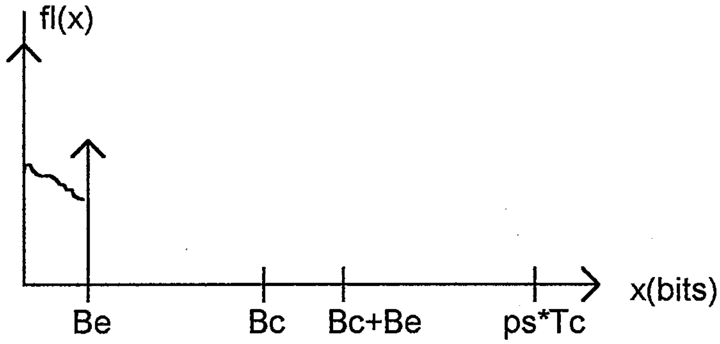

Figure VI-3. The distribution for L(i) (the number of bits carried into the current T_C interval) is equivalent to C(T_C,i-1) after B_c bits have been subtracted. The impulse on the left is produced by the "max" operation in (Eq. VI-2).

To get an idea of what the expected value of L(i) is, it is necessary to make some more approximations. First it will be assumed that the original PDF for M(T_c,i-1) is Gaussian. From experiments, it turns out this is a reasonable assumption when E[on]+E[shortOff] is much less than T_C. For numerical convenience, the PDF can then be approximated as three impulses, one at 0, one at B_e/2, and one at B_e denoted p1,p2 and p3 respectively. Because the variance of L(i) will be small compared to those of the other random variables to be considered, the variance will be assumed to be zero and only an estimate of the mean will be used. If the three probabilities are defined as,

$p1 = prob(M(T_C, i-1) < Bc)$ (the impulse produced at zero)

$p3 = prob(M(T_C, i-1) > Bc + Be)$ (the impulse produced at B_e)

$p2 = 1 - p1 - p3$ (the remaining probability approximated by an impulse at B_e/2) then an estimate can be made of the expected value of L(i) as,

$$E[L] \cong \frac{B_e}{2} p2 + B_e p3$$

The p2 and p3 can be obtained by integrating a Gaussian PDF over the appropriate ranges. The mean and variance of this Gaussian will have the mean and variance from (Eq. VI-7,8) with $t - t_0 = T_C$. Although approximate, this estimate is a function of all the parameters that affect the true value including T_c, $\lambda_{shortOff}$, μ, B_c, B_e, and the port speed.

Now that an estimate of the number of bits carried in from the previous T_c interval has been obtained, it is necessary to consider the PDF of the frame size. This will be necessary because the decision of how to mark a frame will be made only after a frame has actually arrived and its size has been added to the cumulative bit counter.

PDF of Frame Size

Recall that in the two-state on-off model the PDF of the message size is an exponential distribution. The frames, however, are limited to some maximum size. A common value for the maximum frame size is 4096 bytes = 32768 bits. It is necessary that the PDF for the frame size be determined.

Let the message size PDF be denoted $f_X(x) = \mu\exp(-\mu x)$ where X is the random variable describing the size, in seconds, of the message and denote the PDF for the frame size distribution as $f_Y(y)$ (also units of seconds). Note that if a message arrives that is less than or equal to the maximum frame size, only one frame is produced and the frame is the same size as the message. If, however, the message is larger than the maximum frame size, one or more full size frames plus a smaller frame are needed to hold the message. Graphically, this is shown in (Fig. VI-4) and is also illustrated in the following table (Table VI-1) where t_f denotes the time for a maximum sized frame.

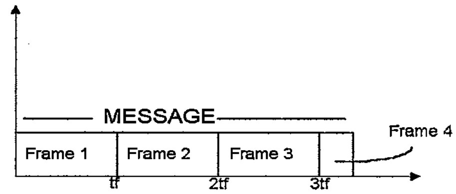

Figure VI-4. Example where a message is big enough to fill 3 full-sized frames and one smaller frame.

Comments	Message Size	Frame PDF $f_Y(y)$	Impulse Component
No full frames	$0 \le x < t_f$	$f_Y(x)=f_X(x)$	0
One full frame	$t_f \le x < 2t_f$	$f_Y(x-t_f)=1/2\ f_X(x)$	$f_Y(t_f)=1/2\ [F_X(2t_f)-F_X(t_f)]$
Two full frames	$2t_f \le x < 3t_f$	$f_Y(x-2t_f)=1/3\ f_X(x)$	$f_Y(t_f)=2/3\ [F_X(3t_f)-F_X(2t_f)]$
Three full frames	$3t_f \le x < 4t_f$	$f_Y(x-3t_f)=1/4\ f_X(x)$	$f_Y(t_f)=3/4\ [F_X(4t_f)-F_X(3t_f)]$
etc.			

Table VI-1. Illustration of how the PDF of the message size maps to both an impulse component, for full sized frames, and a weighted version of the original PDF for partially filled frames. "F_X" denotes the cumulative density function of f_X.

From the table it can be seen that the PDF for Y can be written,

$$f_Y(y)=\sum_{j=0}^{\infty} f_X(y+jt_f)[1-\frac{j}{j+1}] \qquad \text{for } 0\le y<t_f \qquad \text{(Eq. VI-9a)}$$

and,

$$f_Y(y)=\sum_{j=0}^{\infty} \int_{jt_f}^{(j+1)t_f} f_X(\tau)d\tau[\frac{j}{j+1}] \qquad \text{for } y=t_f \qquad \text{(Eq. VI-9b)}$$

and it is noted that (Eq. VI-9a) describes the smooth part of the PDF and (Eq. VI-9b) describes an impulse function that occurs at t_f. (See Fig. VI-5.)

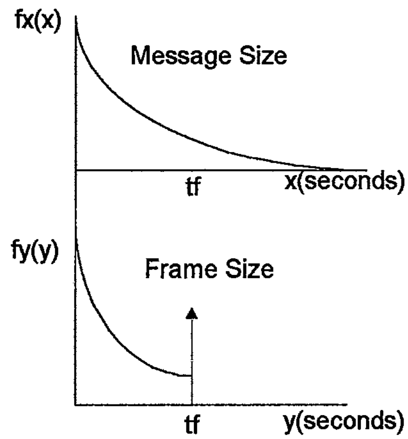

Figure VI-5. The message size PDF maps to a truncated exponential distribution produced by the partially filled frames and an impulse produced by the maximum sized frames.

To find the expected value of the frame size, the component of two moments M_1 and M_2 will be considered such that $E[Y]=M_1+M_2$. M_1 and M_2 are computed as,

$$M_1 = \sum_{i=0}^{\infty} \int_0^{t_f} \tau f_X(\tau + jt_f) d\tau [1 - \frac{j}{j+1}]$$

$$M_2 = t_f \sum_{j=0}^{\infty} \int_{jt_f}^{(j+1)t_f} f_X(\tau) d\tau [\frac{j}{j+1}]$$

$$= t_f \sum_{j=0}^{\infty} [F((j+1)t_f) - F(jt_f)][\frac{j}{j+1}]$$

Applying the equations for M_1 and M_2 to the problem at hand, the following two things can be observed. First,

$$f_X(\tau + jt_f) = \mu e^{-\mu(\tau + jt_f)} = e^{-\mu j t_f} f_X(\tau)$$

and second, that the integral evaluates as below.

$$\int_{-\infty}^{t_f} \tau f_X(\tau)d\tau = \int_{0}^{t_f} \tau\mu e^{-\mu\tau}d\tau$$
$$= \frac{e^{-\mu\tau}}{\mu}(-\mu\tau - 1)\Big|_0^{t_f}$$
$$= \frac{e^{-\mu t_f}}{\mu}\left(-\mu t_f - 1\right) + \frac{1}{\mu}.$$

The equations below summarize the computations necessary to compute the expected frame size where the maximum frame size is t_f and the message sizes are produced by an exponential distribution with average size $1/\mu$:

$$\frac{E[FrameSize]}{portSpeed} = E[Y] = M_1 + M_2 \qquad \text{(Eq. VI-10)}$$

where,

$$M_1 = B_1\sum_{j=0}^{\infty}(B_2)^j[1 - \frac{j}{j+1}]$$

with,

$$B_1 = \frac{e^{-\mu t_f}}{\mu}\left(-\mu t_f - 1\right) + \frac{1}{\mu}$$
$$B_2 = e^{-\mu t_f}$$

and

$$M_2 = t_f\sum_{j=0}^{\infty}[F((j+1)t_f) - F(jt_f)][\frac{j}{j+1}]$$

with

$$F(t) = 1 - e^{-\mu t}$$

"FrameSize" has units of bits and Y has units of seconds in (Eq. VI-10). The two are related by the port speed of the circuit.

An equation for the expected frame size has now been obtained. An equation for the variance could also be computed, but for the time being just the mean should be sufficient. The reason: the variance of the frame size will be small relative to the other variables being considered.

Combining these Results

Had it been possible to obtain the actual PDFs of M(t,i), L(i), and the frame size, then these three PDFs would have been convolved to obtain the PDF for C(t,i). However, it was not possible to come up with actual PDFs; only equations for the mean and variance of the M(t,i) were available in the literature and there were nonlinearities imposed by the operation of the leaky-bucket algorithm. As a result, it was necessary to make some simplifications and consider only the mean and variance of these random variables. The variance of L(i) and the Frame size were assumed to be small. This assumption was made based on the observation that the C(t,i) varies from 0 to B_c+B_e bits whereas, based on examining the actual traffic statistics, the frames are typically on the order of 2000 bits and L(i) is a "compressed" (see Fig VI-3) PDF limited to B_e. With these assumptions, the convolution,

$$f_{C(t+\Delta t,i)} = f_{M(t,i)} * f_{L(i)} * f_{frameSize}$$

becomes the convolution of $f_{M(t,i)}$ with two impulse functions, one at E[L(i)] and one at E[FrameSize], and so

$$E[C(t+\Delta t,i)] \approx E[M(t,i)] + E[L(i)] + E[\text{Frame Size}] \qquad \text{(Eq. VI-11a)}$$

$$\text{var}[C(t+\Delta t,i)] \approx \text{var}[M(t,i)] \qquad \text{(Eq. VI-11b)}$$

where the E[FrameSize] is included because the contents of the current frame is added to the cumulative bit counter before the decision of how to mark the frame is made and Δt is used here to denote that the frame being tagged is assumed to have arrived and its size added to the cumulative bit counter during the time between t and t+Δt.

These equations (Eq. VI-11a,b) can now be used to estimate the probability of frames being marked non-DE, DE, or discarded.

To compute these probabilities it is first observed that the probability of a frame arriving anytime during the T_c interval is uniformly distributed. C(t,i) is a function of time depending on where in the T_c interval it is measured and so the probabilities should be computed at all times during the interval and the results should be averaged – effectively integrated. For computational ease, the T_c interval was divided into ten discrete time steps. At each time step, the mean and variance of M(t,i) was computed from (Eq. VI-7,8) and the E[L] and E[FrameSize] were added to the mean. With the assumption that M(t,i) was Gaussian, the probability that this random variable exceeded either B_C or B_C+B_e was computed by integrating a Gaussian PDF. Next, the results from these ten discrete time steps were averaged to get the resulting average p(DE) over the entire T_c interval which correspond to %DE.

In summary, the algorithm to compute %DE for a given set of model and PVC parameters by using the approximation developed here is given below. To perform the desired operation of rapidly obtaining the model parameters given the observed statistics and PVC parameters, this algorithm should be placed inside a minimization routine.

1) Compute E[L(i)] by
   a) computing the mean and variance of $M(T_c,i-1)$ using (Eq. VI-7,8).
   b) integrating a Gaussian PDF with this mean and variance to find p2 and p3 in the equation for E[L(i)].
2) Compute E[FrameSize] using (Eq. VI-10).
3) Divide the T_c interval into 10 discrete time steps. Compute $E[C(t+\Delta t,i)]$ and $var[C(t+\Delta t,i)]$ at each time step using (Eq. VI-11a,b) where E[M(t,i)] and var[M(t,i)] are computed from (Eq. VI-7,8).
4) At each time step in (3) integrate a Gaussian PDF with this mean and variance to compute prob(DE). Store this set of results from each time step.
5) Average all the prob(DE)'s from each time step computed in (4) to obtain the estimate of the %DE.

Comparing the Approximation to the Simulation

The performance of this approximation will now be compared to the performance of the simulation. As a demonstration, the following values were chosen:

E[on] = 0.03
portSpeed = 64,000
T_C = 1.0 sec.
Bc = 32,000 (case 1), 16,000 (case 2)
Be = 16,000 (case 1), 32,000 (case 2)
CIR = 32,000 (case 1), 16,000 (case 2)

and the E[shortOff] time was varied from 0.2 to 0.02 to which resulted in utilizations from 13% to 60%.

Two cases were chosen. Case 1 was with Bc=32K, and Be=16K. This produced a CIR of 32K which was 50% of the port speed. Note that the two lines on the right in (Figure VI-4) represent this case. The second case was with Bc=16K, and Be=32K which gives a CIR of 16K, 25% of the port speed. The pair of lines on the left side of (Figure VI-6) show this case. Note that in both cases, the approximation slightly overestimates the %DE for low utilizations, but underestimates it for higher utilizations. It may be possible to improve this situation, but the approximation appears to perform adequately for the needs of this research which are: rapidly estimate the model parameters for a large collection of PVC measurements taken during the "busy hour." These results are typical of other tests of the approximation using various port speeds and parameters.

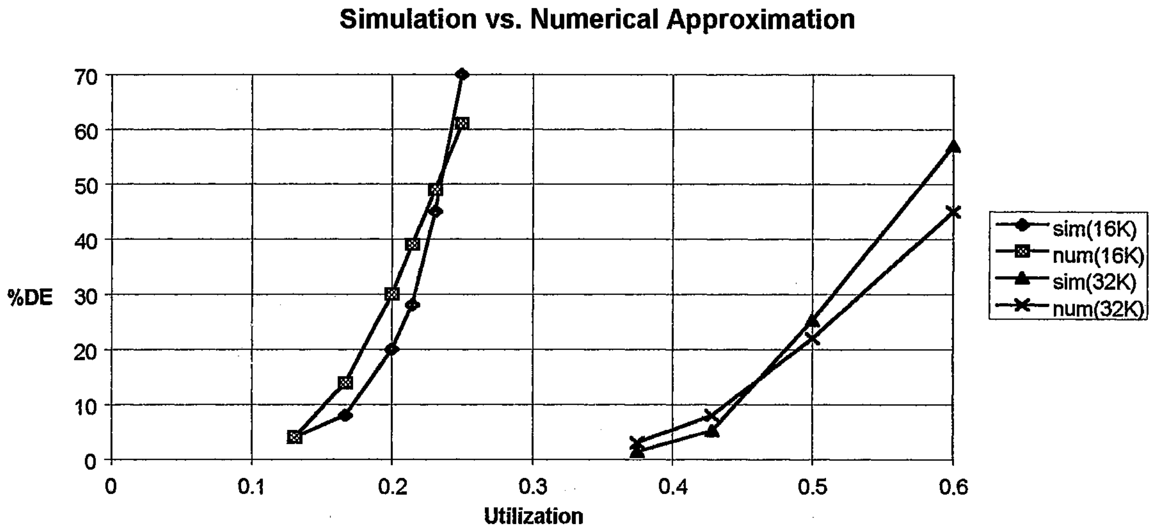

Figure VI-6. Comparison between numerical approximation to leaky bucket algorithm and simulation results. In this plot portSpeed=64K. The two lines on the left represent the results when Bc=16K and Be=32K; the two on the right are when Bc=32K and Be = 16K. (T_C = 1.0 s in both cases.)

Summary

The approximation developed in this section provides a faster method of estimating the %DE frames resulting from an on-off traffic source for a given port speed, Bc, Be, and CIR. When placed inside a minimization routine, this approximation makes it possible to rapidly estimate the two-state model parameters, $\lambda_{shortOff}$ and μ, that would produce the measured E[FrameSize] and %DE values. With this information, the other two three-state model parameters, p and $\lambda_{longOff}$ that fit the measurements can be found mathematically.

In the next chapter, the characteristics of the aggregate traffic stream, produced by several traffic models will be investigated.

VII. The Aggregate Streams

This chapter describes the methods used to determine the aggregate traffic streams (the mixture of multiple virtual circuit traffic sharing the same trunk) by using the approximate PVC routing coupled with the traffic model that has been developed in the previous chapters. The statistical characteristics of the aggregate stream will also be discussed in this chapter.

Overview of Virtual Circuit Routing

As virtual connections are added to the network, their route is chosen based on a routing algorithm which chooses the "best" route through the network based on distances of the trunks, number of trunks that must be traversed, the number of virtual circuits already routed over each trunk and the "effective bandwidth" available on each trunk.

The specific implementation of the routing algorithm is the choice of the Frame Relay equipment provider. Some vendors, such as Stratacom, choose to develop and use proprietary routing schemes. Others choose public domain routing algorithms such as the Open Shortest Path First (OSPF) [48] routing algorithm with some minor modifications.

It is possible to *approximate* the actual routing of the PVCs by using an optimal routing strategy with a cost function chosen correctly. That is, by assigning costs to links in a manner similar to how the network equipment vendor does, it is possible to obtain a close estimate of the actual routing. It is not possible to exactly match the routing on the network without knowing the exact order in which the circuits were originally routed. This is because circuits will be routed over one link until that link is no longer the least costly at which time the remaining circuits will be routed over other links. In practice, however, the approximate routing is close enough.

For this research, a software tool was developed that represents the links, switches, and PVCs and routes the PVCs using a routing strategy similar to that used by the equipment provider. Discussions with network engineers indicate that the routing by this tool accurately approximates the routing in the actual network [49].

It can now be assumed that the trunk traffic can be described by the aggregate of the multiple three-state traffic models with parameters determined from the techniques described in

Chapters V and VI. The validity of this assumption will be tested by comparing measurements of trunk traffic to the estimated aggregate stream.

Sampling the Aggregate Stream

The traffic model developed in this research is a continuous time Markov process. However, the measurements of the actual traffic (described in Chapter IV) are obtained in the form of a count of the number of bytes that have traversed the PVC during the period between sampling times -- a discrete process. To reconcile the two, it will be necessary to convert the output of the model to a discrete time model. In (Eq. VII-1), $M_C(t)$ represents the continuous time output of the traffic model which is either a 1 for "on" or a 0 for "off" and $M(kT_s)$ is the sampled sequence that will be used to represent the output as,

$$M(kT_s) = \int_{(k-1)T_s}^{kT_s} M_C(t)dt \qquad \text{(sec)} \qquad \text{(Eq. VII-1)}$$

where k=1,2,... and T_s is a new variable defined to be the sampling interval. The value of each sample in the sequence $M(kT_s)$ describes the amount of time, in seconds, that the model is on during the interval T_s.

The choice for T_s must now be made. In Chapter II, the Cascade switch buffer configuration was discussed. The total buffer capacity was given as 32kbytes, or 256kbits. On a T1 trunk, this buffer will hold 256/1544 = 0.16 (s) worth of traffic. For the model to describe how the amount of traffic in the switch gets discarded over time, the traffic model should be sampled faster than this rate. Based on this observation, a sampling time of 0.1 (s) is tentatively chosen (this time could be reduced or increased as necessary depending on buffer sizes and trunk speeds or the results of further investigation).

Statistical Characteristics of Models

This section will discuss the statistical characteristics of the two-state model and the three state model developed in this research. The mean, variance, and autocorrelation of the two different models will be compared. Also, the statistical qualities of the aggregate will be discussed.

Two State Model

For the two state model, (Eq. VI-7,8) from Chapter VI can be used here directly to describe the mean and variance, μ_{M2} and σ^2_{M2} respectively, of M2(kT_S), the sequence created by integrating the output of the two state traffic model over the interval T_s as in (Eq. VII-1), as,

$$\mu_{M2} = E[M2(kT_s)] = \frac{\lambda_{shortOff}}{_{shortOff}\lambda + \mu} T_s \qquad \text{(sec)} \qquad \text{(Eq. VII-2)}$$

for the mean and,

$$\sigma^2_{M2} = \text{var}[M2(kT_s)] = \frac{2\lambda_{shortOff}\mu T_s}{(\lambda_{shortOff} + \mu)^3} \qquad (\text{sec}^2) \qquad \text{(Eq. VII-3)}$$

for the variance. Note that these equations are normalized and don't account for the port speed of the line. They describe the amount of time and variance, in seconds and seconds^2 respectively, that the two-state model can be expected to be on during the time interval T_s.

Three State Model

With the three state case, there is no equation available from the literature that can be used in a straightforward fashion so some approximate results will be developed.

Below is a summary of the assumptions made in developing the original three state model:

μ in 2-state and 3-state model are the same (1/E[on])
λ in 2-state model is $\lambda_{shortOff}$ in 3-state model (1/E[off] = 1/E[shortOff])
$\lambda_{longOff}$ in 3-state model = 1/E[longOff]

p is the probability of choosing the shortOff distribution, 1-p is probability of choosing the longOff
E[longOff] >> E[shortOff]
E[longOff] >> $T_C \geq T_s$

These assumptions were necessary to develop the method of fitting the model and will also be used to develop the approximations for the mean and variance here.

The mean and variance of the random process, M3(kT_s), created by integrating the output from the 3-state model, as in (Eq. VII-1), will be denoted μ_{M3} and σ_{M3}^2 respectively and are,

$$\mu_{M3} = \frac{E[on]\cdot T_s}{E[on] + p\cdot E[shortOff] + (1-p)\cdot E[longOff]} \quad \text{(sec)} \qquad \text{(Eq. VII-4)}$$

for the mean. Note that in (Eq. VII-4) the numerator is the expected time in the on state and the denominator is the expected time for an entire cycle. And,

$$\sigma_{M3}^2 = \frac{\mu_{M3}}{\mu_{M2}}(\sigma_{M2}^2 + \mu_{M2}^2 - \mu_{M2}\mu_{M3}) \quad (\text{sec}^2) \qquad \text{(Eq. VII-5)}$$

for the variance.

Proof of (Eq. VII-5):

Consider $f_{M2}(\tau)$ and $f_{M3}(\tau)$ to be the PDFs of the random processes M2(kT_s) and M3(kT_s) respectively, described above. The mean and variance for $f_{M2}(\tau)$ are given (Eq. VII-2,3), but as mentioned in Chapter VI, the exact PDF remains unknown.

Since T_s << E[longOff], the distribution of the observations of the 3-state process, $f_{M3}(\tau)$ will be related to $f_{M2}(\tau)$ by way of,

$$f_{M3}(\tau) = (1-q)\delta(0) + qf_{M2}(\tau) \qquad \tau \geq 0 \qquad \text{(Eq. VII-6)}$$

where q is some parameter between 0 and 1. In other words, f_{M3} will be the combination of two things: 1) a distribution with exactly the same shape at $f_{M2}(\tau)$, but weighted down by a factor of q and 2) an impulse at 0 with probability (1-q) produced by samples of the process when the model is in the longOff state.

Applying the definition of expected value to both sides of (Eq. VII-6),

$$\int_0^\infty \tau f_{M3}(\tau)d\tau = 0 + q\int_0^\infty \tau f_{M2}(\tau)d\tau$$

so,

$$q = \frac{\mu_{M3}}{\mu_{M2}}$$

and it is also possible to apply the second moment operator to both sides of (Eq. VII-6) and show that,

$$q = \frac{E[M3^2]}{E[M2^2]}$$

so q is the ratio of both the means and the second moments of the distributions.

By using the equation, $\sigma_X^2 = E[X^2]-\mu_X^2$ and equating the expressions for q, the following can be written,

$$\frac{\mu_{M3}}{\mu_{M2}} = \frac{\mu_{M3}^2 + \sigma_{M3}^2}{\mu_{M2}^2 + \sigma_{M2}^2} \qquad \text{(Eq. VII-7)}$$

which can be solved for σ_{M3}^2 to obtain (Eq. VII-5).

Example

This example will test the usefulness of the equations in this chapter by creating an aggregate traffic stream consisting of 15 circuits with the same parameters used in the example in Chapter V where the three-state model was fit to the actual measurements. The variance and mean using (Eq VII-4,5) will then be compared to the measurements obtained by simulation. Repeating the parameters used in Chapter V here,

E[on] = 0.025 (sec)
E[shortOff] = 0.0285 (sec)
E[longOff] = 10.0 (sec)
p = 0.993
portSpeed = 64,000 bits/sec

and the sampling time of the aggregate stream, T_S will be 0.1 (s).

An aggregate stream consisting of 15 streams with these parameters was generated. (Fig. VII-1) shows 60 seconds (600 samples) of this stream. (Fig. VII-2) shows the histogram of the sampled values, $M(kT_s)$.

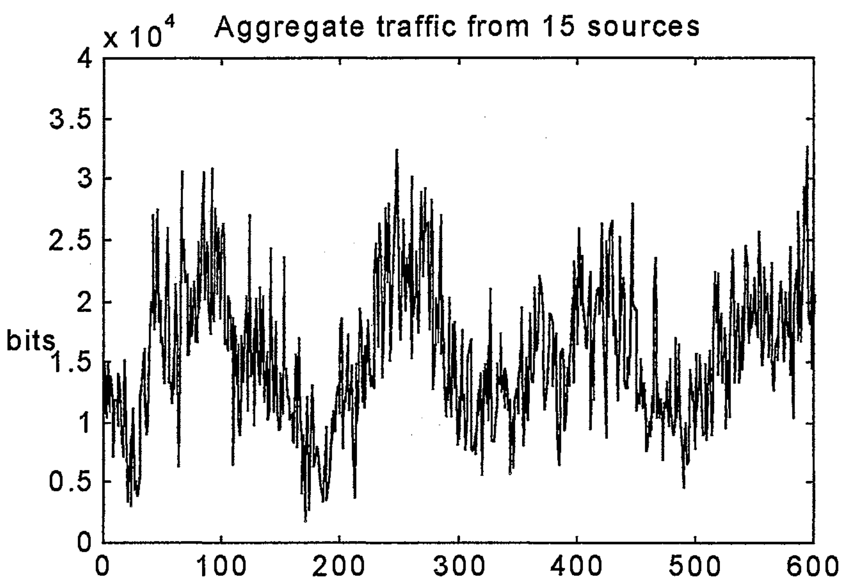

Figure VII-1. Sixty seconds of aggregate stream produced by 15 three-state traffic sources. T_s = 0.1 seconds.

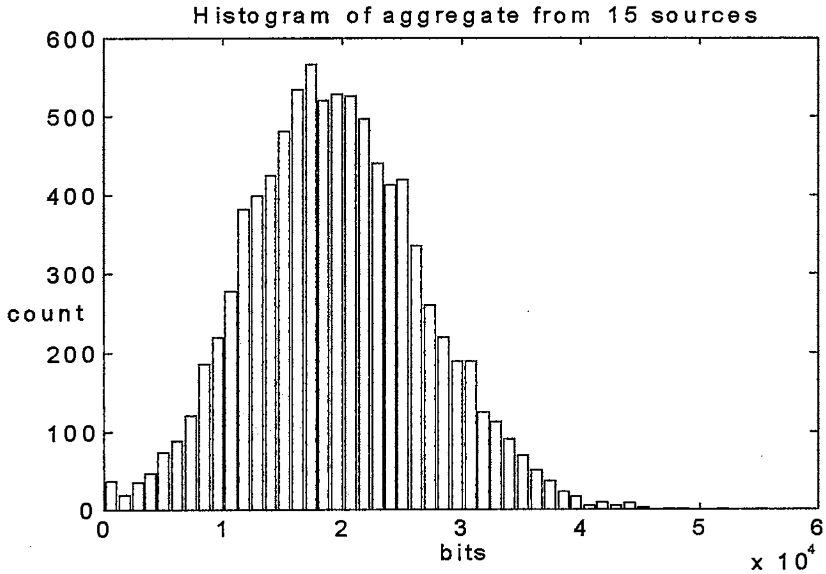

Figure VII-2. Histogram of samples for aggregate traffic stream produced by 15 three-state traffic models. T_s = 0.1 seconds.

Measuring the mean and standard deviation of this aggregate stream yields:

Measured mean = 1.949e4 (bits)

Measured standard deviation = 7.384e3 (bits)

Now, using (Eq. VII-4), the theoretical mean and variance will be computed as,

$$\mu_{M3} = \frac{0.025(0.1)}{0.025 + (0.993)0.0285 + (0.007)(10.0)} \quad \text{(sec)}$$
$$= 0.0203$$

which is the mean for a single three-state model. So this says that during a 0.1 (sec) time period, the model can be expected to be 0.0203 (sec) which is approximately equivalent to 20%.

The variance (Eq. VII-5) requires the result of (Eq. VII-4) and also the mean and variance during the bursting periods given by (Eq. VII-2) and (Eq. VII-3) respectively.

From (Eq. VII-2), the μ_{M2} is computed to be,

$$\mu_{M2} = \frac{\frac{1}{0.0285}(0.1)}{\frac{1}{0.025} + \frac{1}{0.0285}} = 0.047 \qquad \text{(sec)}$$

and the variance from (Eq. VII-3) is,

$$\sigma_{M2}^2 = \frac{2\frac{1}{0.025}\frac{1}{0.0285}(0.1)}{(\frac{1}{0.025} + \frac{1}{0.0285})^3} = 6.63(10^{-4}) \qquad (\text{sec}^2)$$

so now the variance (Eq. VII-5) for the three-state model can be computed as,

$$\sigma_{M3}^2 = \frac{0.0203}{0.047}(6.63(10^{-4}) + 0.047^2 - (0.0203)(0.047)) = 8.28(10^{-4}) \quad (\text{sec}^2)$$

Note that these equations do not yet account for the number of circuits or port speed.

To compare the values to measured values, the number of circuits and port speed of the circuits must be included. Since the models are statistically independent, their means and variances can be added directly. The aggregate mean, μ_{agg} ,given that there are 15 circuits each with a port speed of 64000 is,

$$\mu_{agg} = (0.0203)(15)(64000) = 1.948(10^4) \qquad \text{(bits)}$$

which closely matches the measured mean.

To compute the standard deviation of the aggregate, σ_{agg}, the variances must be added and then the port speed must be included after taking the square root of the total variance.

$$\sigma_{agg} = (8.28(10^{-4})(15))^{1/2}(64000) = 7.13(10^3) \quad \text{(bits)}$$

which is slightly lower than the measured value of 7.38×10^3 probably due the approximations made in the development of (Eq. VII-5). This amount of accuracy should be sufficient for practical design problems where a high degree of precision is not required.

Importance of Representing Burstiness

In this section, the statistical characteristics from the three-state model developed in this research will be compared to those of a two state model fit to match the utilization and expected frame size. It should be noted that choosing a two-state model can be a mistake resulting in under-estimating the variance and autocorrelation and thus under-designing the trunks. This will serve to illustrate the importance of choosing a traffic model such as the three-state model that reflects the burstiness of the circuits.

Differences in Variance

Again, the example measurements used throughout the research will be used here. To fit the two-state model to match the utilization and expected frame size requires a μ and $\lambda_{shortOff}$ parameter of 0.025 and 0.1 respectively. The utilization from (Eq. III-6) is 0.2. Again, assuming a T_S of 0.1 seconds and using (Eq. VII-2), μ_{M2} can be obtained as,

$$\mu_{M2} = \frac{0.025(0.1)}{(0.025+0.1)} = 0.02 \qquad \text{(sec)}$$

and the variance is:

$$\sigma_{M2}^2 = \frac{2(\frac{1}{0.025})(\frac{1}{0.1})(0.1)}{(\frac{1}{0.1}+\frac{1}{0.025})^3} = 6.4(10^{-4}) \qquad (\text{sec}^2)$$

which for our example of 15 circuits and 64000 bits/sec would give standard deviation of,

$$\sigma_{agg} = (15(6.4x10^{-4}))^{1/2}(64000) = 6.2(10^3) \qquad \text{(bits)}$$

which is significantly lower than the standard deviation of $7.13\text{x}10^3$ obtained by assuming the traffic was produced by a three-state model.

Differences in Autocorrelation

Even more interesting than the difference in variance is a comparison of the autocorrelation of the two-state and three-state models. (Figure VII-3) shows the autocorrelation function, $R(\tau)$, of the aggregate for 15 two-state traffic models. There is no significant correlation for non-zero values of τ. Recall in Chapter VI that for the two-state case (Eq. VI-6) showed that the asymptotic decorrelation time is $(\lambda_{shortOff}+\mu)^{-1}$ = $(10+40)^{-1}$ = 0.02 (s) which in this example is less than the sampling time of 0.1(s). For the aggregate of three-state traffic models, however, (see Figure VII-4) there are significant values of R(τ) for τ≤10(secs). Certainly it is no coincidence that E[longOff] was 10.0(sec) in the model.

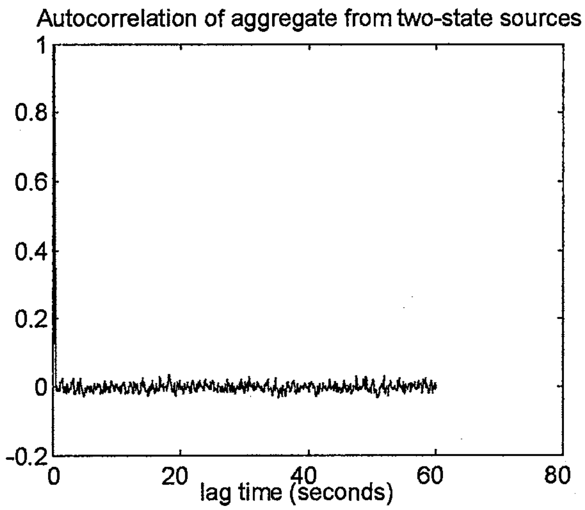

Figure VII-3. Autocorrelation of aggregate stream from 15 two-state on-off processes. Note that the mean of the signal was removed and the autocorrelation was normalized before plotting.

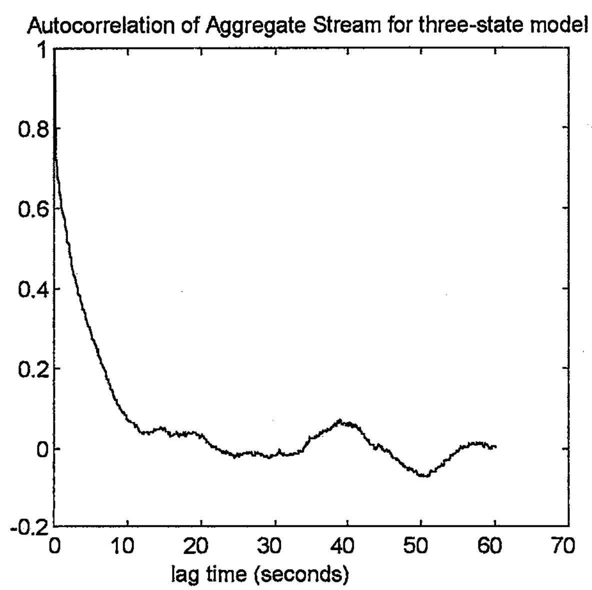

Figure VII-4. Autocorrelation of aggregate stream of 15 sources. Note there is significant correlation within 10 seconds (100 samples). The mean was removed and the autocorrelation was normalized before plotting.

The autocorrelation function is important because it directly impacts the amount of buffering that is required by the switches. Conversely, for a fixed amount of buffer space $R(\tau)$ directly impacts the size of the trunk necessary to prevent packet losses. The reason is as follows: $R(\tau)$ is a measure of how likely the traffic is to stay high for several time intervals once it has become high. This is important because a trunk can tolerate being overloaded for only a few milliseconds. During this time, the buffers will begin filling. If traffic continues to be offered to the switch at level exceeding the trunk speed, the buffers will overflow.

In a recent paper [50] statistics of LAN traffic were collected over a period of three years. The authors found that the traffic exhibited "self-similar" characteristics. Specifically they determined that it contained correlation over the range of milliseconds, seconds, hours, and even days. Since a good deal of the traffic carried on Frame Relay is from interconnected LANs, these facts are important (see also [51]). Clearly they support the notion that the traffic model should contain long-term correlations, perhaps even more than the 10 seconds chosen when fitting the three-state model in this example. This is certainly an area for further research.

VIII. Switch Buffering

Introduction

In previous chapters, a model for PVC traffic along with a method of setting its parameters from actual traffic measurements was developed. The characteristics of the aggregate stream produced by combining several of these sources was briefly explored.

The remaining problem in this research is to quantify the level of service the customer can expect to receive given the characteristics of the traffic he is transmitting, the traffic transmitted by other users and the configuration of the network itself.

For the problem addressed in this chapter, it is assumed that a three-state traffic source with appropriate parameters has been found for each PVC feeding into the switch. The items to be determined are the size of the switch's trunk circuit and the amount of buffering necessary to attain a desired level of performance for the customers. Conversely, given the size of the trunk, the amount of buffering in the switch, and the traffic models sharing the trunk, the quality of service received by those customers can be determined.

The format of this chapter will be as follows: First, the current method of sizing trunks in the network will be described. Next, using some simplifying assumptions, the range of trunk sizes possible will be explored via an example. Finally, it will be shown that there are some equations available that, when used with the three-state traffic model, can be used to provide a more accurate value of the necessary trunk size for the desired level of QoS. A discrete event simulation was written for this research and will be used to verify the usefulness of the equations.

Oversubscription

In many of today's Frame Relay networks, links are sized based on the oversubscription factor (OS) which is defined as,

$$OS = \frac{\sum CIR}{TrunkBW} \qquad \text{(Eq. VIII-1)}$$

and is the sum of the Committed Information Rates (CIR)s of all PVCs on the link divided by the actual bandwidth of the trunk.

This factor is used in conjunction with the routing algorithm in the following manner: A target oversubscription rate is set for every link on the network. The PVCs are then routed over the links until the links are "full" which occurs when the following inequality holds:

$$\sum CIR > OS \cdot TrunkBW .$$

There are a couple of difficulties with using this technique. First, the target oversubscription rate must be chosen correctly so that the links are loaded neither too lightly nor heavily. Second, the notion of an oversubscription rate relies on there being a strong relationship between the CIR and the amount of traffic the customer will actually transmit. Figure VIII-1 shows that, in fact, these parameters are not highly correlated. (The linear regression correlation coefficient was computed to be 0.589.) The numbers for this plot were obtained by adding the total number of octets that traversed the trunk during the busy hour and comparing this to the sum of the CIRs for all the PVCs routed on that trunk. Each point on the plot represents one trunk in the network.

From one set of collected statistics, which represented the busy hour of a single day for the entire network, the average relationship between CIR and actual traffic was 0.044, meaning that the average customer used only 4.4% of his CIR.

Also important, this plot shows that the total CIRs routed on a trunk and the actual traffic are not highly correlated. This suggests relying on the CIR alone to size the network is an imprecise method which can be improved by knowing more about the actual traffic. This chapter will discuss methods that are more accurate than using the CIR because they use the three-state traffic model, fit from measured statistics, to determine the probability of discarded packets on the trunks.

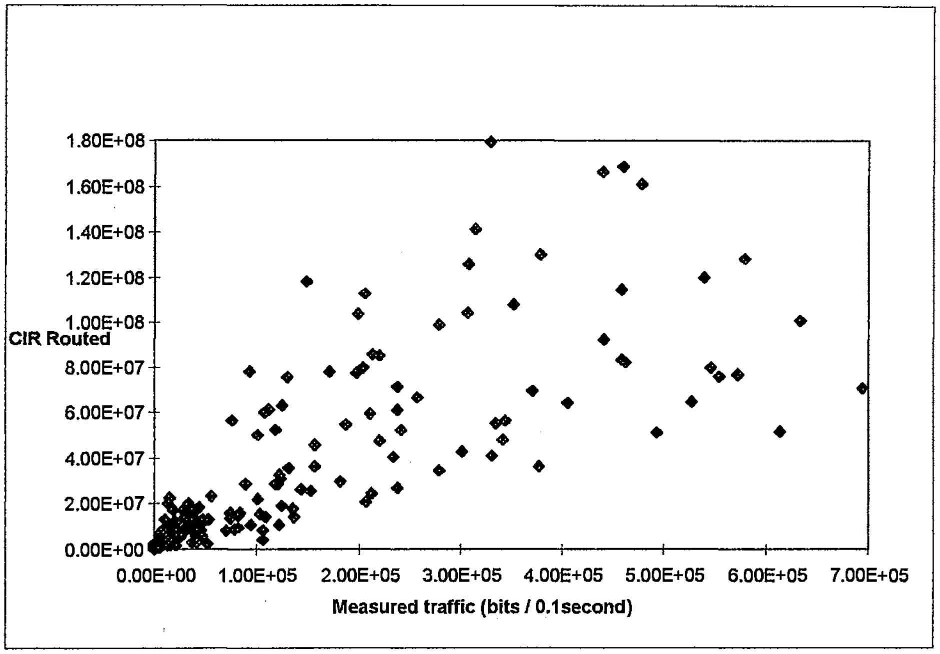

Figure VIII-1 - The relationship between CIR and actual traffic. Each point represents the totals for an individual trunk.

Current issues

Problems similar in nature to this one appear in the literature. In the past they were approached using queuing theory. More recently, work has been oriented towards "fluid-buffer" [18-19,37,40,44,52,56-58] approximations to telecommunications traffic buffering problems. The results in this area are most applicable to this research because the on-off model, which is being used here, can be thought of as a fluid source.

The primary limitation of the results from the literature is that they are too specific and computationally intensive, mainly because the work has been done from the point of view of an ATM switch designer. ATM has strictly defined classes of traffic making it straightforward to estimate the amount of bandwidth consumed by aggregate traffic.

The goal is to find some equations that are not computationally intensive but which provide more guidance than simply using the oversubscription ratio for sizing the trunks.

First, consider the range of possible trunk sizes. As a "worse-case" the network designer could size the trunk to handle the port speeds of all the PVCs using that link. In this situation, every customer could simultaneously transmit at their full port speed and no traffic would be lost.

In such a configuration, the network would have no statistical multiplexing gains over a normal TDM network. Clearly, sizing the network in this way would be unreasonable and not cost effective, as in most normal situations there is an infinitesimally small probability that all of the customers would use their PVCs simultaneously. At the other end of the spectrum, each link could be sized so small that it is capable of providing service to only one active customer, and if two or more PVCs become simultaneously active, the provided service would degrade precipitously. Clearly this too is not reasonable. Trunk sizing between these two extremes is required.

To specify the level of performance required, some target Quality of Service (QoS) will be chosen to be the minimum acceptable performance customers should receive (assuming they are not transmitting DE traffic). This target should be used in sizing the trunks.

The effect of buffer size on oversubscription

As shown in Figure VIII-2, the typical packet switch can be broken into three main stages. The first stage is where traffic arrives and waits to be serviced by the switching matrix.

The switching stage is where the routing of the traffic is performed. The switch matrix examines the frame headers and moves the traffic from the input queue to the appropriate output port buffer.

The third stage is the output stage where traffic waits to be placed on the trunk and sent to the next switch. Traffic being moved from the input queue to the output queue may be discarded if there is no room in the output queue.

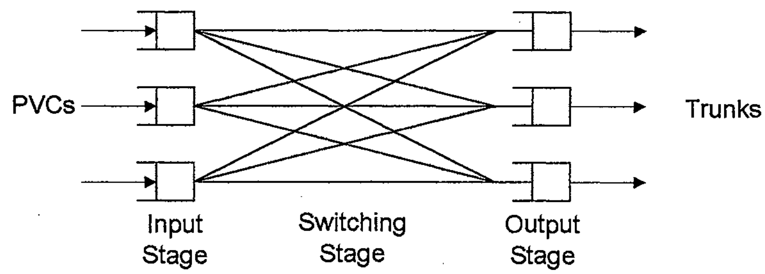

Figure VIII-2. The three main stages in a packet switch. Buffering can occur in both the input and output stages.

If the effects of the input and output buffers are combined into a single buffer then the system becomes much simpler to analyze. Figure VIII-3 shows this simplified model for the case of one trunk. The effects of the input queue are neglected from this model. This model is generally considered to be adequate for modeling the statistical multiplexing properties of a packet switch. [18,19,23,41,58]

The size of this buffer affects how the output trunk needs to be sized. Two extreme cases of either no buffer or infinite buffering in the switch will be compared to illustrate how the amount of switch buffering affects the oversubscription ratio necessary to achieve a certain QoS.

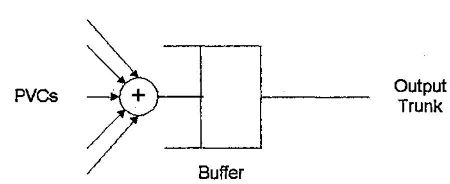

Figure VIII-3. Single-buffer approximation for several PVCs sharing a single output trunk.

No buffering

The "no buffering" case will be defined as the hypothetical scenario where the amount of storage, for traffic to be buffered within the switch, is negligible. The implication of this is that if the sum of the instantaneous input traffic exceeds the output trunk rate then traffic must be discarded; there is nowhere for it to be held while waiting for free capacity on the output trunk. It should be pointed out that, in reality, the switch must have at least a small amount of buffering because it takes the switch matrix a finite amount of time to successively service the traffic at each input port. Also unaccounted for, in this hypothetical case, is that the switches service traffic a frame at a time whereas this no buffering case assumes bits from several different input ports are interleaved on the output trunk. Nevertheless, this case is interesting to examine as an extreme situation.

For this example, a 50 Mbit /sec trunk will be assumed. The question is: How many PVCs can be placed on the trunk so that the probability of the circuits exceeding the line rate is less than 1 in 10^5 (this is approximately 1 second in a 24 hour period)? Assume the probability of each PVC being active is 0.04, the port speed of each PVC is 500 kbits/sec, and the CIR of each is 250 kbits/sec.

For the trunk to be overloaded, more than 100 circuits must be active at the same time. Let N be the number of circuits total. The random variable X will be used to denote the number of circuits that are active at any given time. An equation can be written using the cumulative distribution function for the binomial distribution,

$$F(x) = \sum_{k=0}^{x} \binom{N}{k} p^k (1-p)^{N-k}$$

where p is the probability that each of the PVCs is active, N is the total number of PVCs and x is the maximum number of PVCs that can be active without the trunk becoming overloaded.

For this example, it is necessary to find the N that satisfies,

$$10^{-5} = 1.0 - \sum_{k=0}^{100} \binom{N}{k} (0.04)^k (1-0.04)^{N-k}.$$

Solving for N yields 1611 which is how many circuits, with a burst speed of 500 kbits/sec, that can be loaded on the trunk, given the above assumptions. This corresponds to an oversubscription of

$$OS = \frac{1611(250)}{50000} = 8.06.$$

Figure VIII-4 shows a plot of the oversubscription factor necessary to obtain a p(Blockage) of 1e-5 for various ratios of port speed to CIR and various activity factors (the probability a PVC is active). In this plot, the ratio of the average port speed to the total trunk bandwidth was 100 (denoted X in the equation above and on the plot).

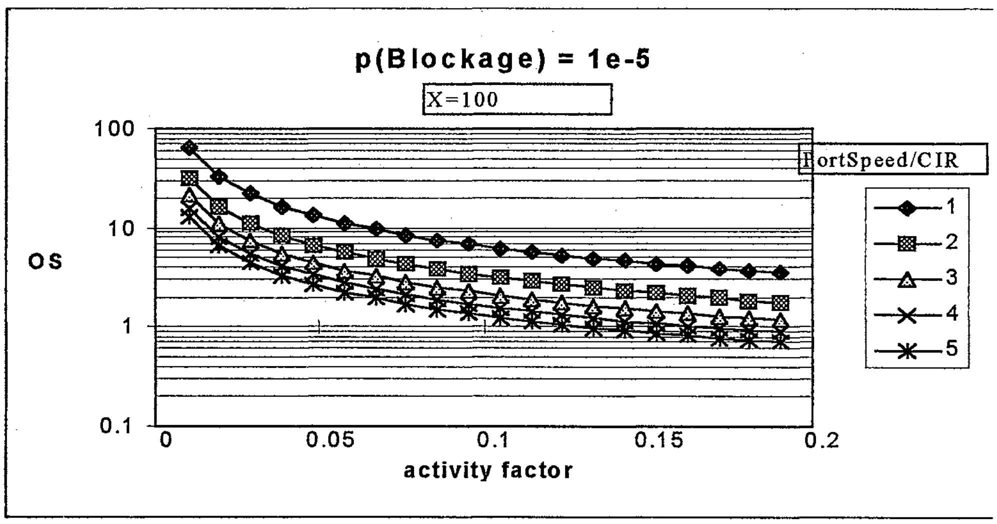

Figure VIII-4. - Oversubscription versus activity factor for various ratios of PVC/CIR (the legend). This plot was generated for with X = 100. The activity factor is the probability that a PVC is active, the oversubscription is defined to be the total CIR routed on a trunk divided by the trunk speed.

The same plot with X = 10 is shown in Figure VIII-5. Notice that as the size of the trunk compared to the PVCs gets smaller, the oversubscription rate must be less to achieve the p(Blockage). This means that for a given probability of blockage, large trunks can be packed more densely than smaller ones.

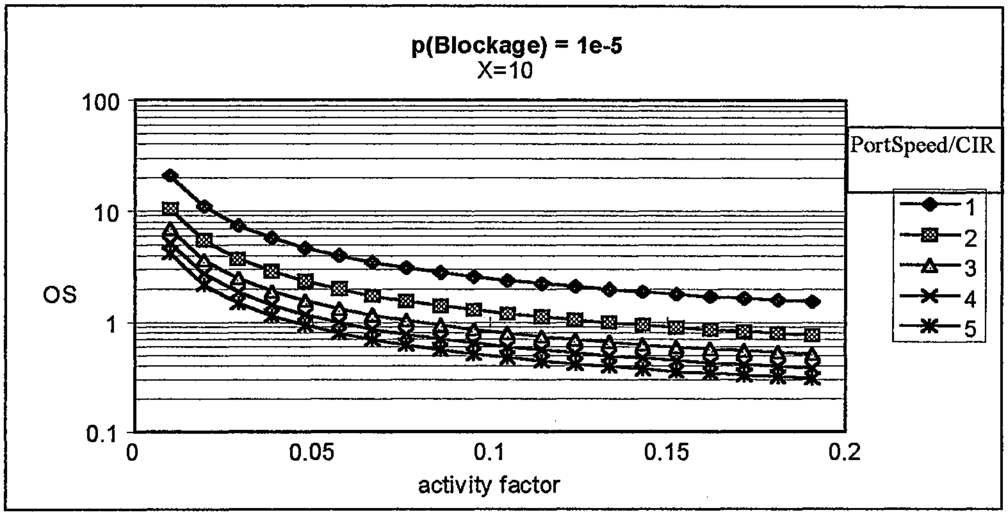

Figure VIII-5. - Oversubscription versus activity factor for various ratios of portSpeed/CIR (the legend). This plot was generated for with X = 10. Notice that a lower oversubscription is required to achieve the same p(Blockage) relative to X=100 (Figure VIII-4).

It is believed, but unproved, that these plots may also be useful for describing the no-buffer oversubscription required when the PVCs consist of a mixture of various model parameters, particularly in cases where there is a large number of PVCs sharing a trunk. Further investigation would require using the multinomial rather than binomial distribution.

Since the equations used for these plots assume no buffering, they could be considered the worst case in terms of the amount of oversubscription that can be safely applied to a trunk. A trunk with adequate buffering can have higher oversubscription rates, as will be demonstrated in the next section.

Infinite Sized Buffers

Under the assumption the switch has infinitely large buffers and as long as the trunk is sized just greater than the average load, no traffic will be discarded and the buffer contents will not grow without bound. In this case, it is not necessary to consider the amount of burstiness in the traffic load because the infinitely sized buffer can temporarily hold even the longest bursts. It should be noted that an infinitely large buffer means a potential for very large delays, particularly when the utilization is close to one.

This case will now be compared to the no-buffer case. The trunk sizing requirement for the infinite buffer case is,

$$N \times probActive \times portSpeed = trunkBW.$$

Here, N is the number of PVCs, probActive is the probability that a particular PVC is in the on state, and trunkBW is the bandwidth of the trunk. Using the definition of the oversubscription,

$$OS = \frac{N \times CIR}{trunkBW},$$

the OS in this case can be written,

$$OS_{\inf Buffer} = \frac{CIR}{probActive \times portSpeed}.$$

For the current example, the OS can be calculated to be 12.5 as compared to 8.06 computed previously for the same case with no buffer.

Figure VIII-6 shows the oversubscription rates necessary for a variety of port speed to CIR ratios and activity factors (probActive above) for the infinite buffer case.

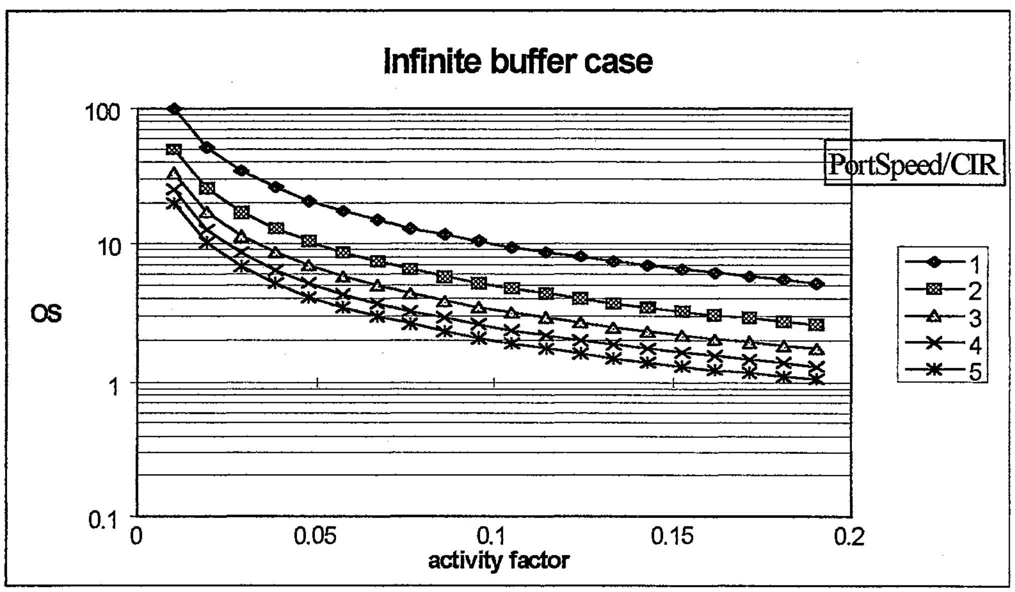

Figure VIII-6 - The infinite buffer case. In this case the average traffic offered is equal to the trunk capacity.

The infinite buffer case differs from the no buffer case, considered previously, in the following important way: For infinite buffering it was assumed that the switch could store data indefinitely while waiting for capacity to become available on the output trunk. The no buffer case had no storage capacity and excess input traffic had to be discarded. It should be clear that, in the real world, the range of allowable oversubscriptions lies between these two extremes. It is this finite but non-zero buffer size case that will be considered next.

Finite Buffer Size

To more accurately compute the probability of loss both the size of the buffer and the burstiness of the traffic must be considered. The remainder of this chapter will focus on applying results from the literature to describe the probability of loss for the case of a finite buffer. It will be necessary to tailor these equations for the special case of the three state traffic model which is unique to this research.

First, the probability of overflow for an on-off source will be described. Next, the case of several homogenous on-off sources feeding a buffer will be considered. Finally, the case of a mixture of on-off sources with different model parameters will be addressed. For the subsequent discussion, c will be used to represent the speed of the link, r will represent the rate at which the traffic source generates data in bits / second, and B is the size of the buffer in bits.

Exact Analysis [23]

The first attempt to analyze the buffer driven by homogeneous two-state on-off source models was made by Anick et al [23]. The essence of this analysis can be illustrated with the following: Consider the two-state exponential on-off model, discussed in previous chapters, where λ is the rate the source goes from the off state to the on state and μ is the rate the source goes from the on state to the off state. Suppose there is one such source that generates data at the rate r.

For this example to be interesting, $r > c$. That is, when the traffic source is on, the source is generating traffic faster than the trunk can handle it and the buffer fills at a rate r-c. When the source is off, the buffer empties at the rate c.

Now, letting Q(t) be a random variable describing the contents of the buffer at time t, the probability of the buffer overflowing is,

prob(overflow) = prob(Q(t) > B)

provided that during the solution process the buffer is assumed to be of infinite size, i.e. Q(t) is unbounded. For cases where the prob(overflow) is small, this has been shown to be a valid assumption [56]. It will also be assumed that prob(overflow) is essentially the equivalent to prob(discard), the probability that a frame is discarded.

Define, $P_i(t,x)$ to be the probability that at time t, i sources are on and the buffer contents do not exceed x. (For this example, i = 0 when the single source is off and i = 1 when the source is on.)

In a small amount of time Δt, three things can happen: The source can be off and turn on, the source can be on and turn off, or the source can remain in its current state while volume is either added or removed from the buffer. The following equations can be written [23],

$$P_0(t+\Delta t,x) = P_0(t,x-\Delta t(-c))(1-\lambda\Delta t) + P_1(t,x)\mu\Delta t \qquad \text{(Eq. VIII-2)}$$

for the probability the source is in state 0 at $t+\Delta t$ with buffer content less than x. And,

$$P_1(t+\Delta t,x) = P_0(t,x)\lambda\Delta t + P_1(t,x-\Delta t(r-c))(1-\mu\Delta t) \qquad \text{(Eq. VIII-3)}$$

for $P_1(t+\Delta t,x)$. These equations assume that compound events, e.g. the sources turning off then on during Δt, occur with a negligible probability and that the buffer contents change only when the source model stays in its current state.

Collecting terms and dividing both sides by Δt, (Eq. VIII-2) can be rewritten,

$$\frac{P_0(t+\Delta t,x) - P_0(t,x+c\Delta t)}{\Delta t} = -\lambda P_0(t,x+c\Delta t) + \mu P_1(t,x)$$

which, for Δt very small, simplifies to,

$$\frac{\partial P_0(t,x)}{\partial t} - c\frac{\partial P_0(t,x)}{\partial x} = -\lambda P_0(t,x) + \mu P_1(t,x).$$

It is the steady state probabilities that are of interest, so the first partial derivative term (the one with respect to t) is zero. Including this, and performing similar steps on the equation for $P_1(t+\Delta t,x)$ allows the pair of equations to be written in matrix form as [23],

$$\mathbf{D}\frac{\mathbf{d}}{\mathbf{dx}}\mathbf{F}(x) = \mathbf{MF}(x) \qquad \text{(Eq. VIII-4)}$$

where,

$$\mathbf{D} = \begin{bmatrix} -c & 0 \\ 0 & r-c \end{bmatrix},$$

$$\mathbf{M} = \begin{bmatrix} -\lambda & \mu \\ \lambda & -\mu \end{bmatrix},$$

and

$$\mathbf{F}(x) = transpose[F_0(x) \quad F_1(x)].$$

It should be noted that $F_0(x)$ and $F_1(x)$ are $P_0(t,x)$ and $P_1(t,x)$ defined above but without the time dependence which is dropped because only the long-time probabilities are of interest.

This differential equation can be solved to obtain **F**(x) and since the overall probability of overflow is the sum of the probabilities of overflow when the source is in either of the two states,

$$G(x) \equiv \Pr(\text{buffer content} > x) = 1 - (F_0(x) + F_1(x)).$$

This equation can be solved to obtain [23],

$$G(x) = \beta e^{zx}, \qquad \text{(Eq. VIII-5)}$$

where,

$$\beta \equiv \frac{pr}{c},$$

and

$$z = \frac{\lambda}{c} - \frac{\mu}{r-c}.$$

with $p=\lambda/(\lambda+\mu)$.

When multiple sources, say N, of the same type are all feeding a single buffer, the analysis is not as simple. Multiple sources require finding the eigenvalues and eigenvectors for the system of equations above, which can be computationally time consuming. An approximate upper bound exists [56] where z_0 is the dominant eigenvalue and β_0 is an approximate coefficient corresponding with this eigenvalue. This equation is,

$$G(x) \le \beta_0 e^{z_0 x} \qquad \text{(Eq. VIII-6)}$$

where,

$$z_0 = \frac{\lambda}{c/N} - \frac{\mu}{r - c/N}$$

and,

$$\beta_0 \approx \left[\frac{(1+\xi)^2}{1/(1-p) + \xi^2/p} \right]^N \cdot \frac{\sqrt{c(Nr-c)/(2\pi N)}}{\dfrac{Nr\xi^2}{[(p/1-p)+\xi^2]} - c},$$

and,

$$\xi = \frac{c}{Nr-c}.$$

The Eq. VIII-6 above, was developed [56] for use when there are N identical sources with identical parameters. Later in this chapter the problem where the sources have different parameters will be discussed. Also, Eq. VIII-6 was developed for the two-state traffic model and must be modified to work with the three-state model from this research.

Extension to Three-state model

The approximate time constant of the three-state model can be found using the characteristic that the model produces "bursts" separated by silence periods from the longOff distribution. This means it is possible to represent the on and off rates in Eq. VIII-6 by approximating,

μ^{-1}_{VIII-6}= E[burstLength],

and

λ^{-1}_{VIII-6}=E[longOff]

where E[burstLength] is defined in (Eq. V-2). During these periods of activity the source is not constantly on but rather going between the on and shortOff states very rapidly. The average rate at which the source generates traffic while in this bursting state is,

$r_{avg} = U_{bursting} * \text{portSpeed}$

where,

$$U_{bursting} = \frac{\lambda_{shortOff}}{\lambda_{shortOff} + \mu}$$

as defined in Chapter V. Consequently, r in Eq. VIII-6 is set to r_{avg}.

Simulation Results

In the following section, the usefulness of equation VIII-6 will be assessed by comparing it to the results of discrete event simulations that model the traffic sources, the trunk, and the buffer.

To perform the simulation, a discrete event program was constructed to simulate a buffer being fed by these traffic sources. The buffer was 128 kbits, a reasonable size relative to the trunk size used in this example. The trunk speed was varied from 200 kbits/s to 400 kbits/s, and the percent discarded was recorded for both the 2 and 3 state cases which corresponds to the prob(discard). (Note: The "fluid-model" approximation used here assumes that the packets under consideration are small enough relative to the buffer capacity so that the prob(overflow) for a buffer is the same as the ratio of the dropped packets to the total packets.)

The graph in Figure VIII-7 compares simulation results for a blend of 15 statistically identical sources to Eq. VIII-6. A total of four lines are plotted. The nearly vertical line on the left represents the result of simulating with the two state input sources. The purpose of plotting this line is to show how the two-state model can fail to model the burstiness and thus underestimate the trunk size required.

The second line is the simulation result with three-state sources. The parameters used for both this and the two-state case can be seen in Table VIII-1. Note that the E[on] and utilization are the same for the two models. The higher point-to-point variability in the curve of the three-state simulation curve at low percent discarded was caused by higher sample variance in this range.

Table VIII-1. Three-state and two-state source parameters used for comparison. Both have the same expected on time and utilization.

	Port speed	E[on]	E[longOff]	E[shortOff]	p
3-state source	64000	0.025	10.0	0.0285	0.993
2-state source	64000	0.025	0.10	0.10	no effect

To show the "worst case" where there is no buffer space, the stair-stepped line on the far right is plotted. This line is stair-stepped because there is a certain range of trunk size corresponding to an integer number of PVCs so that the probability of loss stays constant within the range.

The resulting curve from the (Eq. VIII-6) is also included in Figure VIII-7. This curve shows that the approximate equation is reasonably close to the three-state simulation results particularly when the prob(discard) is less than 1%. This is the range of most interest to the sponsor [60].

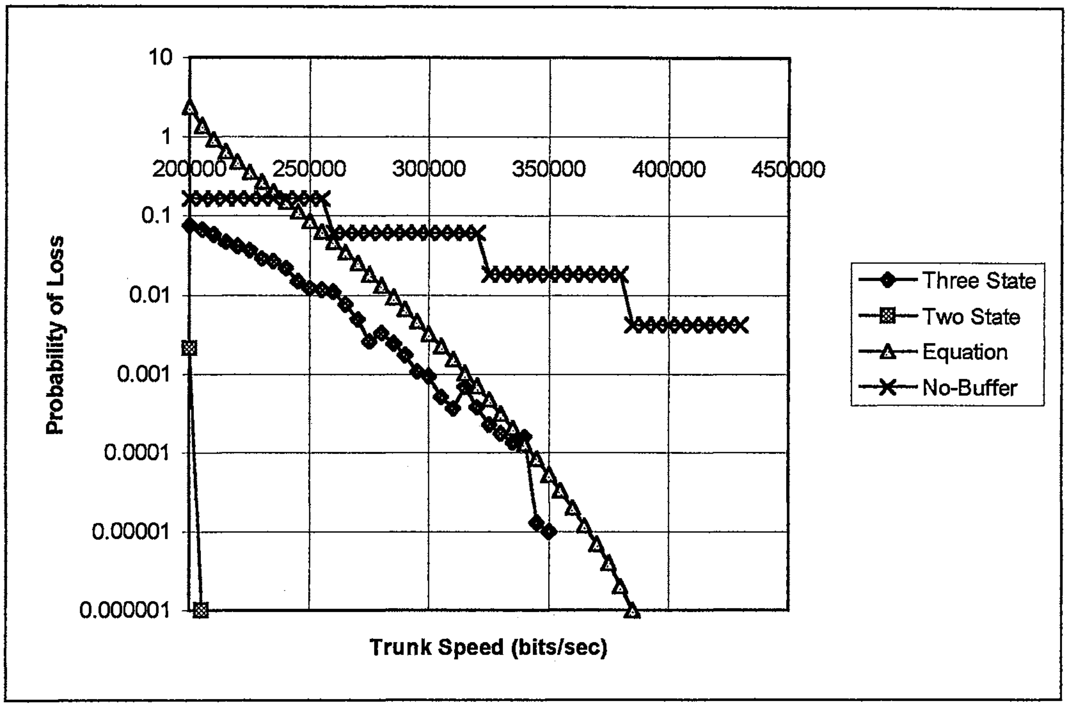

Figure VIII-7. Plot comparing simulations of two state (the straight up and down line on the left) and three state models along with the curve from the approximate equation. Fifteen sources were used and the buffer size was set to be 128kbits. The stair-stepped line to the right represents the worst case when the buffer size is 0.

The mixed case

For an approximate equation to be useful for real world applications, it must give informative results in the "mixed" PVC case. Cases where a switch is loaded with PVCs having identical model parameters, port speeds, and utilizations are rare. A heterogeneous mix of PVC model parameters is the most realistic scenario. Previously, (Eq. VIII-6) was compared to a simulation of the homogeneous case. Now, a similar equation will be compared to a mixture of PVCs with different model parameters.

The equation used above (Eq. VIII-6) is for the case where there are N PVCs with identical parameters. The equation was obtained by finding the dominant eigenvalue in the solution to a differential equation. The solution process involved finding the probabilities that i out of N of the sources were active [23,56]. For the mixed case, the problem is more difficult because there are many more unique combinations of active input sources. Each of these combinations produces a total input rate for which the steady state probability must be computed. Solving this problem is nearly impossible and no known simple solution exists, only approximations [56-58],[61].

The authors of [57] present such an approximate equation (Eq. VIII-7) for the prob(discard) when the sources are a mixture of different types. Their solution process assumes not only that the total number of each different PVC type is large, but also that the equation only needs to give accurate results when p(discard) is very small. In (Eq. VIII-7) below, z_0 is the exact equation for the dominant eigenvalue. The equation for the corresponding coefficient, β_0, is an approximation that has been shown to work reasonably well under most circumstances. The authors mention that in some cases, β_0 can tend to overestimate the prob(discard) particularly when the trunk loading is low.

$$G(x) \approx \beta_0 e^{-z_0 x} \qquad \text{(Eq. VIII-7)}$$

where,

$$z_0 = \frac{c - \Lambda}{\sum_{k=1}^{K} \Sigma_k^2 / \gamma_k}, \quad \beta_0 = \frac{\Sigma}{\sqrt{2\pi}(c - \Lambda)} e^{-z_0^2 \delta},$$

$$\Lambda = \sum_{k=1}^{K} \Lambda_k, \quad \Sigma = \sqrt{\sum_{k=1}^{K} \Sigma_k^2}, \quad \delta = \sum_{k=1}^{K} (\Sigma_k / \gamma_k)^2.$$

The terms Λ, Σ, and γ denote the mean, standard deviation, and time constant respectively of the output of the traffic model and K is the number of different types of sources (e.g. in Table VIII-2, K=4). The authors chose to use the mean, standard deviation, and time constant rather than the normal two-state model parameters, λ and μ, to make the equation more general.

For the three-state model developed in this research, the mean, variance and time constant of the traffic source can be written,

$$\Lambda_k = N_k \cdot U_k \cdot portSpeed_k,$$

$$\Sigma_k^2 = N_k portSpeed_k^{\ 2} \cdot U_k (1 - U_k),$$

and

$$\gamma_k = E[burstLength_k]^{-1} + E[longOff_k]^{-1}.$$

In these equations, N_k is the number of sources of type k, and U_k is the utilization of the k^{th} three state source. These equations come from the fact that the probability density function for the rate at which the source is generating traffic is just two impulses. One impulse is at zero and the other is at the port speed. The amplitude of the impulse at the port speed is equal to the utilization. Recall that in Chapter VI (Eq. VI-6), the decorrelation time-constant for the two-state model was shown to be $(\lambda_{shortOff} + \mu)$ = E[shortOff]$^{-1}$+E[on]$^{-1}$, which also corresponds to how the authors of [57] defined the time constant when using the two-state model. In the same manner that (Eq. VIII-6) was modified for the three-state case by setting λ^{-1}_{VIII-6} = E[burstLength] and μ^{-1}_{VIII-6}=E[longOff], the equation for the time constant, γ_k above, can be obtained. Recall that the justification for being able to do this is that the three-state model can be approximated as a two-state model with a "bursting" state and a longOff state. Simulations were performed to verify this assumption.

Table VIII-2. PVC parameters used in the "mixed" case. The there were a total of 23 PVCs in this case in four different classes. The buffer size was 256kbits.

Number	E[on]	E[longOff]	E[shortOff]	p
5	0.025	10	0.0285	0.993
5	0.05	12	0.01	0.994
7	0.1	22	0.075	0.99
6	0.036	6	0.1	0.99

Figure VIII-9 shows the graph comparing (Eq. VIII-7) and the simulation results using the values in Table VIII-2. From the graph it can be seen that the equation provides a conservative estimate of prob(discard).

In this figure (Fig. VIII-9) the curve from the equation overestimates the prob(discard) from the simulation by approximately a factor of ten. There are a few possible explanations. First, both (Eq. VIII-6) and (Eq. VIII-7) were derived under the assumption that the prob(discard) is very small; thus the equation should be more accurate as the prob(discard) approaches zero. It was not possible to simulate for much less than prob(discard) = 10^{-5}. Second, for the derivation of (Eq. VIII-7) it was assumed that there were a large number of sources, but for this example only 23 PVCs were simulated. A third possible reason for the inaccuracy could be, as mentioned earlier, that the β_0 coefficient used in the mixed source case is off somewhat – the equation for this coefficient is only an approximation. The authors of [57] indicate that this coefficient can be off by more than a factor of ten. They also indicate the problem is most pronounced when the trunk utilization is low (less than 60%). For the example shown in Figure VIII-9, the utilization decreases from about 90% when the trunk speed is 900 kbit/s to about 60% when the trunk size is increased to 1.50 Mbit/s.

It can be seen that this factor of ten difference in the prob(discard) produces an error of approximately 15% when using the equation to estimate the trunk size necessary to attain a given level of performance. If the network designer were to use the equation, as plotted in Figure VIII-9, to determine the trunk size necessary to attain a prob(discard) of $1\text{x}10^{-4}$, then he would infer that a trunk size of approximately 1.55 Mbit/s is necessary to achieve this performance. Using the simulation curve instead shows a trunk size of only 1.32 Mbit/s is necessary to achieve the desired level of performance. These values differ by 17.4%.

Another example, shown in Figure VIII-10, shows how Eq. VIII-7 is affected by different buffer sizes. The parameter values used to create this plot are shown in Table VIII-3. In this plot, there are two equation curves: one for a buffer size of $1\text{x}10^5$ bits, the other for $5\text{x}10^5$ bits. Simulation points were generated for each of the two buffer size cases. This plot shows that for the case of the buffer size of $5\text{x}10^5$ bits, the simulation line appears to be approaching the theoretical curve as the prob(discard) gets smaller. However, for the $1\text{x}10^5$ bit buffer case, the simulation curve crosses the curve from Eq. VIII-7 effectively showing that in certain conditions the equation is not a reliable method of predicting the prob(discard). This problem will be addressed later in this section. In this plot, the simulation points were obtained by averaging the results of 4,000 simulation runs to reduce the simulation sample variance. This produced a smoother curve than those seen in previous plots.

Also plotted in Figure VIII-10 is the Gaussian approximation which, according to the literature, has been used to predict the prob(discard) for the mixed case [61]. In [61], the authors recommend using this Gaussian approximation since, as mentioned above, no closed form equation exists for the prob(discard) of the mixed case. (Recall that (Eq. VIII-7) proposed in [57] is merely an approximation.) This Gaussian approximation curve is plotted by using the mean and variance for the aggregate of the three-state sources. (This aggregate mean and variance is the same as the aggregate mean and variance computed as part of (Eq. VIII-7).) The prob(discard) is then simply the probability that the aggregate traffic exceeds the trunk rate. (This is essentially a Gaussian approximation to the no-buffer case with the binomial distribution discussed earlier in this chapter.) Notice that this Gaussian approximation does not consider the burstiness of the traffic sources. It can be seen in the Figure VIII-10 that this curve is more conservative than the curve from (Eq. VIII-7). It appears that (Eq. VIII-7), when there is adequate buffering in the switch, does a better job predicting the prob(discard) than the Gaussian approximation. The authors of [56-58] confirm this fact.

Naturally, there should be some concern that Eq. VIII-7 underestimates the prob(discard) for small buffer sizes. This author believes the failure is caused by a breakdown, under certain circumstances, in some of the original assumptions used to derive approximate equations. In deriving these equations one of the main assumptions was that there would be a large number of sources feeding the buffer – essentially an assumption that a single PVC can't seriously affect the system. This assumption would be invalid if any of the following were true: the buffer size is small, there are a small number of sources, or some of the sources have a bit rate on the same order of magnitude as the trunk speed. Generally speaking, the actual network should not operate under these conditions. But, in the event that it does, recall that the leaky-bucket traffic policer tags excessively bursty traffic as being discard-eligible (DE) and so the traffic that will be lost during the periods of overflow will be the customers that are excessively bursty, not the customers who stay within their committed burst size (Bc).

Table VIII-3. Parameter values to generate Figure VIII-10.

Number	E[on]	E[longOff]	E[shortOff]	p
10	0.02	4	0.02	0.983
10	.014	5	0.01	0.965
10	.01	3	0.075	0.960
10	0.036	2	0.01	0.970

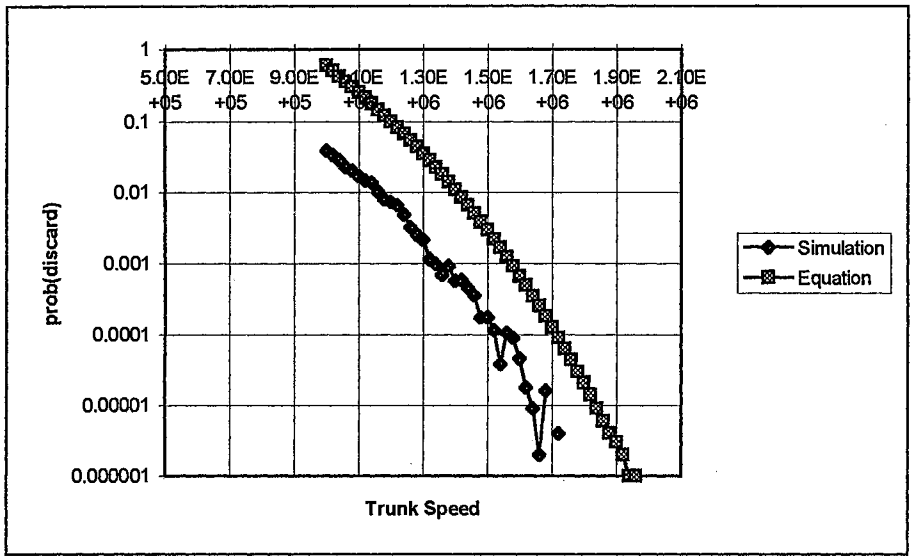

Figure VIII-9. Comparison of simulation and equation for "mixed" pvc case.

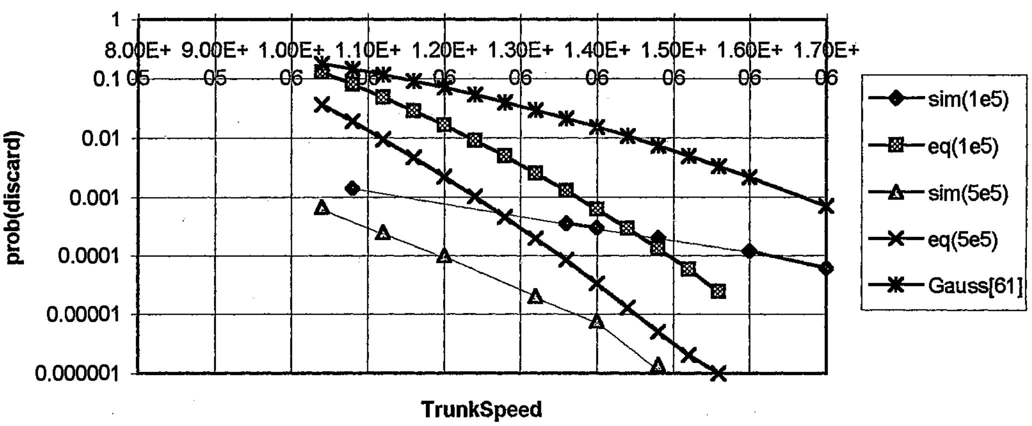

Figure VIII-10. Plot showing the effect of buffer size on probability of discard. This figure shows that Eq. VIII-7 can underestimate the prob(discard) when the buffer size is small.

Design Tool Implementation

The approximate solution to the prob(discard) for the "mixed" pvc case was implemented as part of the design tool that is being constructed in conjunction with this research. The prob(discard) on a trunk is one of the parameters critical to the customer's quality of service.

To compute the prob(discard) on a given trunk, it is necessary to determine the traffic model parameters for every PVC on that trunk. The steps necessary are described in various places throughout this research and are summarized below:

1) Compute the three-state traffic model parameters for each PVC using the statistics from the busy-hour via the techniques described in this research.
2) Route the PVCs using a routing algorithm that approximates the routing algorithm used in the actual network.
3) For each trunk, combine the PVC statistics, the trunk speed, and the buffer size using the "mixed" pvc case equations given in chapter VIII (Eq. VIII-7).

This was performed for the trunks of a working Frame Relay network. Because there was "live" traffic on the network, it was not possible to vary the trunk or buffer size and fully test the accuracy of the model. Decreasing the trunk size until packets were dropped and then comparing this to the equation would have been ideal. However, it was possible to obtain the prob(discard) from the equation and then verify that the network was not actually discarding a greater amount of traffic than the model predicted.

For example, applying (Eq. VIII-7) to one T1 trunk loaded with 113 PVCs showed an average load of 460 kbits / sec with a standard deviation of 594 kbits / sec. Using the computed model parameters shown in Appendix I , and a buffer size of 100 kbits, the equation showed that the prob(discard) was essentially zero ($2\text{x}10^{-77}$). The measurements of actual traffic collected during the time interval showed that there were, as predicted, no discards. If the trunk size were decreased by one-half to 750 kbits / sec, (Eq. VIII-7) predicts a prob(discard) of $8.2\text{x}10^{-7}$ and further decreasing it to 600 kbits / sec results in a prob(discard) of $6.31\text{x}10^{-2}$. It can be concluded that this trunk is underloaded by about a factor of two.

Summary

The current and somewhat crude method of sizing trunks based on an oversubscription rate was described. The problems with this technique were noted, namely the lack of a strong correlation between actual traffic and the total CIR on a trunk. It was concluded that the trunks should be sized based on measurement of the actual traffic. Next, the two cases of extremely large buffers and very small buffers were compared to see how the buffer size affected the prob(discard) in a Frame Relay switch. The effects of realistic sized buffers were then considered. Some approximate equations for finding the prob(discard) were provided by modifying equations in the literature to work with the three-state traffic model. A discrete event simulation was used to compare the equations to experimental results. For the cases tried, the approximate equations predicted the prob(discard) reasonably well. Except in cases where the buffer size was small, they did a better job of predicting than other methods proposed in the literature.

IX. Conclusion

This chapter will summarize the research done, the major findings and describe some areas for future research.

Review of Motivation

This research was motivated by the need for a comprehensive Frame Relay network design tool. The theory behind the operation of the tool comes from the research described in this paper. At the current time, providers of Frame Relay have no such tool available. Rather, the sizing of network trunks is done by making adjustments after problems occur. The results of this research allow the network engineering process to be more proactive. This tool will also result in monetary savings; it is believed that portions of the sponsor's network are over-designed and this tool will assist in identifying them. Additionally, the tool will yield improved overall network performance. That is, it will help the network designer more efficiently allocate network resources -- reducing, on average, the network delay and number of trunks the traffic must traverse as it flows across the network. It will do this by helping the network designer identify, via the routing simulations and estimating the load on each trunk, areas where trunk overloading could be occurring. Trunk overloading can cause traffic on those trunks to be dropped, and can also cause new PVCs to be diverted around the congestion resulting in a higher hop-count than would be necessary otherwise.

More specifically, this tool can be used to estimate the bandwidth requirements for each Frame Relay trunk by considering the busy-hour traffic characteristics of the virtual circuits routed on each trunk. Using the tool, the network designer will also be able to perform "what if" analysis on a model of the actual network by changing the layout and size of trunks and then rerouting the virtual circuits. In this manner, network "hot-spots" can be identified and resources in the actual network can be re-allocated or increased accordingly. The tool also has the ability to give an indication of the Quality of Service the customer, routed over a given set of trunks and switches, can expect by estimating the loading on those trunks and switches. Estimates for the probability of the customer's traffic being discarded can be made by considering the amount of discard eligible traffic they are transmitting and the loading on the trunks occupied by their PVC. By considering trunk loading, the number of trunks traversed by a given PVC, and the lengths of these trunks, estimates of PVC delay can also be obtained.

This work was funded by and is being done in cooperation with LDDS WorldCom of Tulsa, Oklahoma. This enables actual traffic measurements and realistic network topologies to be included in the research. The assumptions made in the research have been validated by comparing to actual traffic measurements. Further establishing the need for this research is the fact that Cascade Communications Corporation, a major vendor of Frame Relay switch equipment, has parallel interest in these issues and has been helpful in providing technical details and comments.

Work Done

This section will provide a summary of the research completed.

Background Research

The basics of Frame Relay were provided. In particular, the key components of Frame Relay that influence the research were detailed. The "Leaky-Bucket" traffic policer was discussed in detail because its behavior is useful for estimating the traffic patterns.

A discussion of the various traffic models that have been used in the literature has been included. In particular, the importance of representing burstiness in the model was discussed. The two-state on-off model was introduced as a starting point for developing the traffic model used in this research. During the course of this research, it was discovered that the two-state traffic model was not adequate and needed to be extended into a more detailed model with parameters that can be set to match the measurements of the actual traffic.

The method used to collect the actual traffic measurements from the network including a discussion of the Frame Relay Management Information Base (MIB) and how these measurements provided the average frame size, % DE frames and utilization was also provided.

Leaky Bucket Simulation Program

The operation of the simulation program, developed for this research, that modeled a traffic source, placed the messages into frames, and tagged the frames using the leaky bucket algorithm was described. Through the use of this simulation program, it was shown that the two-state traffic model could not effectively match the actual traffic measurements and that some modifications to the model were necessary.

Three-State Traffic Model

By modifying the two-state traffic model into one where the "off" state dwell time comes from a hyperexponential distribution rather than an exponential distribution, it was shown that the model could be configured to match the utilization, %DE, number of frames, and average frame size. Some concern existed about the effect of choosing the value of E[burstLength] somewhat arbitrarily. However, further investigation showed that the impact on the outcome, in terms of probability of loss, does not appear to be significant.

Numerical Approximation to Leaky Bucket

It was necessary to develop an approximation to the simulation program that could rapidly estimate the three-state traffic model parameters from the observed statistics and PVC parameters. This approximation was developed and performs adequately.

Discussion of Aggregate Traffic

The mean, variance, and autocorrelation of the aggregate of several of these three state models was investigated. An approximate equation for the variance of the sampled and integrated output of the three-state model was developed. In this discussion, the autocorrelation functions for the two-state and three-state models were compared.

Network Modeling Program

A software package was developed that models network nodes, links, and PVCs. It routes the PVCs in a manner similar to the actual routing algorithm. Combining this program with the information about the aggregate traffic above, estimates for the aggregate on each trunk can be obtained.

In recent months, the routing algorithm used in the Frame Relay network has been changed to one which considers both available bandwidth and "administrative" costs of the links. The tool was modified to reflect these changes.

Modeled the Frame Relay Switch

The behavior of the switch was modeled. This was done by assuming the input sources shared a common buffer space of finite size and the buffer is serviced at a fixed rate proportional to the trunk speed. Using a more complex model would have made it overly difficult to mathematically model the switch buffers. A discrete event simulation was constructed to simulate multiple three-state traffic models feeding such a switch. This simulation program was compared to mathematical equations to describe the operation of the switch.

Obtained a mathematical model for Frame Losses

By slightly modifying existing equations from the literature, some simple and useful equations were obtained. Under realistic conditions, the equations give conservative results for the probability of packet discard.

Future Research

Some areas for future research were identified during the course of performing this work. One area, identified in Chapter VI, is to further investigate the probability density function (PDF) of the total on-time of the three state model during some fixed time interval. It was sufficient to assume in Chapter VI that the PDF was Gaussian, but with further effort it might be possible to produce a more precise equation.

Another interesting project would be to correlate measurements of actual traffic to the type of customer using that particular PVC. With this information it might be possible to determine traffic characteristics for classes of customers, e.g. financial, data, internet provider, or voice. Also, knowledge of the time of day certain types of customers are most likely to use the network could be exploited to increase efficiency.

Finally, more mathematical analysis of the probability of discard when a heterogeneous mixture of three-state on-off models are sharing a switch buffer could yield an equation more accurate than the one used for this purpose in Chapter VIII.

Appendix I - Computed Statistics from a T1 Frame Relay Trunk

The following table contains the model parameters computed for PVC traffic measurements from one T1 trunk. In the cases where the PVCs did not have discard eligible (DE) traffic during the collection period, a two-state model was fit (these are the rows where p = 1).

E[on]	E[shortOff]	E[longOff]	p	PortSpeed
0.000718	0.1442092	0	1	1536000
0.000572	0.2602224	0	1	1536000
0.04894	5.6472622	0	1	56000
0.05032	7.0925374	0	1	56000
0.00173	0.0323663	0	1	1536000
0.029018	5.296426	0	1	56000
0.002262	0.00001	899.9046	0.97619	56000
0.001254	0.014817	0	1	384000
0	0	0	0	384000
0	0	0	0	384000
0.001462	0.0163988	0	1	384000
0.025104	0.9214874	103.0431	0.810682	384000
0.001368	0.0136413	0	1	384000
0.002156	0.7599102	0	1	384000
0.006174	0.2954349	0	1	128000
0.00196	0.2399107	0	1	384000
0.006309	6.4685115	0	1	384000
0.009298	0.808884	0	1	56000
0.003931	0.5682239	0	1	128000
0.010857	29.989143	0	1	56000
0.018269	26.452319	0	1	56000
0.013714	31.020768	0	1	56000
0.014542	11.67377	0	1	56000
0.013714	29.986286	0	1	56000
0.016752	5.9832476	0	1	56000
0.015802	9.8743077	0	1	56000
0.013714	29.986286	0	1	56000
0.010857	29.989143	0	1	56000
0.007528	0.0563912	16.84339	0.987216	384000
0.004526	0.9213995	0	1	384000
0.01105	0.4425793	0	1	384000
0.021353	0.1452518	0	1	384000
0.001083	0.83225	0	1	1024000
0	0	0	0	256000
0.000701	1.0507004	0	1	56000
0.016748	0.2223598	0	1	56000
0.020674	1.1302212	0	1	56000

0.013554	0.1462186	0	1	56000
0.00432	7.5587051	0	1	256000
0.00415	0.6518269	0	1	256000
0.005618	2.9844156	0	1	256000
0.00546	0.1910498	16.24265	0.960698	256000
0.005813	6.2010841	0	1	256000
0.004798	1.893936	0	1	256000
0.007425	4.1400405	0	1	256000
0.00562	0.1745704	27.05845	0.963962	256000
0.005448	1.3899013	0	1	256000
0.008669	0.380869	7.893024	0.922092	256000
0.000938	59.999063	0	1	256000
0.008988	59.991013	0	1	256000
0.005195	1.059894	0	1	256000
0.00702	9.8830903	0	1	256000
0.007753	49.992247	0	1	256000
0.005354	20.449192	0	1	256000
0.022601	0.026751	45.54414	0.99013	56000
0.021474	0.0238137	52.17218	0.990943	56000
0.011858	0.1426486	0	1	64000
0.015427	0.1838642	0	1	64000
0.014923	0.2086239	0	1	64000
0.00575	59.99425	0	1	64000
0.007951	0.3195604	0	1	128000
0.030042	7.4699583	0	1	56000
0.032633	11.359772	0	1	56000
0.025601	0.1587012	272.4598	0.96314	56000
0.001342	0.4184345	0	1	1536000
0	0	0	0	1536000
0.001419	0.0352022	0	1	1536000
0.030992	0.2688085	0	1	256000
0.000161	21.951059	0	1	1024000
0.00162	56.24838	0	1	1536000
0.000752	0.0862208	0	1	1536000
0.011029	7.6812788	0	1	256000
0.003705	0.0235276	40.05067	0.994554	384000
0.006283	4.7057592	0	1	384000
0.003742	1.1311889	0	1	256000
0.003085	0.7707755	0	1	256000
0.000292	29.999708	0	1	1536000
0.000292	29.999708	0	1	1536000
0.00043	9.9995701	0	1	1536000
0.000292	29.999708	0	1	1536000
0.000374	14.28534	0	1	1536000
0.000485	8.7373788	0	1	1536000
0.000369	6.2496307	0	1	1536000
0.000418	10.111942	0	1	1536000
0.0048	0.7414684	0	1	256000
0.00929	0.4263339	0	1	56000

0.003909	0.165232	0	1	128000
0.0017	1.588406	0	1	1024000
0.000304	20.454242	0	1	1536000
0.000838	0.9991624	0	1	384000
0.000639	0.2458683	0	1	1536000
0.001592	0.1740178	0	1	1536000
0.00724	0.186767	0	1	56000
0.010685	7.0759291	0	1	56000
0.007343	0.4004507	0	1	56000
0.008233	2.0418809	0	1	56000
0.012664	12.663392	0	1	56000
0.007458	0.6303876	0	1	56000
0.009571	4.9628043	0	1	56000
0.007011	1.2659739	0	1	56000
0.007741	2.7529947	0	1	56000
0.006851	0.938527	0	1	56000
0.006895	1.0960459	0	1	56000
0.00674	0.5457459	0	1	56000
0	0	0	0	56000
0.038388	64.247327	0	1	56000
0.00752	0.8084367	0	1	56000
0.009289	5.8348664	0	1	56000
0.013154	13.62321	0	1	56000
0.016794	20.913439	0	1	56000
0.008652	0.5296253	0	1	56000
0.012602	0.5938666	0	1	56000

References:

[1] William H. Gates, The Road Ahead, Viking, New York, 1995.

[2] see http://www.disney.com on the World Wide Web.

[3] Andrew F. Inglis, Electronics Communications Handbook, McGraw-Hill, New York, 1988.

[4] Uyless Black, Frame Relay Networks: Specifications and Implementations - Second Edition, McGraw-Hill, New York, 1996.

[5] Leonard Kleinrock, "On the Modeling and Analysis of Computer Networks", Proceedings of the IEEE, Vol. 81, No. 8, August 1993.

[6] Aaron Kirshenbaum, Telecommunications Network Design Algorithms, McGraw-Hill , New York, 1993.

[7] Mario Gerla, Jose Augusto Suruagy Monteiro, and Rodolfo Pazos, "Topology Design and Bandwidth Allocation in ATM Nets", IEEE Journal on Selected Areas in Communications, Vol 7, No. 8, October 1989.

[8] Aaron Kershenbaum, Parviz Kermani, and George A. Grover, "MENTOR: An Algorithm for Mesh Network Topological Optimizations and Routing", IEEE Transactions on Communications, Vol 39, No. 4, April 1991.

[9] Clyde L. Monma and Diane D. Sheng, "Backbone Network Design and Performance Analysis: A Methodology for Packet Switching Networks", IEEE Journal on Selected Areas in Communications, Vol SAC-4, No. 6, September 1986.

[10] Averill M. Law and Michael G. McComas, "Simulation Software for Communications Networks: The State of the Art," IEEE Communications Magazine, March 1994.

[11] William Stallings, ISDN and Broadband ISDN with Frame Relay and ATM.- Third Edition, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1995.

[12] Cascade Communications Company, Network Services Technology Overview, Part Number 104-00016-00, November 1994.

[13] Vern Paxson, "Empirically Derived Analytic Models of Wide-Area TCP Connections," IEEE/ACM Transactions on Networking, Vol. 2, No. 4, August 1994.

[14] David W. Petr, Victor S. Frost, Nynn A. Neir, Ann Demirjis, and Cameron Braun, "Simulation Comparison of Broadband Networking Technologies," Simulation, January 1995.

[15] Vern Paxson and Sally Floyd, "Wide Area Traffic: The Failure of Poisson Modeling," IEEE/ACM Transactions on Networking, June 1995.

[16] Kerry W. Fendick, Vikram R. Saksena and Ward Whitt, "Dependence in Packet Queues," IEEE Transactions on Communications, Vol. 37, No. 11, November 1989.

[17] Messaoud, Laddada, "Delay/Loss Analysis for Virtual Networks with Bursty Bandwidth" Ph.D. Dissertation, Oklahoma State University, Stillwater, December 1994.

[18] Victor S. Frost and Benjamin Melamed "Traffic Modeling for Telecommunications Networks" IEEE Communications Magazine. March 1994.

[19] G. D. Stamoulis, M. E. Anagnostou, and A. D. Georgantas, "Traffic Source Models for ATM Networks: A Survey," Computer Communications, June 1994.

[20] K. Sam Shanmugan and A. M. Breipohl, Random Signal - Detection, Estimation and Data Analysis, John Wiley and Sons, New York, 1988.

[21] Donald Gross and Carl M. Harris, Fundamentals of Queueing Theory, John Wiley and Sons, New York, 1974.

[22] Leonard Kleinrock, Queueing Systems - Volume II: Computer Applications, John Wiley and Sons, New York, 1976.

[23] D. Anick, D. Mitra, and M.M. Sondhi, "Stochastic Theory of a Data-Handling System with Multiple Sources," The Bell System Technical Journal, October 1982.

[24] Ferit Yegenoglu and Bijan Jabbari, "Modeling of Aggregated Bursty Traffic Sources in ATM Multiplexers," IEEE Conference Name Unknown, ISBN# 0-7803-0950-2/93, 1993.

[25] Harry Heffes and David M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," IEEE Journal on Selected Areas of Communications, Vol. SAC-4, No. 6, September, 1986.

[26] Boris Igelnik, Yaakov Kogan, Vladimir Kriman and Debasis Mitra, "A New Approach for Stochastic Fluid Models of Multiplexers with Heterogeneous Sources," Queueing Sytems Vol. 20 (1995), pp. 85-116.

[27] Daniel Gillespie, Markov Processes - An Introduction for Physical Scientists, Academic Press, Boston, 1992.

[28] R. Ramakumar, Engineering Reliability - Fundamentals and Applications, Prentice Hall, New Jersey, 1993.

[29] M. L. Chaudhry, A First Course in Bulk Queues, John Wiley and Sons, New York, 1983.

[30] C. Heckart, B. Gage, "Voice Gets Framed," Byte Magazine, Vol. 21, No. 2, February 1996.

[31] Brogan, Modern Control Theory - Third Edition, Prentice Hall, New Jersey, 1991.

[32] William J. Beyda, Basic Data Communications - A Comprehensive Overview, Prentice-Hall, New Jersey, 1989.

[33] Private Conversations with WorldCom Engineers Mike Johnson, Loren Lovell, and Chris Painter, September 1995.

[34] Daniel L. Babcock, Managing Engineering and Technology, Prentice-Hall, New Jersey, 1991.

[35] Tomy M. J. NG and Doan B. Hoang, "Joint Optimization of Capacity and Flow Assignment in a Packet-Switched Communications Network," IEEE Transactions on Communications, Vol. COM-35, No. 2, February 1987.

[36] Vikram R. Saksena, "Topological Analysis of Packet Networks", IEEE Journal on Selected Areas in Communications, Vol. 7, No. 8, October 1989.

[37] Krzysztof Debicki and Tomasz Rolski, "A Gaussian Fluid Model," Queuing Systems, Vol. 20 (1995), pp. 433-452.

[38] Milena Butto', Elisa Cavallero, and Alberto Tonietti, "Effectiveness of the "Leaky Bucket" Policing Mechanism in ATM Networks", IEEE Journal on Selected Areas in Communications, Vol. 9, No. 3, April 1991.

[39] Tetsuya Takine, Tatsuya Suda, Toshiharu Hasegawa, "Cell Loss and Output Process Analysis of a Finite-Buffer Discrete-Time ATM Queuing System with Correlated Arrivals", IEEE Transactions on Communications, Vol 43. No. 2/3/4, February/March/April 1995 pp 1022-1036.

[40] Hiroshi Yamada, "Cell/packet loss behavior in a statistical multiplexer with bursty input," Performance Evaluation Vol. 17 (1993), North-Holland.

[41] D. D. Botvich and N. G. Duffield, "Large Deviations, the shape of the loss curve, and economies of scale in large multiplexers," Queueing Systems Vol. 20 (1995) pp. 293-320.

[42] San-qi Li and Chia-Lin Hwang, "On the Convergence of Traffic Measurement and Queuing Analysis: A Statistical-Matching And Queuing (SMAQ) Tool," Available on Dr. Li's home page at the University of Texas.

[43] Chia-Lin Hwang and San-qi Li, "Link Capacity Allocation by Input Power Spectrum," Available on Dr. Li's home page at the University of Texas. Also presented at the 14th International Teletraffic Congress, June, 1994.

[44] Cheng-Shang Chang and Joy A. Thomas, "Effective Bandwidth in High-Speed Digital Networks," IEEE Journal on Selected Areas in Communications," Vol. 13, No. 6, August 1995.

[45] Wolfgang Fischer and Kathleen Meier-Hellstern, "The Markov-modulated Poisson process (MMPP) cookbook," Performance Evaluation, Vol. 18 (1993), pp. 149-171, North-Holland.

[46] Lennert Rade and Bertil Westergren, Beta Mathmatics Handbook, CRC Press, Boca Raton, 1990.

[47] Arthur W. Berger and Ward Whitt, "A Comparison of the Sliding Window and the Leaky Bucket," Queueing Systems, Vol. 20, pp. 117-138, 1995.

[48] The OSPF Specification, J. Moy Protean, Inc. October 1989. Available on the internet as RFC 1131 or RFC 1583 (newest version)

[49] Private discussions with WorldCom Engineer David Barmann, November, 1995.

[50] Will E. Leland, et al., "Self-Similarity in High Speed Packet Traffic - Analysis and Modeling of Ethernet Traffic Measurements," Statistical Science, Vol. 10, No. 1, Feb 1995.

[51] Darryl Veitch, "Novel Models of Broadband Traffic," Proceedings of IEEE Globecom (Houston), Vol. 2 pp. 1057-1061, 1993.

[52] N. Mitrou, S. Vamvakos, K. Kontovasilis, "Modelling, parameter assessment and multiplexing analysis of bursty sources with hyper-exponentially distributed bursts," Computer Networks and ISDN Systems, Vol. 27 (1995), pp. 1175-1192.

[52] Private conversations with Daniel T. Gillespie, author of Markov Processes ([27] above), February, 1996.

[53] Private conversations with Alan Noell, OSU Math department, March 1996.

[54] John M. McQuillan, Ira Richer, and Eric C. Rosen. "The New Routing Algorithm for the ARPANET," IEEE Transactions on Communications Vol Com-28 No. 5, pp 711-719. May 1980.

[55] M.H. MacDougall, Simulating Computer Systems - Techniques and Tools, The MIT Press Cambridge, Massachusetts 1987.

[56] Hwong W. Lee and Jon W. Mark, "Capacity Allocation in Statistical Multiplexing of ATM Sources", IEEE Transactions on Networking, Vol. 3, No. 2, April 1995.

[57] Hwong W. Lee, and Jon W. Mark, "Characterization of Statistical Multiplexing of Heterogeneous ATM Sources", Proceedings of IEEE Infocom, Vol. 1, pp 345-353, Boston MA, 1995.

[58] Guo-Liang Wu and Jon W. Mark, "Computational Methods for Performance Evaluation of a Statistical Multiplexer Supporting Bursty Traffic," IEEE Transactions on Networking, Vol. 4, No. 3, June 1996.

[59] Cheng-Shang Chang and Joy A. Thomas, "Effective Bandwidth in High-Speed Digital Networks", IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6, August 1995.

[60] Discussion with WorldCom network engineer Chris Painter, September 1996.

[61] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks," IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, September 1991.

VITA

Mark E. Allen

Candidate for the Degree of

Doctor of Philosophy

Thesis: TRUNK SIZING IN PACKET NETWORKS – THE EFFECT ON USER QUALITY OF SERVICE

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Kansas City, Kansas, On February 14, 1967, the son of Ray and Nancy Allen.

Education: Graduated from Liberal High School, Liberal, Kansas in May 1985; received Bachelor of Science in Electrical Engineering from the University of Kansas, Lawrence and Master of Science in Electrical Engineering from Oklahoma State University in May of 1990 and July of 1993 respectively. Completed the requirements for the Doctor of Philosophy degree with a major in Electrical Engineering at Oklahoma State University in May 1997.

Experience: Served as project engineer for LDDS WorldCom, of Tulsa Oklahoma, 1990 to 1993; researcher and teaching assistant at Oklahoma State University 1993 to 1996; and signals analyst for Raytheon E-Systems, Garland, Texas 1996 to present.

Professional Memberships: IEEE and HKN