ON THE PERFORMANCE OF PROTOCOLS TO SUPPORT INTEGRATED VOICE AND DATA SERVICES

by
Chin Yuan and John A. Silvester

Technical Report No. CRI 88-60

Computer Research Institute
University of Southern California
Los Angeles, CA 90089-0781

November 1988

Abstract

The good of integrated services networks is to provide various types of services, e.g., voice, data, video, etc., to the customer via a single interface. The key problems facing the network designer are how to build a networking fabric (local area and switched long-haul) to support the different demands of the traffic classes. In this thesis, we address two main topics: switch performance modeling and access protocol design.

In modeling switching systems, we first develop an exact model for delay-constrained voice traffic in a packet switching system. We study the influence of the delay constraint, talkspurt detection threshold and packet size on performance. We also consider a fairness algorithm that randomly discards late packets to improve performance. Next, we consider packet-switched integrated voice and data traffic. An exact model, which requires a two-dimensional Markov chain, is mapped by appropriate approximation into a one-dimensional Markov chain. The simplicity of this model reduces computation difficulty and memory limitation problems. We then extend this model to study integrated voice and data where voice traffic is delay-constrained. Following this, we study burst switching for integrated voice and data traffic using an approximate model. The exact three-dimensional model is decomposed into two two-dimensional Markov chains to simplify not only the model structure but also the analysis. Many of these models are necessarily approximate. The results are validated by simulation showing good agreement.

In the area of access protocol design, we develop a protocol for integrated voice and data services in a bus-type local area network. The protocol fulfills the traffic characteristics and service requirements of both types of traffic. An analytical model is proposed to study the system's performance. It is found that voice behaves like a TDM system and data performance is similar to a CSMA/CD system. In addition, high channel utilization is achieved due to the dynamic allocation of bandwidth between voice and data. A modified protocol to include fairness discarding algorithms for dropping late voice packets in a fair manner is then presented.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Historically, communication networks are developed for specific applications. Examples are the Public Switched Telecommunication Network (PSTN) for voice, the cable television network (CATV) for TV signal transmission, and various packet-switched networks (e.g., Ethernet) for data transmission. A customer needs to have separate access lines to each of these networks to use their services. This environment is expensive and inconvenient, which leads people to develop an integrated network that is expected to handle a variety of services and applications.

Offering various services in a single network promises several benefits: convenience, flexibility, and economy. An unified integrated user interface allows several services to be accessed by users simultaneously and services can be customized to individual needs without having to be concerned with the compatibility of different special-purpose networks. Furthermore, sharing facilities not only increases efficiency, but should also simplify network operations and maintenance, which yields cost saving for both the network provider and the user.

1.1 The Integration of Voice and Data

Voice and data are still the major traffic considered in the current integrated networks. One method to integrate voice and data communication is to digitize voice signals at the source and use a digital transmission fabric. Thus, voice signal is digitized and encoded into a bit stream, which has the same form as digital data. This , in turn, allows the same transmission technique to be used to provide integrated voice and data services for customers.

1.1.1 Traffic Characteristics and Service Requirements

To be able to integrate voice and data effectively, the characteristics of these two types of traffic and their service requirements must be understood. Voice is sensitive to absolute delay but can tolerate a relatively high error rate. In voice communication the delay must be small and uniform; otherwise voice quality will be degraded. On the other hand, in data communication, the absolute delay is not so critical but the transmission error rate must be negligible. Error detection or correction techniques are often used to ensure virtually error-free communication.

1.1.2 Switching Techniques

One of the important issues in integration of voice and data is the switching technique. Traditionally, voice is served by circuit-switched PSTN using analog signals and data is handled by packet-switched data networks using digital signals. Data traffic is bursty and delay-tolerable, which is well suited for packet switching; whereas, voice traffic is delay-sensitive and has traditionally been treated as continuous traffic, thus suitable for circuit switching. Owing to the better quality, higher efficiency and feasibility of simple signal processing of digital transmission, the telephone system is gradually converting from analog to an entirely digital network. This simplifies integration of voice and data through the same switching fabric. The advantage of using the same switching fabric is that a uniform transmission entity simplifies the switching operation.

Basically, we can categorize the current switching techniques for integrated voice and data services into four approaches:

- **Circuit Switching**
  Circuit switching [30] was developed primary for voice communication. The connection path between source and destination is established before the communication begins. Once the path is set up, it is dedicated to these two users until the communication finishes. The key characteristics of circuit switching (a dedicated channel between two users during the transmission period, no node-to-node error checking, etc.) provide a low delay environment for real-time services like voice transmission. The advantage of transmitting data using circuit switching is that the existing telecommunication systems can be used without much modification, i.e., low cost. However, data bursts are usually short and require only a brief connection, in addition, data is error-sensitive so that error checking or error recovery is necessary. These factors result in inefficient usage of the channel if circuit switching is used.

- **Packet Switching**

Packet switching [26,57] is based on the idea of message switching, or store-and-forward switching, in which is not necessary to establish a dedicated path between two users. Rather, if a station wants to send a message it appends the destination address to the message, the message is then passed through the network from node to node. At each node, the entire message is received, stored briefly, and then transmitted to the next node. The only difference between packet switching and message switching is that messages are divided into smaller segments of limited length, called packets, each with its own (packet) header. Unlike circuit switching, packet switching was designed particularly for data communication rather than voice communication. However, provided some case is taken to control delays, packet switching can also be effectively used for voice, with the resulting efficient usage of channel, flexibility of broadcasting or multicasting, and the capability of processing or altering speech information within the network.

- **Burst Switching**

  Burst switching [3,25] was originally designed for integrated voice and data services by GTE laboratories. The idea is to take advantage of the bursty nature of voice and data traffic. It is similar to message switching but is not store-and-forward, hence a burst can be forwarded before it is buffered completely. A Speech Activity Detector (SAD) is used to remove silence periods which improves the bandwidth utilization over traditional circuit switching. A burst is either a voice talkspurt, a data message or a control command. Typically, command bursts have highest priority, voice second priority, and data lowest priority, and bursts with higher priority are always served first. Instead of the conventional central control located at the switching center, e.g., end office in PSTN, burst switching partially distributes control to numerous small *link switches*, which accommodates network expansion and is less sensitive to node failures and overloads.

- **Hybrid Switching**

  The idea of hybrid switching [21] is to combine the features of circuit and packet switching in order to handle each type of traffic in the conventional manner. Voice and data traffic share a common link with either a fixed or movable boundary between allocated bandwidth. Different traffic is switched differently. Voice is handled by circuit switching and data is served by packet switching, respectively. Since the hybrid switching is more compatible with the present telephone system than packet or burst switching, it plays an intermediate role on the way to unified integrated voice and data switching.

1.1.3 Access Protocols

Another important issue for integrated voice and data services is the protocol to access the transmission media. As mentioned previously, customers currently have separate access mechanisms to different networks for different types of services. For integrated voice and data services, we need one access scheme which can satisfy both types of requested services. In the future PSTN, Integrated Services Digital Networks (ISDN) will play an important role to serve various types of traffic [55]. The basic subscriber loop in ISDN contains 2B+D channels. Each B channel has 64 Kbps capacity and can be used for either circuit-switching or packet-switching, the access protocol is X.25 if it is packet-switched. The D channel has either 16 or 64 Kbps capacity and is packet-switched. All traffic over the D channel employs a link-layer protocol known as LAP-D (Link Access Protocol - D channel).

Another type of network of interest, local area networks (LANs), are used in small geographical areas and require different access techniques since a single channel is shared by all users. Many data-oriented efficient channel access protocols to share this link exist: such as *random access protocols* [7,42,47,58,59,63] and *controlled access protocols* [9,35,52]. They are not very suitable for voice transmission due to its different traffic characteristics and service requirements. In order to efficiently integrate voice and data, we need to modify those protocols to fulfill the requirements of both traffic. Several researchers have addressed this issue and many protocols have been proposed for bus-type LANs [2,8,18,22,38,45,46,49] and Ring-type LANs [20,32,54,62]. Most of the proposed protocols are based on the CSMA/CD or Token-Ring protocols.

1.2 Thesis Outline

There are two main ways to compare the system performance of the different switching techniques: *simulation* and *analytical modeling*. Simulation can simulate any type of traffic and arbitrary complex switching systems, but it is time consuming. Analytical modeling can provide exact solutions under restrictive assumptions or for small systems. When the system size grows, problems of computation difficulty and memory limitation increase. To reduce these problems, approximation is used.

In this thesis, we develop an exact model for delay-constrained voice traffic and an approximate model for integrated voice and data traffic for packet switching in chapter 2 and 3. An approximate model to analyze integrated voice and data services for burst switching is proposed in chapter 4. In chapter 5, we present a protocol for integration of voice and data services in LANs, and develop an analytical model to measure the performance.

More specifically, chapter 2 concentrates on modeling delay-constrained voice traffic in a packet switching system. Two difficulties are encountered in analyzing this problem. One is that the non-Poisson packet arrival process is not as easy to model as the conventional Poisson process; the other is that the constant delay constraint makes the problem more complicated. We propose a two-dimensional discrete-time Markov model to solve this problem. Each user is modeled by an alternating renewal process and is described by the first dimension. The delay constraint is mapped into a fixed-size buffer and is represented by the second dimension. The influence of various delay constraints, talkspurt detection thresholds and packet sizes on the performance are analyzed.

In chapter 3, we develop an approximate model for integrated voice and data traffic in a packet-switched system. Most existing exact solutions are only for small systems due to memory constraints in computers. We map the problem into a one-dimensional Markov chain to describe the data behavior only, the effect of voice on data is approximated by voice's steady state behavior and is included in the transition probability of each state. We also extend this model for a system with delay-constrained voice traffic. The voice performance in such a system is measured by the model developed in chapter 2, the results of which are used to analyze the impact on data performance.

Chapter 4 emphasizes performance modeling of burst-switched integrated voice and data traffic. Since voice has *nonpreemptive* priority over data, unlike the packet switching model in the previous chapter, we cannot isolate voice from data, which results in a complex problem. In this chapter, we present an approximate model. We decompose the problem into two two-dimensional Markov models, each corresponds to one performance measure (for voice or data) of interest. The link between these two Markov chains is approximated by their steady state behavior. The relative frequency of fluctuations between voice and data traffic determines which Markov chain should be solved first. We consider examples for two types of data traffic: *interactive* and *file transfer* to demonstrate the prediction capability of our model.

Chapter 5 is devoted to the integration of voice and data in a bus-type local area network. In this chapter, we develop a protocol for integrated voice and data services and then propose an analytical model to study the system's performance. The protocol fulfills the requirements of both types of traffic and achieves high channel efficiency. Announcement and reservation techniques are used to resolve conflicts. The performance of voice traffic is much like a centrally-controlled TDM system and the data traffic behavior is similar to a CSMA/CD system. Finally, fairness discarding algorithms for voice traffic to distribute dropped packets over all active voice users are studied and a modified protocol to include the fairness algorithms is presented.

Chapter 6 concludes this thesis and addresses some issues for future research.

Chapter 2

Queueing Analysis of Delay Constrained Voice Traffic in A Packet Switching System

Voice packet switching has received great attention in recent years. Voice signals are digitized, encoded and packetized at each node. Unlike data traffic, the voice packet stream from a node has very high correlation between consecutive packets. In addition, in order for the speech to be properly reconstructed, a delay constraint must be satisfied. In this chapter, a queueing model which accurately predicts packet loss probabilities for such a system is presented. Analytical results are obtained from an imbedded bivariate Markov chain and are validated by a simulation program. Based on this model, the impact of the delay constraint, talkspurt detecting thresholds and packet sizes on packet loss is studied. Two schemes, *instant* and *random*, for discarding late packets are considered, and a comparison between them is made.

2.1 Introduction

Although voice traffic is usually handled via circuit switching, there is increasing interest in packet switched voice so that voice and data can be integrated over a common network. In a packet switching system, as shown in Figure 2.1, the speech signals are encoded, packetized and fed into a statistical multiplexer. A high speed communication link is connected to the multiplexer and serves the packets on a first-come-first-served (*FCFS*) basis. Most of these systems are equipped with *Speech Activity Detectors* (*SAD*) at each node such that packets are only generated while the node is in a talkspurt period. Since the typical mean duration of the talkspurt and silence period are about the same [6], the use of SADs can potentially double the channel utilization.

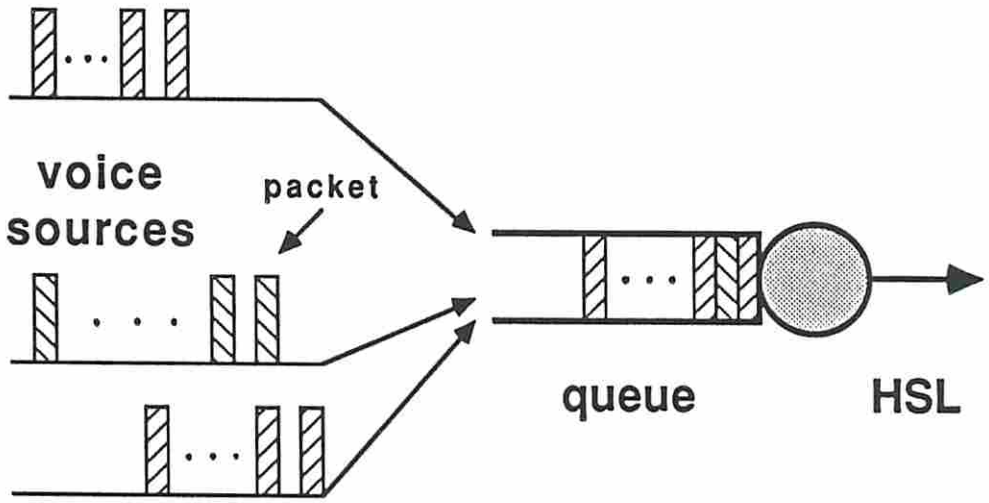


Figure 2.1: A packetized voice statistical multiplexer

For voice continuity and buffering constraints the delay must be kept small. Thus it is necessary to control link delay within some reasonable bound and discard the packets that will exceed the delay. If even a few percent of packets are delayed beyond that limit, voice quality will be seriously degraded [10]. Unlike packet data transmission systems, the average statistics such as mean queue length, mean packet delay, etc. are no longer suitable performance measures. In order to evaluate a voice transmission system, the maximum tolerable delay a voice packet can experience and the probability of packet loss should be taken into account. A model which can accurately predict packet loss probabilities under various delay constraints is needed.

Two major difficulties are found in analyzing this kind of voice transmission system. The first one is that we cannot assume that the arrival of voice packets is a Poisson process unless the population is large [29], since high correlation exists among stream-like packets from a voice node. To solve this problem, Stern [56] considered a *phase process* to represent talkspurt/silence state changes of voice nodes and modeled the system by an imbedded Markov chain in which the queue length is observed at the beginning of each "overload/underload" cycle based on this phase process. Daigle and Langford [11] followed the same idea and modeled the system by a bivariate Markov chain imbedded at instants of phase state changes, queue increments and queue decrements.

The second difficulty is how the delay constraint should be incorporated into the model. The papers mentioned above did not address this issue. In order to determine when a packet should be dropped, we need to remember exactly how long this packet has been in the system. Since the delay constraint is deterministic rather than exponentially distributed, the problem seems difficult. Fortunately, in a packet switching system packets are fixed length. It is obvious that the waiting time of a packet is just proportional to the queue length at the moment it arrives. By this observation, we can simplify the analysis to a finite queue size model, since packets that would exceed the delay constraint are immediately dropped.

In this chapter, we consider a finite buffer statistical multiplexer employing SAD for packetized voice traffic. Two schemes for selecting which packets to discard are considered. In section 2, a frame-based bivariate Markov model imbedded at the beginning of each frame is developed to analyze the performance of this system. The packet loss probabilities are readily found from this model. In section 3, we present some numerical results obtained from this model and find good consistency by comparing to simulation results. A comparison by simulation between two discarding schemes is also made.

2.2 The Model

Consider a system in which N active voice users (i.e., N voice calls) are multiplexed over a common link of capacity C packets per second. Assume each user alternates between a *talkspurt* period and a *silence* period. Fixed length packets are generated every F seconds from users in talkspurt and are fed into the multiplexer. The multiplexer stores the packets into a buffer and then transmits them over the common link on a FCFS basis. Since fixed packet length implies constant packet service rate, the system can compute how long an arriving packet will queue before being served, which is (approximately) equal to the packet transmission time multiplied by the number of packets already in the system. This property can be used to determine whether a packet should be discarded or not. Two schemes are considered here.

In the first scheme, a *buffer size* M is properly selected so that all packets in the buffer can be transmitted within their time constraint. A packet is discarded (or lost) when it arrives to find the system buffer full, we call this the "*instant discarding scheme*". Let D_t be the maximum tolerable delay a voice packet can experience in this multiplexer, then the buffer size is given by $M = \lfloor D_t C \rfloor$. Unfortunately, in this scheme, it is very likely that the next packet from a particular node will again be discarded if the current packet of that node was discarded, which implies that the node has a large packet loss even though the system-wide average packet loss probability is small.

Alternatively, the system temporarily stores every arriving packet into the buffer and determines the "*valid buffer*" every fixed interval instead of checking it at each arrival instant. The "*valid*" means that all the packets left in the buffer must meet their time requirement. At the checking point, the system runs a fairness algorithm to decide which recently arrived packets should be dropped in order to balance each node's packet loss probability. The algorithm we propose is that the system randomly selects a packet to drop from the arrivals during the interval, it repeats this process until the remaining packets all meet their time constraint (i.e., the buffer is *valid*). We call this the "*random discarding scheme*".

Remember that a user in talkspurt will generate a packet every F seconds, thus we let the checking interval be equal to F seconds and call this a *frame*. Let a *slot* be equal to the packet transmission time and γ be the number of slots in a checking interval. Thus at most γ packets will leave the buffer during a frame. Since the talkspurt activity of a particular user is correlated from frame to frame, a two-dimensional imbedded Markov model defined on each frame is used to analyze system behavior. Let $S_n = (t_n, b_n)$ be the system state at the beginning of the nth frame, where t_n is the number of voice users in talkspurt and b_n is the queue length. This discrete-time Markov chain can be characterized by the equilibrium state probabilities

$$\pi_{nm} = Pr\{t = n,\ b = m\}, \quad 0 \leq n \leq N,\ 0 \leq m \leq M$$

and the transition probabilities

$$p_{ij,kl} = Pr\{t_{n+1} = k, b_{n+1} = l \mid t_n = i, b_n = j\}, \quad 0 \leq i, k \leq N,\ 0 \leq j, l \leq M$$

2.2.1 Transition Probabilities

Changes in the queue length do not affect the number of users in talkspurt, so the behavior of t_n is governed by

$$t_{n+1} = t_n - u + v$$

where u is the number of users that switch from *talkspurt* to *silence*, and v is the number of users that switch from *silence* to *talkspurt*. Define $p_{10}(p_{01})$ to be the frame-independent probability that a *talking*(*silent*) user becomes a *silent* (*talking*) user at the end of the observed frame, then assuming exponentially distributed talkspurt and silence periods, we have [31]:

$$p_{10} = 1 - e^{-\lambda F}$$

$$p_{01} = 1 - e^{-\mu F}$$

where $1/\lambda$ and $1/\mu$ are the mean lengths of the talkspurt and silence period, respectively. We assume that a node can make at most one transition during a frame, i.e., a node that switches from talkspurt to silence will not enter into talkspurt state again in the same frame, and vice versa. The probabilities, Φ_{ij}, that j users will be in talkspurt in the next frame when i users are in talkspurt in the current frame, are thus given by

$$\Phi_{ij} = \sum_{u=(i-j)^+}^{min\{i,N-j\}} \binom{i}{u} p_{10}^u (1-p_{10})^{i-u} p_{01}^{j-i+u} (1-p_{01})^{N-j-u}$$

where $(x)^+ \triangleq max\{x, 0\}$.

The transition of the second state variable b is more complicated since a packet can get service immediately if it finds the server idle upon arriving, and it will be dropped if there is no space (scheme 1) or if the system decides to discard it at the checking point (scheme 2).

A. Scheme 1

Recall that M is the buffer size (excluding the one in service) and γ is the number of slots in a frame. The transition of b_n can be obtained by considering the following four cases:

Case 1 : $b_n \geq \gamma$ and $t_n \leq M - b_n + 1$

In this case, exactly γ packets are served in this frame and there is still enough space in buffer for the t_n packets that will arrive. Thus

$$b_{n+1} = b_n - \gamma + t_n$$

Case 2 : $b_n < \gamma$ and $t_n \leq M - b_n + 1$

Since $b_n < \gamma$, all b_n packets are served. Furthermore, some of the newly arriving packets may find an idle server before the end of this frame; they are also served during this frame. Define $\mathcal{R}$ to be the number of arrivals which can be served in this fashion, where $0 \leq \mathcal{R} \leq min\{\gamma - b_n, t_n\}$, Then, suppose $\mathcal{R} = r$, only $t_n - r$ packets will be left in the buffer thus:

$$b_{n+1} = t_n - r$$

with probability $\Psi_{\mathcal{R}}(r)$. A way to compute $\Psi_{\mathcal{R}}(r)$ is discussed in Appendix A.

Case 3 : $b_n \geq \gamma$ and $t_n > M - b_n + 1$

In this case, exactly γ packets can be served. Since $b_n + t_n - 1 > M$, it is possible that some of the new packets find no space in the buffer when arriving and are dropped immediately. Let $\mathcal{D}_1$ be the number of such packets, for a particular $\mathcal{D}_1 = d$, the resulting buffer length is

$$b_{n+1} = b_n + t_n - (\gamma + d)$$

with probability $\Psi_{\mathcal{D}_1}(d)$. Since the buffer provides $M - b_n + 1$ and $M - b_n + \gamma$ spaces at the beginning and the end of this frame, respectively, for t_n arrivals, the value of $\mathcal{D}_1$ will vary from $(t_n + b_n - M - \gamma)^+$ to $t_n + b_n - M - 1$, depending on the arrival instants of the t_n packets. In the Appendix A, we discuss how to find the probabilities $\Psi_{\mathcal{D}_1}(d)$.

Case 4 : $b_n < \gamma$ and $t_n > M - b_n + 1$

As in case 2, all b_n packets are served in the frame, but it is more complicated to determine how many packets will be left in the buffer at the end of the frame. Assume $\mathcal{R}$ is the number of packets being served as in case 2, and $\mathcal{D}_1$ is the number of packets being dropped as in case 3, then, given a pair (r, d), we get

$$b_{n+1} = b_n + t_n - (\gamma + r + d)$$

where $0 \leq r \leq min\{\gamma - b_n, t_n\}$ and $(t_n - r - M)^+ \leq d \leq t_n - r$. The function $\Theta_{\mathcal{R},\mathcal{D}_1}(r, d)$ which is the probability of each (r, d)-pair is discussed in Appendix A.

B. Scheme 2

Unlike the first scheme, the size of the *valid buffer* is not a constant but varies at the checking points. It depends not only on the arrival pattern but also on the dropped packet sequence. Again, four cases are considered:

Case 1 : $b_n \geq \gamma$ and $t_n \leq M - b_n + 1$

The transition of b_n in this case is exactly the same as case 1 of scheme 1:

$$b_{n+1} = b_n - \gamma + t_n$$

Case 2 : $b_n < \gamma$ and $t_n \leq M - b_n + 1$

As we discussed in the case 2 of scheme 1, there are $r \in \mathcal{R}$ packets, which can be served during this frame, the rest will be left in the buffer. Hence

$$b_{n+1} = t_n - r$$

with probability $\Psi_{\mathcal{R}}(r)$.

Case 3 : $b_n \geq \gamma$ and $t_n > M - b_n + 1$

Since $b_n \geq \gamma$, exactly γ packets can be served. Due to randomized arrival patterns, it is possible that the system has to drop some packets to get a valid buffer. The random discarding algorithm is now used to select the packets to be dropped. Let $\mathcal{D}_2$ be the number of packets which should be discarded, where $(t_n - M + b_n - \gamma)^+ \leq \mathcal{D}_2 \leq t_n - M + b_n - 1$, then

$$b_{n+1} = b_n + t_n - \gamma - \mathcal{D}_2$$

which is also the valid buffer size after discarding. For each specified $\mathcal{D}_2 = d$, the resulted $b_{n+1}(= b_n + t_n - \gamma + d)$ has probability $\Psi_{\mathcal{D}_2}(d)$. We discuss this probability in Appendix A.

Case 4 : $b_n < \gamma$ and $t_n > M - b_n + 1$

Again as in case 2, all b_n packets are served and some newly arriving packets, say r packets with probability $\Psi_{\mathcal{R}}(r)$, may find an idle server when arriving so that they can also be served during this frame. However, some of the remaining packets, say d and $0 \leq \mathcal{D}_2 \leq t_n - r$, might be discarded if they did not come at right time. Thus, given r and d, we have

$$b_{n+1} = b_n + t_n - (\gamma + r + d)$$

with probability $\Theta_{\mathcal{R},\mathcal{D}_2}(r, d)$., which is also discussed in Appendix A.

The transition probabilities $p_{ij,kl}$ of both schemes are thus given by

$$\textit{Case 1}: \quad p_{ij,kl} = \begin{cases} \Phi_{ik} & , \ l = i + j - \gamma \ \ (\overset{let}{=} c_1) \\ 0 & , \ \text{otherwise} \end{cases}$$

$$\textit{Case 2}: \quad p_{ij,kl} = \begin{cases} \Phi_{ik}\Psi_{\mathcal{R}}(i - l) & , \ (c_1)^+ \leq l \leq i \\ 0 & , \ \text{otherwise} \end{cases}$$

$$\textit{Case 3}: \quad p_{ij,kl} = \begin{cases} \Phi_{ik}\Psi_{\mathcal{Z}}(i + j - \gamma - l) & , \ M + 1 - \gamma \leq l \leq \min\{c_1, M\} \\ 0 & , \ \text{otherwise} \end{cases}$$

$$\textit{Case 4}: \quad p_{ij,kl} = \begin{cases} \Phi_{ik} \sum_{r\in\mathcal{R}} \sum_{d\in\mathcal{Z}} \Theta_{\mathcal{R},\mathcal{Z}}(r, d) & , \min\{c_1 - r, M + 1 - \gamma\} \leq l \leq c_2 \\ & \quad \text{where} \quad c_2 = \min\{c_1 - r, M\} \\ 0 & , \ \text{otherwise} \end{cases}$$

where $\mathcal{Z} \equiv \mathcal{D}_1$ for scheme 1 and $\mathcal{Z} \equiv \mathcal{D}_2$ for scheme 2.

2.2.2 Voice Discard Probability

Let π denote a row vector of state probabilities, $\pi = [\pi_{00}, \pi_{01}, \pi_{02}, \cdots, \pi_{NM}]$, and $\mathbf{P}$ denote a square matrix (NM × NM) containing the transition probabilities for the imbedded Markov chain described above. Then, we can obtain the equilibrium state probabilities by solving the equation

$$\pi = \pi \mathbf{P}$$

subject to

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \pi_{ij} = 1$$

Since more than one half of elements in transition matrix $\mathbf{P}$ are zero, we can use *Sparse Matrix* techniques [36] to reduce required memory and save computation time.

After obtaining the equilibrium state probabilities, we can evaluate the voice discard probability. Clearly, packet discarding occurs in case 3 and case 4 only. For a given state $S = (t, b)$, it is easy to determine the average number of discarded packets in this state, denoted by β, from

$$\beta = \begin{cases} \sum\limits_{d \in \mathcal{Z}} d \cdot \Psi_{\mathcal{Z}}(d) & \text{, if case 3} \\ \sum\limits_{r \in \mathcal{R}} \sum\limits_{d \in \mathcal{Z}} d \cdot \Theta_{\mathcal{R},\mathcal{Z}}(r, d) & \text{, if case 4} \\ 0 & \text{, otherwise} \end{cases}$$

where $\mathcal{Z} \equiv \mathcal{D}_1$ for scheme 1 and $\mathcal{Z} \equiv \mathcal{D}_2$ for scheme 2.

Then, we can compute the expected number of discarded packets, which is

$$\text{E[number of discarded packets]} = \sum_{t=0}^{N} \sum_{b=0}^{M} \beta \cdot \pi(t, b)$$

We can also find the expected number of arriving packets from

$$\text{E[number of arriving packets]} = \sum_{t=0}^{N} \sum_{b=0}^{M} t \cdot \pi(t, b)$$

and thus the average discard probability is

$$P_{dis} = \frac{\text{E[number of discarded packets]}}{\text{E[number of arriving packets]}}$$

2.3 Numerical Results

In this section, we study the performance of this statistical multiplexer for packetized voice transmission using the Markov model described in the previous section. We take a link

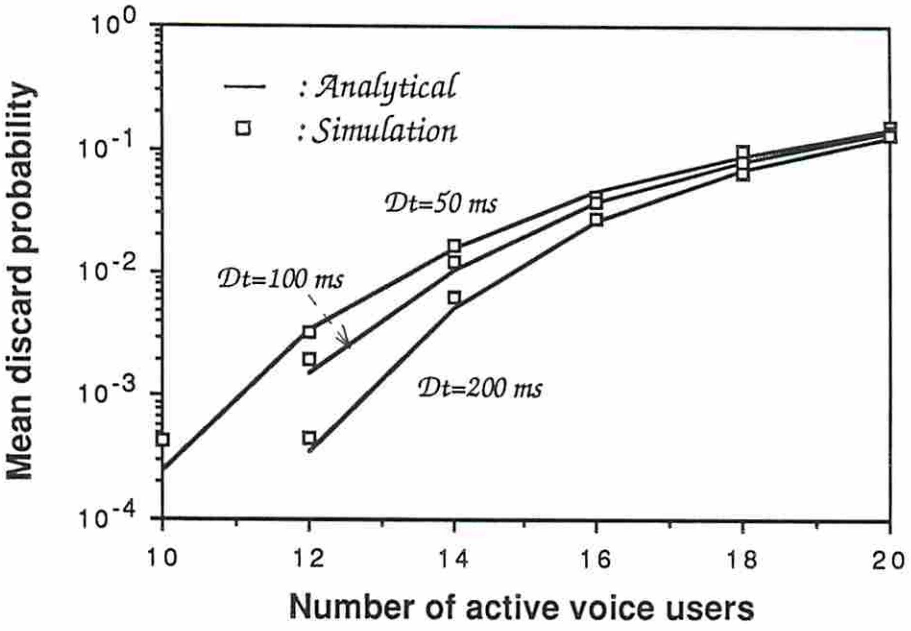


Figure 2.2: Mean packet discard probability vs. Number of active voice users (scheme 1, high detection threshold, packet size=1024 bits)

with capacity (C) of 500 Kpbs and assume all voice nodes have a 64 Kbits/s encoder to quantize the speech signal, thus we have eight slots in each frame. The mean talkspurt and silence durations in a voice call depends on the detection threshold in each speech activity detector (SAD), if hangover and fill-in techniques are used we have a longer talkspurt/silence period [24] [23]. Here, we use the numbers given in [11] [40], i.e., we set the average talkspurt/silence period to be 220/380 ms for low threshold, 440/560 ms for medium threshold, and 1350/1650 ms for high threshold. The maximum tolerable delay (D_t) for a voice packet was selected to be 50, 100 and 200 milliseconds. The major performance is the discarding probability for both schemes under different talkspurt/silence periods and delay constraints. To verify the accuracy of our model, we also present some simulation results.

Figures 2.2 - 2.4, respectively, show the discard probabilities against the number of talking users for scheme 1 under different delay constraints for the three different talkspurt detection thresholds.

Here, we assume the packet size to be 1024 bits. It is shown that the larger the delay constraint, the smaller the discard probability. Alternately, for a given discard probability, a larger delay constraint can support more talking users. Simulation results are also plotted for comparison; good consistency is found. Figures 2.5 - 2.7 show similar

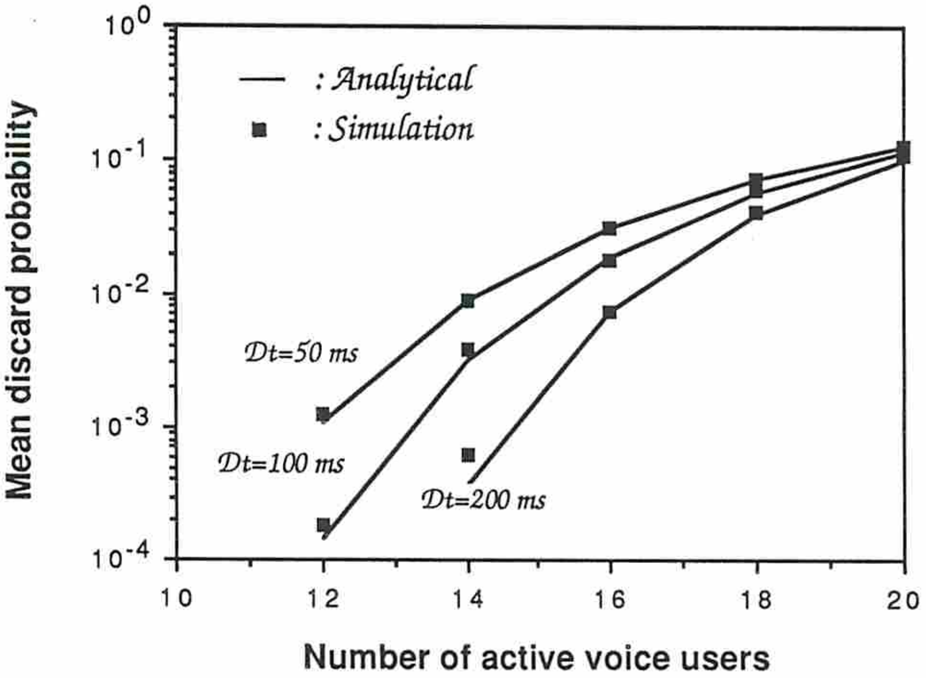


Figure 2.3: Mean packet discard probability vs. Number of active voice users (scheme 1, medium detection threshold, packet size=1024 bits)

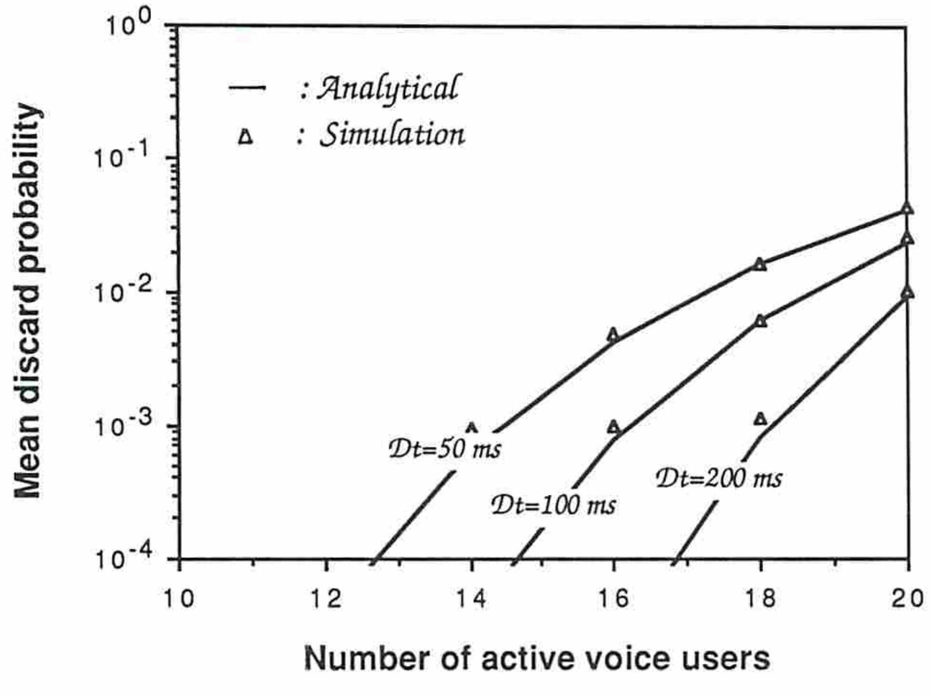


Figure 2.4: Mean packet discard probability vs. Number of active voice users (scheme 1, low detection threshold, packet size=1024 bits)

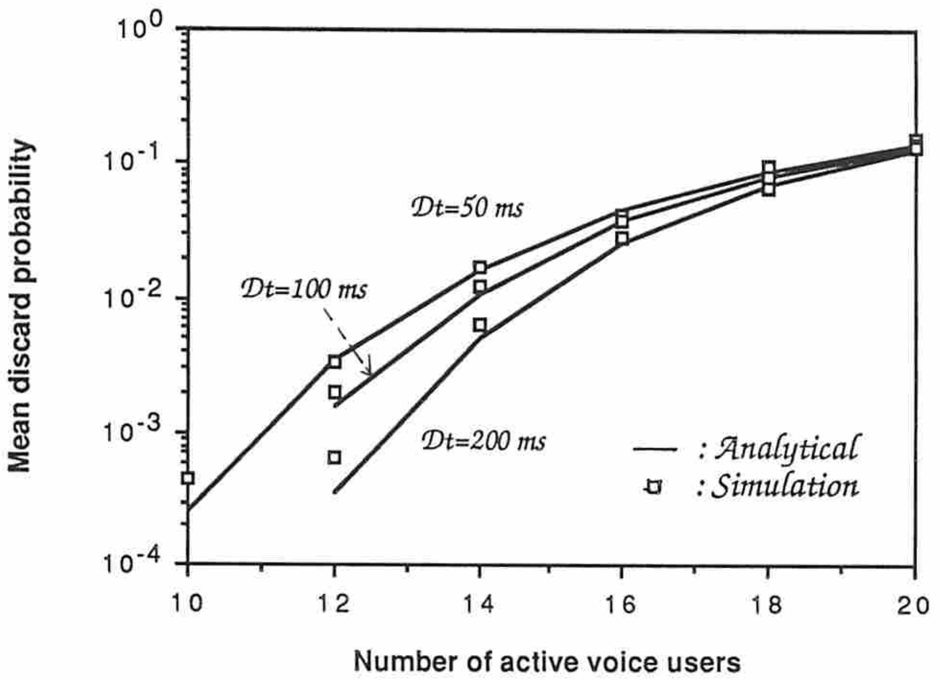


Figure 2.5: Mean packet discard probability vs. Number of active voice users (scheme 2, high detection threshold, packet size=1024 bits)

results for scheme 2.

In Figures 2.8 - 2.10, we compare the discard probabilities under a given delay constraint for the three detection thresholds.

It is found that the lower the threshold, the better the performance, i.e., the discard probability decreases. For example, when the delay constraint is 100 ms and the number of talking users is 18, the mean discard probabilities are 0.081, 0.058 and 0.006 from high to low threshold, respectively. We also compare the discard probabilities under various packet size assumptions for given detection thresholds, which are depicted in Figures 2.11 - 2.13.

Three packet sizes, 512 bits, 1024 bits and 2048 bits, are selected and the results show that the smaller packet size has better performance. However, smaller packet size means more header overhead, there is thus a trade-off here.

Next, we compare the two discarding schemes. The discard probabilities for the low threshold case are summarized in Table 2.1, where N is the number of talking users. We see that the mean packet discard probability of scheme 1 is slightly lower than that of scheme 2, but the difference is very small. In general, we can consider that both have the same average statistics.

We are also interested in the number of packets lost per talkspurt. We consider the

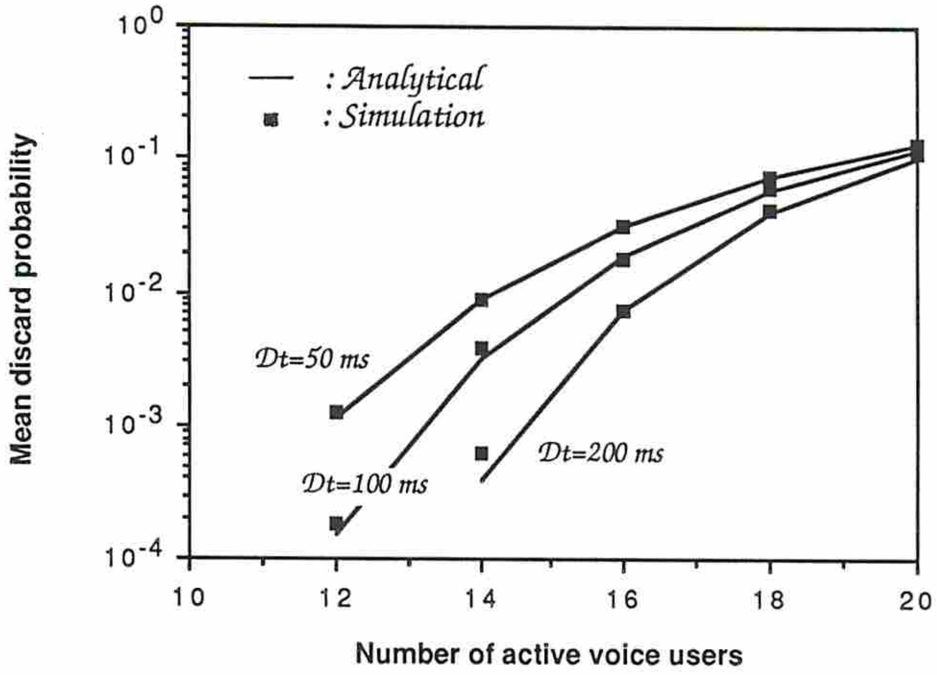


Figure 2.6: Mean packet discard probability vs. Number of active voice users (scheme 2, medium detection threshold, packet size=1024 bits)

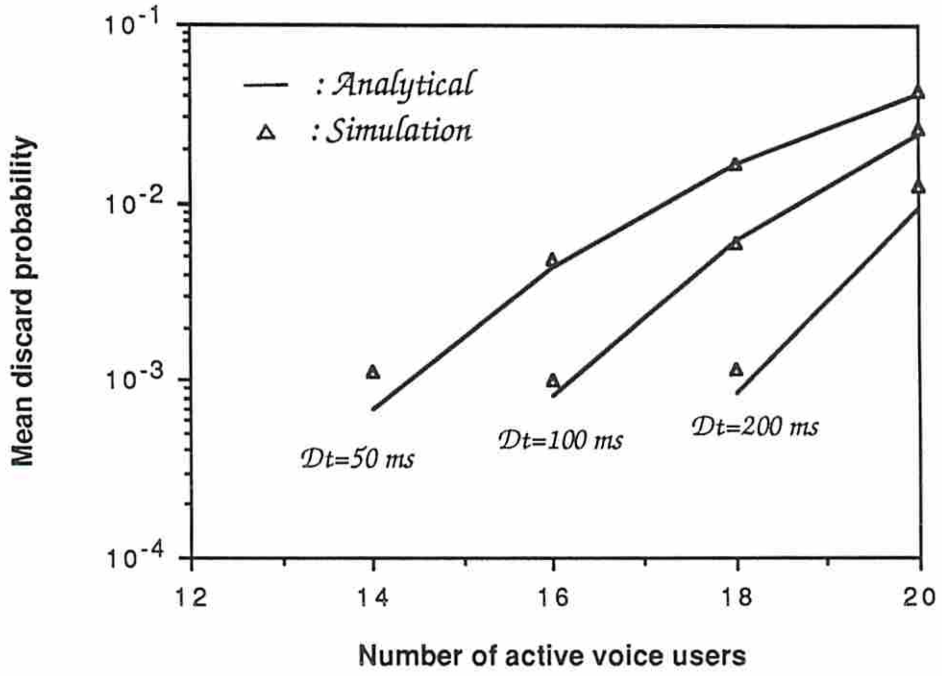


Figure 2.7: Mean packet discard probability vs. Number of active voice users (scheme 2, low detection threshold, packet size=1024 bits)

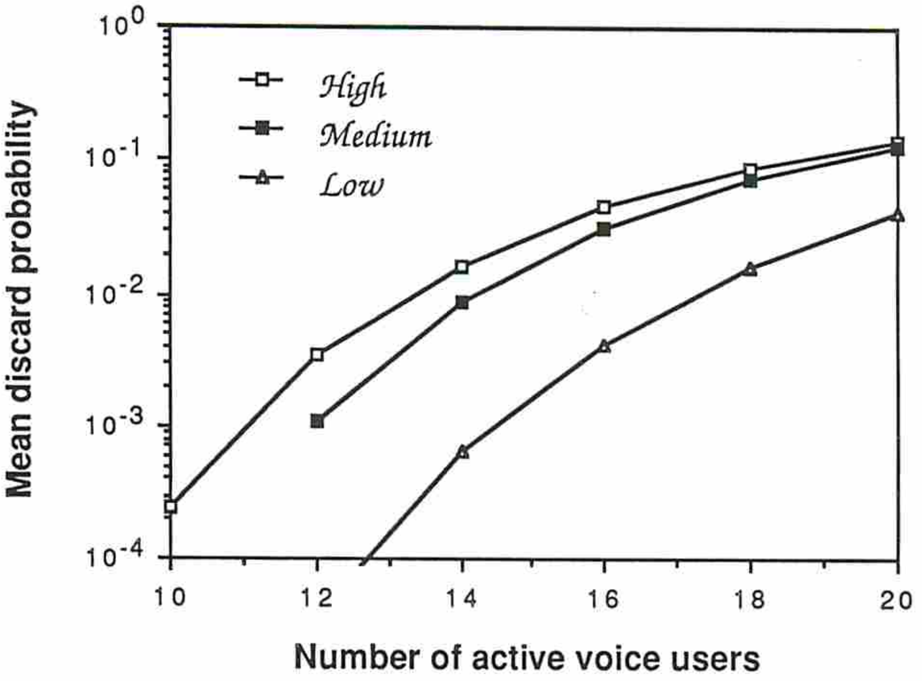


Figure 2.8: Mean packet discard probability vs. Number of active voice users (scheme 1, D_t=50 ms, packet size=1024 bits)

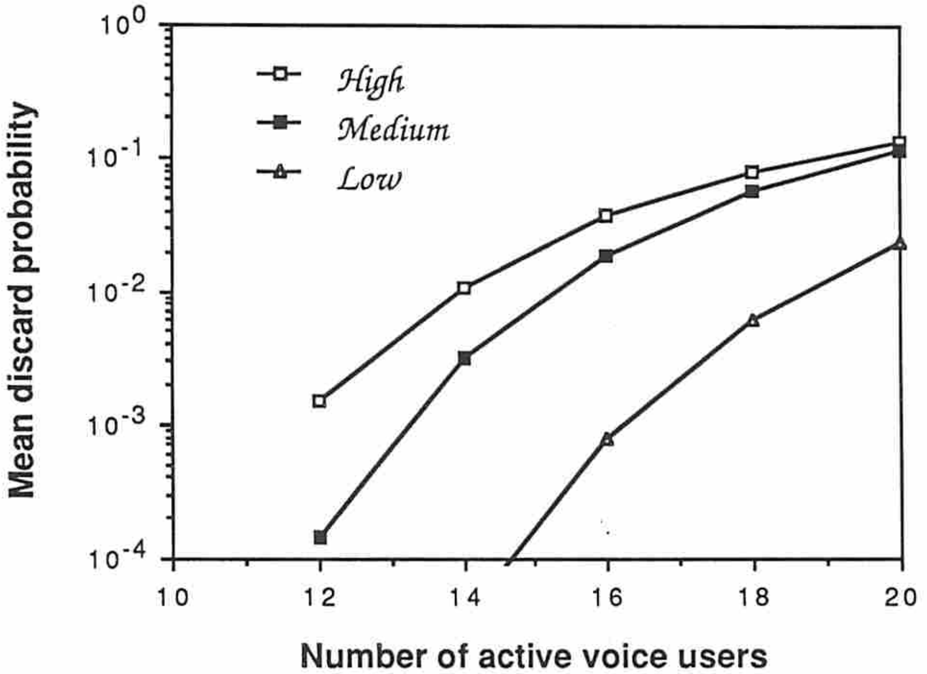


Figure 2.9: Mean packet discard probability vs. Number of active voice users (scheme 1, D_t=100 ms, packet size=1024 bits)

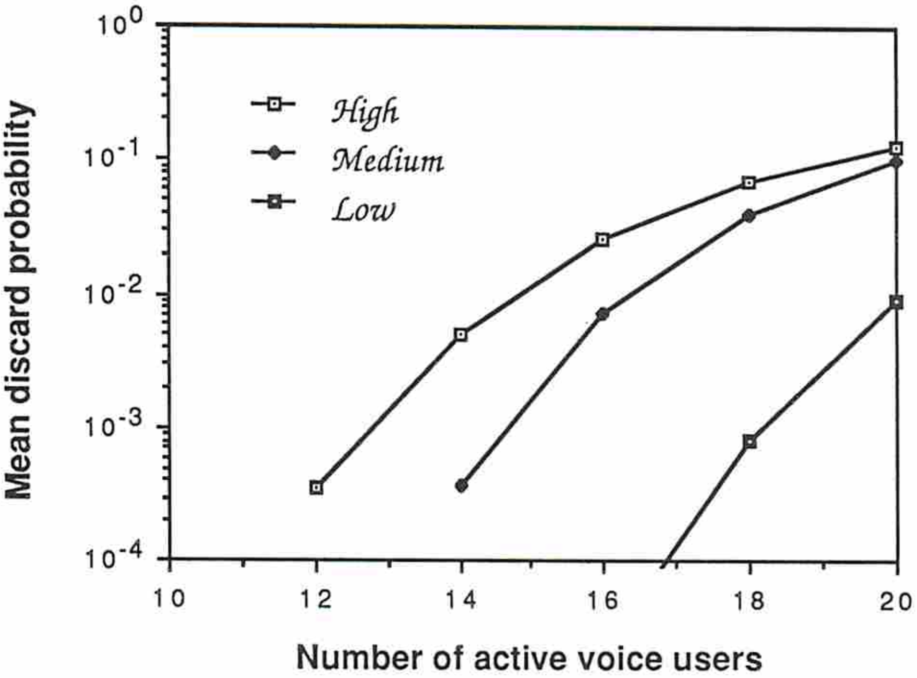


Figure 2.10: Mean packet discard probability vs. Number of active voice users (scheme 1, D_t=200 ms, packet size=1024 bits)

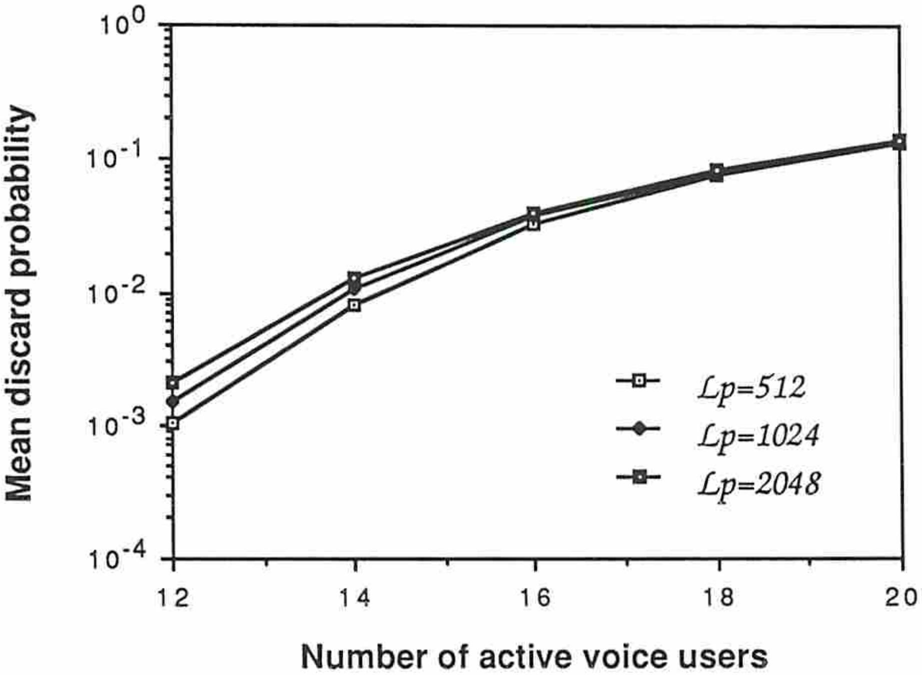


Figure 2.11: Mean packet discard probability vs. Number of active voice users (scheme 1, high detection threshold, D_t=100 ms, L_p=packet size)

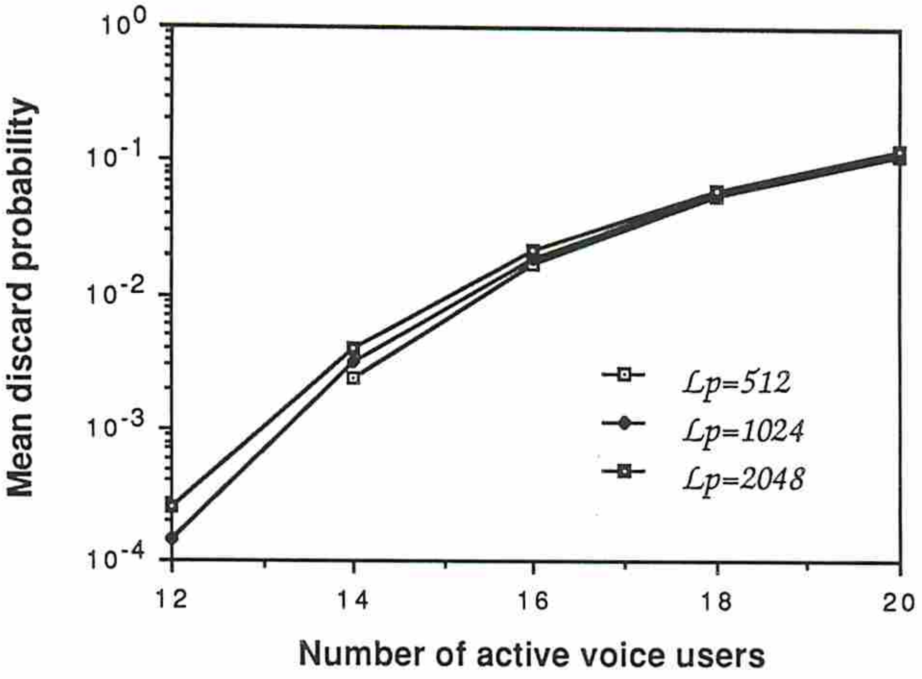


Figure 2.12: Mean packet discard probability vs. Number of active voice users (scheme 1, medium detection threshold, D_t=100 ms, L_p=packet size)

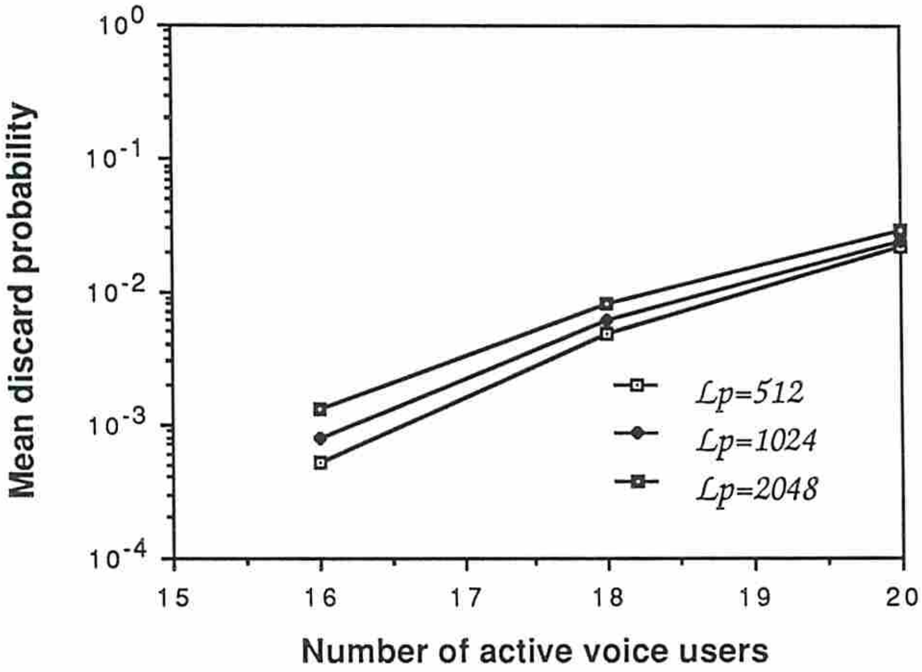


Figure 2.13: Mean packet discard probability vs. Number of active voice users (scheme 1, low detection threshold, D_t=100 ms, L_p=packet size)

D_t	N	Scheme 1	Scheme 2
50 ms	20	4.123e-02	4.182e-02
	18	1.623e-02	1.662e-02
	16	4.307e-03	4.488e-03
	14	6.380e-04	6.786e-04
	12	3.756e-05	4.102e-05
	10	2.357e-07	2.687e-07
100 ms	20	2.412e-02	2.450e-02
	18	6.045e-03	6.194e-03
	16	7.882e-04	8.208e-04
	14	3.824e-05	4.076e-05
	12	4.513e-07	4.947e-07
	10	3.558e-11	4.124e-11
200 ms	20	9.472e-03	9.570e-03
	18	8.265e-04	8.484e-04
	16	2.037e-05	2.136e-05
	14	7.664e-08	8.295e-08
	12	1.688e-11	1.903e-11
	10	0	0

Table 2.1: Mean Discard Probabilities of Scheme 1 & Scheme 2

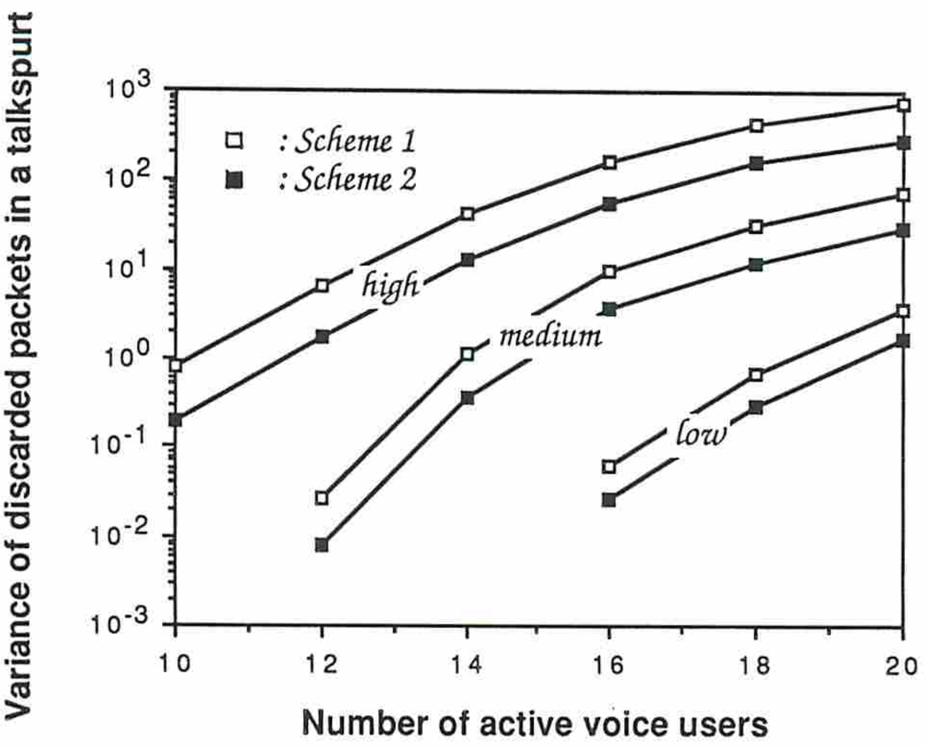


Figure 2.14: Comparison of scheme 1 & scheme 2 in variance of discarded packets per talkspurt (D_t=100 ms, packet size=1024 bits)

variance of the number of discarded packets per talkspurt for both schemes in Figure 2.14. We see that the variance of scheme 2 is always less than scheme 1, which is an indication that scheme 2 is fairer than scheme 1, since it spreads the lost packets over all concurrent talkspurts. In Figure 2.15 we set the average packet loss probability to be 0.01 and show the distribution function of the number of lost packets per talkspurt. We see that although scheme 2 has more talkspurts with some loss (k=0), it has much fewer talkspurts with loss of a lot of packets ($k \geq 2$). For instance, in this example, scheme 2 never loses more than 11 packets but there is a 0.2% chance to lose more than 11 packets in scheme 1. If the voice coding scheme is able to withstand loss of one or two packets, we have much better overall performance given by the random discarding scheme (scheme 2).

2.4 Summary

In this chapter, we develop a Markov model for analyzing delay constrained packetized voice traffic. By comparison to simulation results, it is demonstrated that this model can predict the system performance well. Numerical results are computed under various delay constraints, talkspurt detection thresholds and packet sizes. The results can be summarized as follows. First, a larger delay constraint can support more users since the

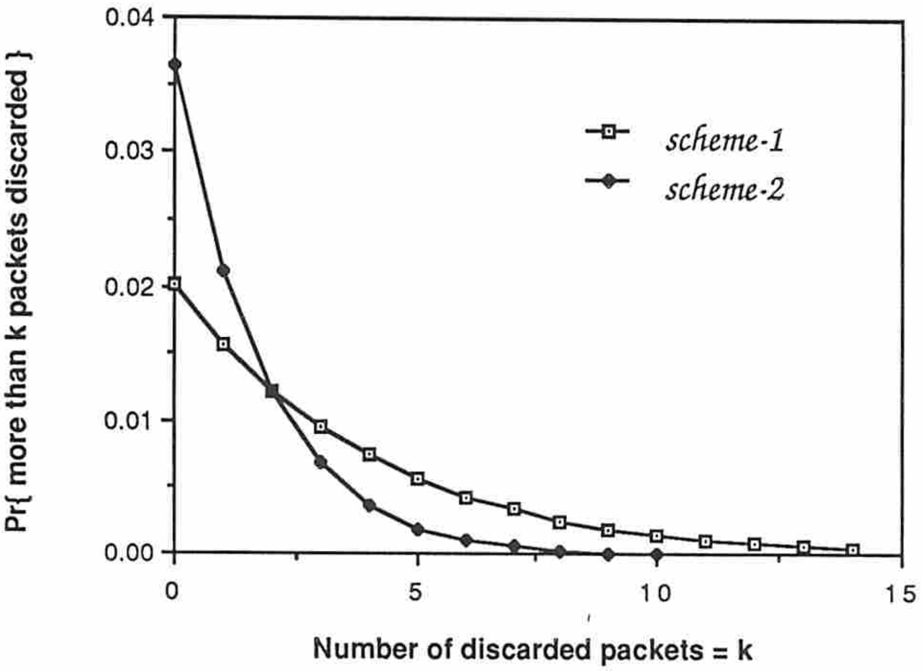


Figure 2.15: Comparison of scheme 1 & scheme 2 in probability of number of discarded packets per talkspurt (D_t=100 ms, packet size=1024 bits)

corresponding buffer size is larger so more arriving packets can be stored. Second, the lower detection threshold has better performance since the corresponding mean discard probability for a given delay constraint is smaller. Third, given a detection threshold, the smaller packet size has lower discard probability but the percentage of header overhead in each packet increases. Two schemes for discarding the over-aged packets are considered. The first scheme does not need extra buffer space to store arriving packets in a frame but has the effect of continually dropping packets from the same user. On the other hand, the second scheme needs extra buffer space to store arriving packets and perhaps some processing capability to select discarded packets but is fairer in that discarded packets are spread across all users. From the simulation results, it is shown that better performance can be obtained by using this latter scheme which we call *random discarding*.

Chapter 3

Performance Modeling of Integrated Voice/Data Traffic in Packet Switching

One candidate switching technique for integrated voice and data services is packet switching. Compared with circuit switching, packet switching is more flexible and efficient, at the expense of increased processing and queueing delays. Since voice is usually given priority over data, its performance can be studied by ignoring data, as was discussed in the previous chapter. On the other hand, data analysis is more difficult since it is affected by the voice traffic. In this chapter, we propose a simple approximate model to measure data performance. The influence of *delay-constrained* voice traffic on data is also studied via the voice model described in chapter 2 and data model proposed in this chapter.

3.1 Introduction

In an integrated voice and data packet switching system, both types of traffic share a common transmission link by using of a time division multiplexing technique. Transmissions are performed on fixed-duration frame basis and each frame is partitioned into two parts. One part is dedicated to data traffic only and the other part is shared by voice and data traffic but voice has priority over data. The boundary between the two traffic types within a frame is "movable" since data traffic is allowed to use any frame capacity temporarily unused by voice traffic. Speech Activity Detectors (SAD) are usually employed to increase the channel utilization and support more voice users. Thus, the system must provide buffers to store voice and data packets when no capacity is available upon arrival. As mentioned in the previous chapter, a voice packet can be kept in the buffer only limited amount of time. On the other hand, a queued data packet can stay in the buffer until it is served.

Since voice has priority over data, the performance of voice traffic can be obtained without considering the data traffic. On the other hand, the performance of the data traffic has proven very difficult to obtain, since we have to know voice statistics while analyzing data. There are several approaches that can be used to find the data performance for such a packet switched TDM link. The first approach is simulation. Although this technique is useful for any complex problem, the disadvantage is that it usually consumes a large amount of computer time to give accurate results. The second approach is to model this system as a (two-dimensional) Markov chain and find the "exact" solution [16,14,61]. Once the model is defined, we can either solve the resultant state equations or use a generation function technique to find the performance. In practice, however, significant computational difficulties and memory requirements limit this approach to small systems only. The third approach is to use approximations to simplify the problem. The quasi-static approximation [4,19,15] removes the correlation effect of voice arrivals, which is too optimistic. The fluid/diffusion approximation [13,39] tries to estimate the voice correlation effect, the results are better. In this chapter, we introduce a new approximate model which reduces the problem to a one-dimensional Markov chain. The analysis is thus greatly simplified.

In section 2, we analyze the voice performance by its steady state behavior with the assumption that a voice arrival can stay in the buffer only until the beginning of the next frame, which greatly simplifies the analysis. Data performance is measured by our proposed approximate one-dimensional model. In section 3, we consider a more general delay-constraint on the voice traffic which allows a voice packet to stay in the buffer a predetermined time. Using the model described in chapter 2, we can find the voice performance. The results are used with our proposed data model to analyze the data performance and study the influence of different delay constraints on data.

3.2 The Basic Model

3.2.1 System Environment

We consider that N voice users and a large number of data users are connected to a statistical multiplexer and gated into a TDM link. As shown in Figure 3.1, the link is logically divided into frames whose duration, F, is equal to the packet generation time of a voice user (i.e., each voice user will generate exactly one packet during a frame). Each frame can be further subdivided into γ slots, where a slot is the time necessary to transmit a packet. We assume that there are γ_d slots are dedicated to data, the remaining

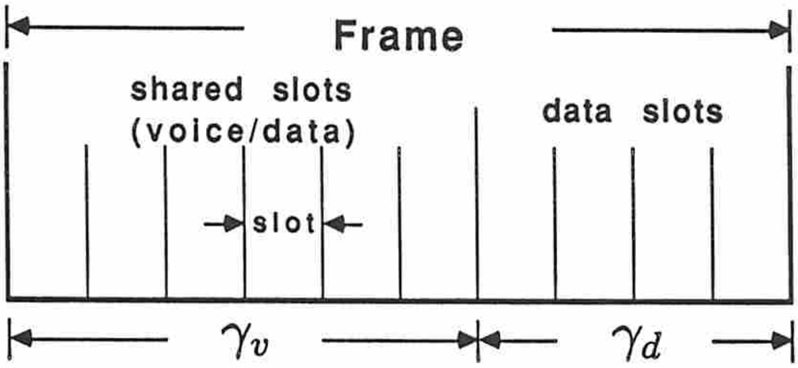


Figure 3.1: Frame structure for integrated voice/data services

γ_v $(= \gamma - \gamma_d)$ slots are shared by voice and data. The system examines the buffer at the beginning of each frame to determine how many voice and data packets are served in this frame. As many voice packets as possible are served (up to the number of reserved slots) and data uses any unused slots. To satisfy the delay requirement, a voice packet arriving in the previous frame is discarded if it cannot receive service in the current frame. We thus model the system as a queueing system with γ-servers and a gate at the entrance of the TDM link, which opens at the beginning of each frame, as shown in Figure 3.2. To simplify the analysis, we make the following assumptions:

1. Each voice user is alternating between *talkspurt* and *silence* periods which are exponentially distributed with mean λ_v^{-1} and μ_v^{-1}, respectively.
2. SAD is used such that packets are only generated while the voice user is in talkspurt period.
3. The arrival process of data packets is Poisson with rate λ_d.
4. The size of the voice buffer is equal to the number of voice users, since at most N voice packets will be generated during a frame; the data buffer is finite and large enough such that the probability of blocked data packets is negligible.

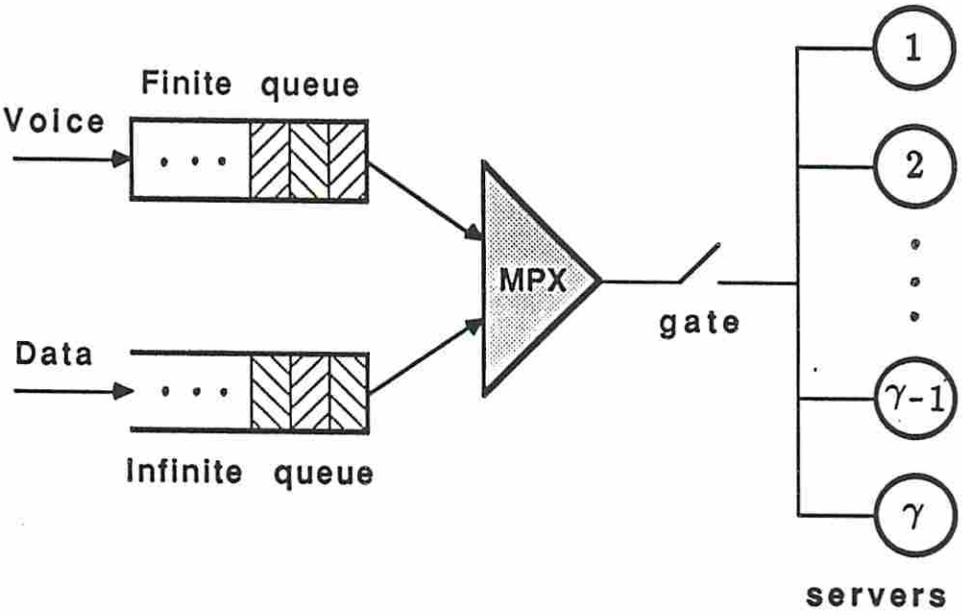


Figure 3.2: A packet switched model for integrated voice/data services

3.2.2 Voice Performance

Since a voice user is always alternating between *talkspurt* and *silence*, it can be modeled as an independent two-state Markov chain [5], as shown in 3.3. The steady state behavior of a voice user is described by the limiting state probabilities of the *talkspurt* and *silence* states [27], which are defined by π_t and π_s with

$$\begin{aligned} \pi_t &= \frac{p_{01}}{p_{01}+p_{10}} \\ \pi_s &= \frac{p_{10}}{p_{01}+p_{10}} \end{aligned} \tag{3.1}$$

where p_{10} and p_{01} are the transition probabilities from the *talkspurt* and the *silence* state to the other state, respectively, and are given by

$$p_{01} = 1 - e^{-\lambda_v F}$$

$$p_{10} = 1 - e^{-\mu_v F}$$

Since the frame length is very small compared to the talkspurt or silence period (i.e., $F << \lambda^{-1}$ and $F << \mu_{-1}$), we can approximate p_{01} and p_{10} by

$$p_{01} \approx \lambda_v F$$

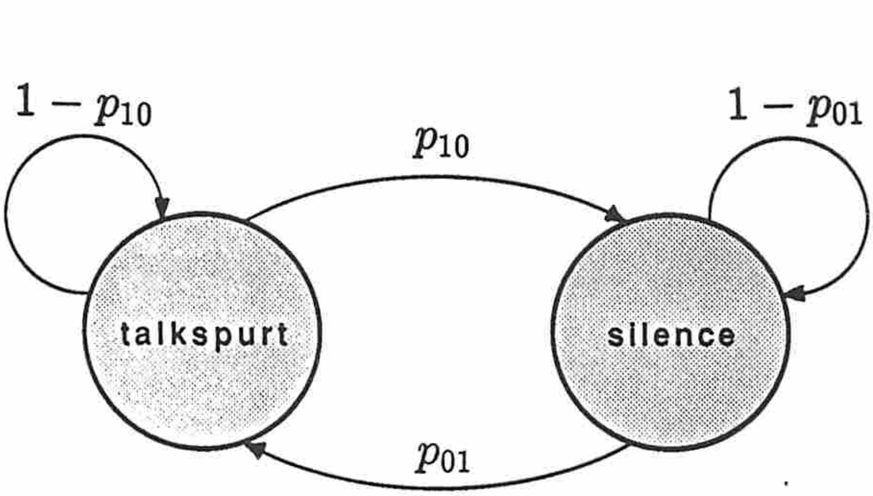


Figure 3.3: A two-state Markov model for a voice user

$$p_{10} \approx \mu_v F$$

The probabilities π_t and π_s then become

$$\pi_t = \frac{\mu_v^{-1}}{\lambda_v^{-1} + \mu_v^{-1}}$$

$$\pi_s = \frac{\lambda_v^{-1}}{\lambda_v^{-1} + \mu_v^{-1}}$$

Let V be the number of voice users in talkspurt, the steady state number of voice users in the system can be characterized by a binomial distribution

$$P_V(k) = Pr\{V = k\} = \binom{N}{k} \pi_t^k \pi_s^{N-k} \tag{3.2}$$

The voice performance measure that we are interested is the average packet loss probability ϕ, which is readily computed by

$$\phi = \frac{1}{N\pi_t} \sum_{k=\gamma_v+1}^{N} (k - \gamma_v) \binom{N}{k} \pi_t^k (1 - \pi_t)^{N-k} \tag{3.3}$$

which agrees with the result from Weinstein [60], if π_t is replaced by η, the *activity factor* of a voice user.

3.2.3 Data Performance

Once the voice traffic slots allocation in a frame are allocated, the remaining unused slots can all be used by data. Thus, the data performance depends on how many slots in a frame are occupied by voice users, in other words, how many voice users are in talkspurt at the beginning of this frame. The most straightforward way to solve this problem is to model the system at frame boundaries as a two-dimensional Markov chain, with one dimension for the number of voice users in talkspurt and the other dimension for the number of data packets in the queue. However, as discussed above, this approach is intractable for large systems.

To overcome this drawback, we propose an approximate one-dimensional Markov model with state D, which represents the number of data packets in the queue at the frame boundary. The influence of voice traffic on data performance is adequately represented by its steady state behavior, since talkspurt and silence periods are much longer than frames so that data transitions occur much more frequently than voice transitions.

Let D_n be the state at the nth frame, the state transition from D_n to D_{n+1} depends not only on the number of data packet arrivals but also on the number of available slots for data in the current frame, which we denote it by K_n. The state transition probability is given by

$$\begin{aligned} p_{ij} &= Pr\{D_{n+1} = j | D_n = i\} \\ &= \sum_{l=0}^{\infty} \sum_{k=\gamma_d}^{\gamma} Pr\{K_n = k|i\} \frac{(\lambda_d F)^l}{l!} e^{-\lambda_d F} \delta(l,k) \end{aligned} \tag{3.4}$$

where

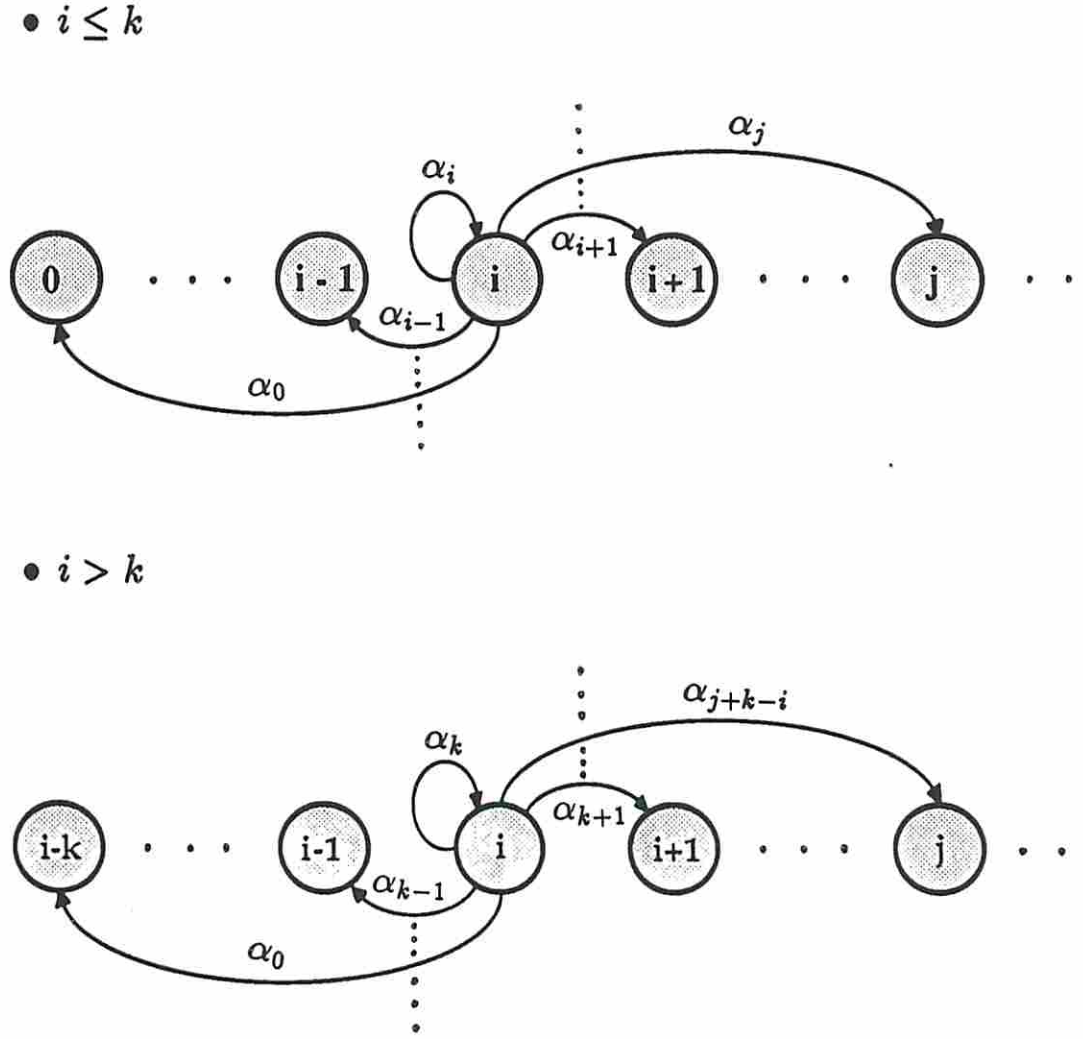
$$\delta(l,k) = \begin{cases} 1 & , \ j = \max\{i-k, 0\} + l \\ 0 & , \ \text{otherwise} \end{cases}$$

From Bayes' theorem, we can rewrite $Pr\{K_n = k|i\}$ by

$$Pr\{K_n = k|i\} = \frac{Pr\{i|K_n = k\}Pr\{K_n = k\}}{\sum_{k=\gamma_d}^{\gamma} Pr\{i|K_n = k\}Pr\{K_n = k\}} \tag{3.5}$$

The term $Pr\{K_n = k\}$ is the probability of k slots being available for data, which can be approximated by the steady state probability of $\gamma - k$ voice users being in talkspurt, i.e., $P_V(\gamma - k)$. The other unknown term $Pr\{i|K_n = k\}$ can be derived from a conditional Markov process given $K_n = k$.

The state transition diagram of this conditional Markov chain is shown in Figure 3.4. The transition probability is thus given by



- $i \leq k$

- $i > k$

$$\text{where } \alpha_j = \frac{(\lambda_d F)^j}{j!} e^{-\lambda_d F}$$

Figure 3.4: State Transition of the model conditioned on k servers

$$P_{ij|k} = Pr\{D_{n+1} = j | D_n = i, K_n = k\}$$

$$= \begin{cases} \dfrac{(\lambda_d F)^j}{j!} e^{-\lambda_d F} & , i \leq k \ , \ 0 \leq j < M_d \\[2ex] \dfrac{(\lambda_d F)^{j+k-i}}{(j+k-i)!} e^{-\lambda_d F} & , i > k \ , \ 0 \leq j < M_d \\[2ex] \displaystyle\sum_{m=M_d}^{\infty} \frac{(\lambda_d F)^m}{m!} e^{-\lambda_d F} & , i \leq k \ , \ j = M_d \\[2ex] \displaystyle\sum_{m=M_d+k-i}^{\infty} \frac{(\lambda_d F)^m}{m!} e^{-\lambda_d F} & , i > k \ , \ j = M_d \\[2ex] 0 & , \text{otherwise} \end{cases} \quad (3.6)$$

which can be used to find $Pr\{i|K_n = k\}$.

After computing the transition probabilities p_{ij}, we can solve the original Markov chain to obtain the steady state probability π_i, and then the mean queue length of data packets is calculated by

$$\overline{D} = \sum_{i=0}^{M} i \cdot \pi_i \quad (3.7)$$

3.2.4 Numerical Results

In this section, we give an example to evaluate the accuracy of our model. Let each voice user have 64 Kbps PCM digital signals when in talkspurt and the mean holding time of talkspurt and silence periods are 1.36 seconds and 1.8 seconds, respectively [40]. We consider a T1 link which has twenty-four slots in each frame. Six slots are dedicated for data users and the remaining eighteen slots are shared by voice and data users with voice priority. To keep the voice discard probability less than 1 percent, we limit the number of voice users to be 32. The data arrival process is Poisson with a rate of λ_d packets/frame. Since six slots are reserved for data, the traffic intensity of data is thus given by $\lambda_d/6$ packets/slot. Figure 3.5 shows the mean data queue length versus the data traffic intensity. As the traffic intensity increases to one, the mean data queue length should go to infinity if a "fixed" boundary is used. Since the boundary between voice and data is "movable", data has more than its reserved number of slots to serve packets and the mean queue length is low even when the data traffic intensity exceeds one. Simulation results are plotted for comparison, good consistency is shown. We also demonstrate the comparison between "fixed" boundary and "movable" boundary. As the data traffic intensity increases, movable boundary performs better than fixed boundary since data can

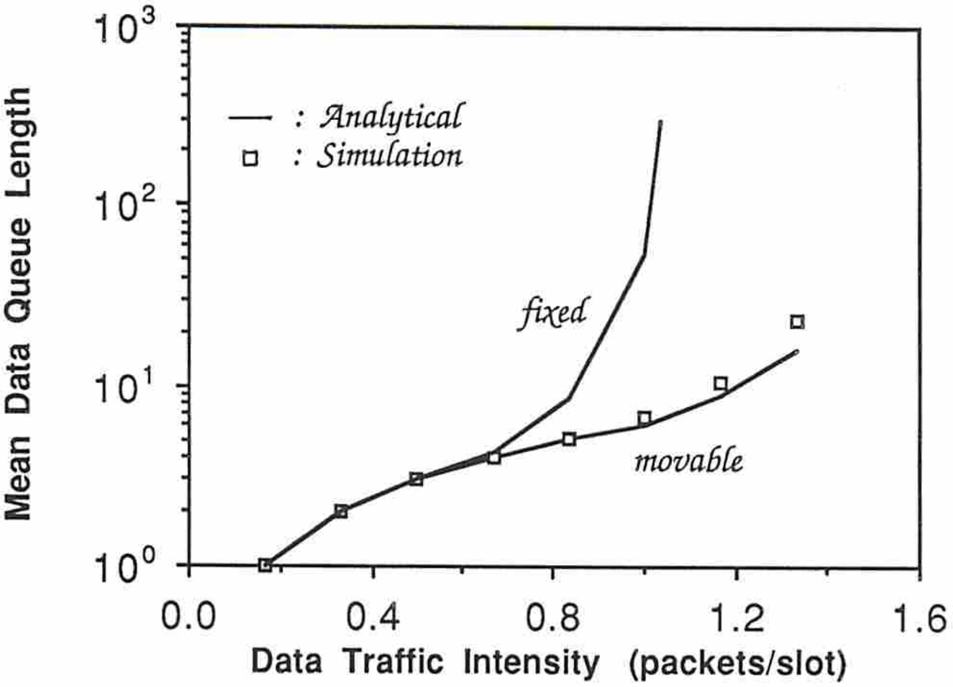


Figure 3.5: Mean data queue length vs. Data traffic intensity

steal some bandwidth from voice.

3.3 Extended Model with Delay-Constrained Voice Traffic

In chapter 2, we developed a model to measure the performance of delay-constrained voice traffic in a packet-switched system. We now study the influence of that traffic on data performance. Due to the simplicity of the model for data in the previous section, it is easily to extend the results to include the impact of delay-constrained voice traffic on data.

3.3.1 Voice Performance

As discussed in chapter 2, the performance of delay constrained voice traffic can be measured by solving a two-dimensional Markov chain. The state of this Markov chain is represented by $S = (t, b)$, where t is the number of voice users in talkspurt and b is the number of voice packets in the buffer.

In the environment we have here, there are two things different from the model described in chapter 2. One is that "gated service" is used instead of "immediate service", i.e., the arrivals in the current frame get service only from the next frame. The other is that the queue length found upon arrival does not directly give us the waiting time any more. The reason is, that in addition to the time of serving the packets in the queue, we have to include the time spent transmitting the data packets in each frame.

Define M_v as the appropriate buffer size at each frame boundary such that all packets in the buffer can be transmitted within their delay constraint, it is given by

$$M_v = D_t - \lceil \frac{D_t}{\gamma} \rceil \cdot \gamma_d$$

where D_t is the delay constraint in slots. Assume that the system at the beginning of the nth frame is in the state $S_n = (t_n, b_n)$ and let $N(b_n)$ be the number of voice packets that can be added to the buffer at this moment, then $N(b_n) = M_v - b_n$.

When $t_n \leq N(b_n)$, all t_n arrivals are accepted since the buffer is never full. When $t_n > N(b_n)$, the first $N(b_n)$ arrivals definitely get into buffer, the remaining $t_n - N(b_n)$ arrivals may or may not be discarded depending on their arrival instants. Let Z be a random variable to represent the number of the remaining arrivals that are discarded. In general, as discussed in chapter 2, we can summarize the transitions as follows:

$$\begin{aligned} t_{n+1} &= t_n - u + v \\ b_{n+1} &= \begin{cases} t_n + max(0, b_n - \gamma_v) & , t_n \leq N(b_n) \\ b_n + t_n - (\gamma_v + Z) & , t > N(b_n) \end{cases} \end{aligned} \tag{3.8}$$

where $u(v)$ is the number of voice users that switch from *talkspurt (silence)* to *silence (talkspurt)*.

Define $\Psi_{Z_{ij}}(k)$ as the probability that $Z = k$, given $t_n = i$ and $b_n = j$, the transition probability $p_{ij,kl} = Pr\{t_{n+1} = k, b_{n+1} = l | t_n = i, b_n = j\}$ is then written by

- For $t_n \leq N(b_n)$, $$p_{ij,kl} = \begin{cases} \Phi_{ik} & , l = j - \gamma_v + i \\ 0 & , \text{otherwise} \end{cases}$$
- For $t_n > N(b_n)$, $$p_{ij,kl} = \begin{cases} \Phi_{ik} \Psi_{Z_{ij}}(i + j - \gamma_v - l) & , l_L \leq l \leq l_U \\ 0 & , \text{otherwise} \end{cases}$$

where $l_L = b_n + N(b_n) + 1 - \gamma_v$ and $l_U = \min\{i + j - \gamma_v, b_n + N(b_n)\}$. Recall that Φ_{ik} is the probability that k voice users will be in talkspurt in the next frame when i voice users are in talkspurt in the current frame, which has been already defined in chapter 2, and the derivation of the probability $\Psi_{Z_{ij}}(\cdot)$ is discussed in Appendix A as $\Psi_Z(\cdot)$, where $Z = \mathcal{D}_1$ if using scheme 1 and $Z = \mathcal{D}_2$ if using scheme 2.

After computing the transition matrix $\mathbf{P}=[p_{ij,kl}]$, we solve the equation $\pi = \pi\mathbf{P}$ with $\sum_i \sum_j \pi_{ij} = 1$ to obtain the equilibrium state probability vector $\pi = [\pi_{ij}]$. The average discard probability is thus given by

$$P_{dis} = \frac{\sum_{i=0}^{N} \sum_{j=0}^{M_v} \left(\sum_{k} k \cdot \Psi_{Z_{ij}}(k) \right) \cdot \pi_{ij}}{\sum_{i=0}^{N} \sum_{j=0}^{M_v} i \cdot \pi_{ij}} \tag{3.9}$$

3.3.2 Data Performance

The data performance analysis is the same as in section 3.2.3 except that the probability $Pr\{K_n = k\}$ (the probability of k slots available for data in a frame) is approximated by the results obtained in the previous section instead of the voice steady state behavior. If k is greater than γ_d then the voice buffer length must be equal to $\gamma - k$; on the other hand, when the buffer length is equal to or greater than γ_d, k must be equal to γ_d, therefore we have

$$Pr\{K_n = k\} = \begin{cases} \sum_{i=0}^{N} \pi_{i,\gamma-k} & , k > \gamma_d \\ \sum_{i=0}^{N} \sum_{j=\gamma_v}^{M_v} \pi_{ij} & , k = \gamma_d \end{cases}$$

Then, using (3.4), (3.5) and (3.6), we can compute the transition probability p_{ij} and solve the balance equation to get the steady state probability π_i and the mean queue length of data packets $\overline{D}$ is obtained.

3.3.3 Numerical Results

To study the influence of delay-constrained voice traffic on data performance, we show several examples. The system environment here is assumed to be the same as that in the previous section except that each voice packet is constrained by a tolerable delay D_t. We vary this delay constraint to be 25 ms, 50 ms, 100 ms and 200 ms and plot the mean data queue length against data traffic intensity, as shown in Figure 3.6. We also plot the curve for *instant discarding* (i.e., $D_t = 0$), from Figure 3.5, in this figure. It is shown that the larger the delay, the longer the data queue length. Since a longer delay constraint results in more queued voice packets in the buffer, less bandwidth can be stolen by data from voice. Thus, the data queue length increases as the voice delay constraint increases.

3.4 Summary

In this chapter, we have proposed an approximate one-dimensional model to measure data performance in a packet switched integrated voice and data system. This model is simple

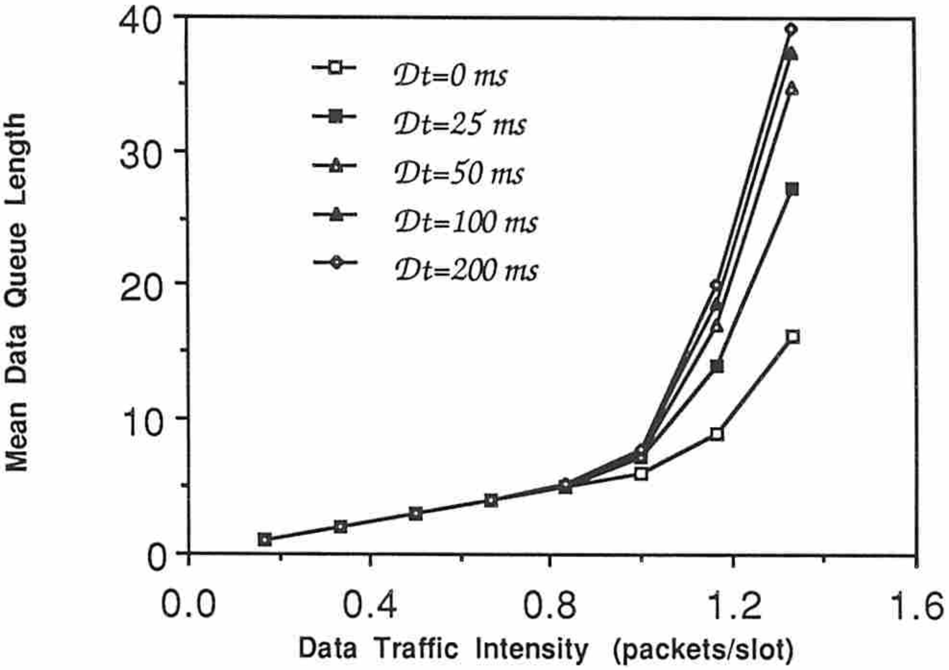


Figure 3.6: Effect of delay-constrained voice traffic on data performance

but still accurate. Furthermore, it does not have the memory limitation problem when modeling a larger system due to the one-dimensional state representation. We also study the influence of various voice traffic delay constraints on data performance, the results show that there is a tradeoff between decreasing the voice discard probability and decreasing the mean data queue length.

Chapter 4

A Performance Model for Burst-Switched Integrated Voice/Data Traffic

4.1 Introduction

Burst Switching [3,25] is a switching technique developed by GTE laboratories for serving the integrated voice and data traffic on a TDM link. Both types of traffic are treated in a unified manner and a *burst* is either a talkspurt or a data message. The switch employs speech activity detection for voice so that only the talkspurts are transmitted, which results in highly efficient utilization of the channel bandwidth. When a burst arrives, an available channel is dedicated for its transmission. Once the burst ends, the channel becomes available again. As mentioned in chapter 1, voice and data have different performance requirements. Voice samples cannot suffer longer delay (otherwise, they are useless) but some may be dropped with little effect of voice quality, whereas data messages can be delayed but no bits may be lost. To achieve these performance goals, the switch gives nonpreemptive priority to voice and buffers data messages if necessary. Thus, when all output channels are busy, an incoming voice burst is blocked and clipped from the front end (*freeze-out*) until an output channel becomes available; on the other hand, an incoming data burst is buffered in the switch until a channel becomes available. The performance measures that we are interested are in thus the average voice cutout fraction, ϕ, and the mean waiting time of a data burst, W.

To completely describe the behavior of such a burst-switched system, as shown in Figure 4.1, a three-dimensional state representation $S = (t, b, q)$ is necessary, where t is the number of voice users in talkspurt, b is the number of channels occupied by data users and q is the number of data bursts in the queue. Due to computational limitations, this exact model

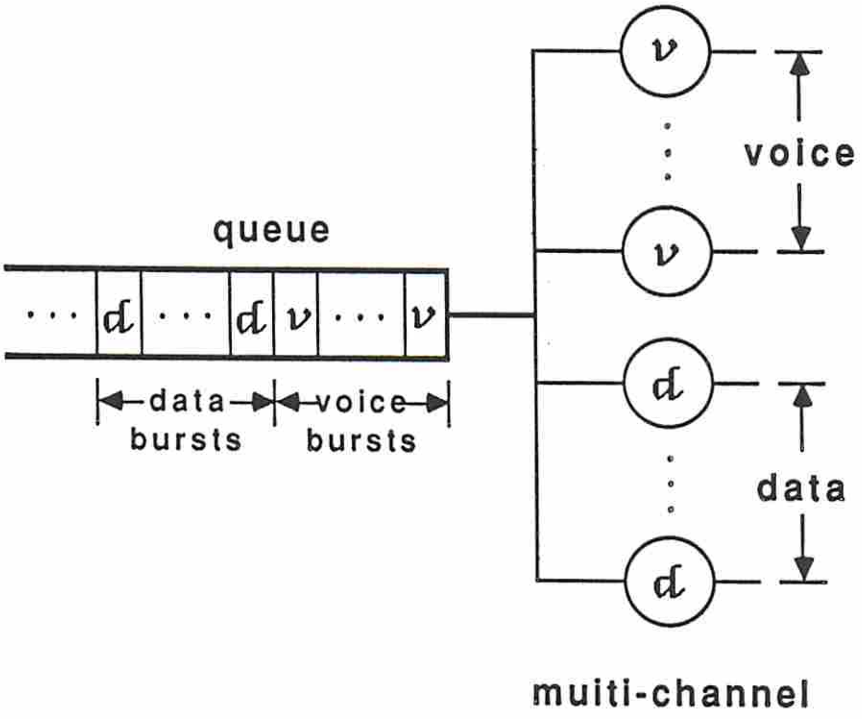


Figure 4.1: System model for a burst-switched node

is intractable for large systems. Several approximate models have been proposed in the past few years. O'Reilly [50,51] analyzed the performance of interactive data traffic by using the fluid-flow approach with the assumption that voice has preemptive priority over data traffic. Ma and Mark [43] assumed that voice has nonpreemptive priority over data traffic and decomposed the three-dimensional state space into two subspaces in analysis. One consists of *free-server* states which are two-dimensional, the other consists of three-dimensional *conditional-queue* states. Several approximations for various types of data traffic are considered and give accurate results for most traffic levels, but the approach is quite complex.

In this chapter, we propose a different approximate model for burst-switched integrated voice/data traffic, where voice has nonpreemptive priority over data. The model is simple but accurate. In section 4.2, we describe the general ideas of our approach, discuss the state representation and the state transitions, and obtain the performance measures of interest. In section 4.3, we discuss the approximations for various types of data traffic to compute transition probabilities. In section 4.4, numerical examples are evaluated and the analytical model is validated by simulation. Section 4.5 concludes the chapter.

4.2 Model Description

We consider a system consisting of N voice users and a large number of data users which are connected to a burst-switched node and gated into a multi-channel link. The link has C identical channels where part, say C_d channels, are reserved for data only and the others are shared by voice and data. Voice has nonpreemptive priority over data traffic. Several assumptions are made to simplify the analysis:

- Each voice user is alternating between talkspurt and silence periods which are exponentially distributed with mean λ_v^{-1} and μ_v^{-1}, respectively.
- The arrival process of data bursts is Poisson with rate λ_d and the length of a data burst is exponentially distributed with mean μ_d^{-1}.
- The buffer to store data bursts has infinite capacity.

Based on these assumptions, the system can be described by a 3-dimensional Markov process with the state $S = (t, d, q)$ as mentioned before. To simplify the analysis, we approximate this Markov process by two two-dimensional Markov processes with states $S_1 = (t, d)$ and $S_2 = (d, q)$, respectively. We use the first Markov process to evaluate the average voice cutout fraction ϕ and the second Markov process to compute the mean data waiting time W. With this decomposition, the statistics of q are needed when solving the first Markov chain. Similarly, solution of the second Markov chain needs the statistics of t. We will discuss how we estimate these statistics in the next section.

4.2.1 The Average Voice Cutout Fraction ϕ

Given a state (t, d), the transitions in the first Markov process can be generally categorized into four cases, as shown in Figure 4.2, where

Case 1 : $d < C_d$.

Since not all reserved channels for data are occupied, transitions of t and q behave as two independent birth-and-death process.

Case 2 : $d \geq C_d$ and $t + d = C$.

In this case, all channels are busy and no voice burst is waiting for service. The transition depends on the length of data queue. When a talkspurt ends, since no voice burst is queued, the released channel is taken by a data user only if the data queue is not empty. Similarly, when a data burst finishes, the state switches to $(t, d-1)$ only if the data queue is empty.

Case 3 : $d = C_d$ and $t + d > C$.

The second condition in this case implies that at least one voice burst is queued and clipped. Since data users did not occupy any shared slot, the transition of d depends on the status of the data queue only, as in case 2, and the transition of t is still a birth-death process.

Case 4 : $d > C_d$ and $t + d > C$.

The transition in this case is similar case 3 except that the next state will be $(t, d-1)$ with probability 1 when a data burst completes. This is due to the fact that some voice bursts are waiting for the shared slots currently occupied by data.

Let $q_{ij.kl}$ be the transition rate from state (i,j) to state (k,l), then the transition rates are summarized as follows:

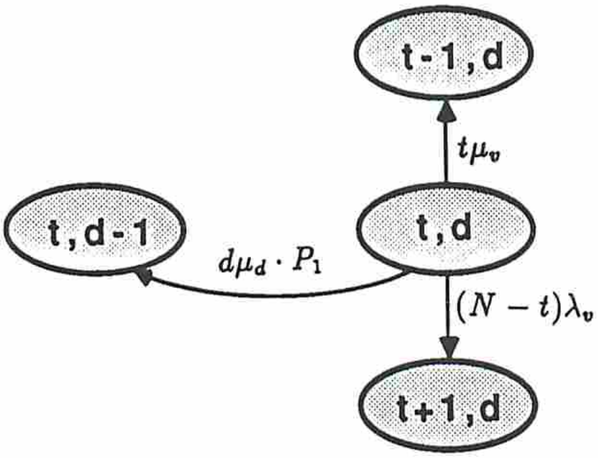
$$\textit{Case 1}: \quad q_{ij,kl} = \begin{cases} (N-i)\lambda_v & , k=i+1 \;,\; l=j \\ i\mu_v & , k=i-1 \;,\; l=j \\ \lambda_d & , k=i \;,\; l=j+1 \\ j\mu_d & , k=i \;,\; l=j-1 \\ 0 & , \text{otherwise} \end{cases}$$

$$\textit{Case 2}: \quad q_{ij,kl} = \begin{cases} (N-i)\lambda_v & , k=i+1 \;,\; l=j \\ i\mu_v \cdot P_1 & , k=i-1 \;,\; l=j \\ i\mu_v \cdot \overline{P_1} & , k=i-1 \;,\; l=j+1 \\ j\mu_d \cdot P_1 & , k=i \;,\; l=j+1 \\ 0 & , \text{otherwise} \end{cases}$$

$$\textit{Case 3}: \quad q_{ij,kl} = \begin{cases} (N-i)\lambda_v & , k=i+1 \;,\; l=j \\ i\mu_v & , k=i-1 \;,\; l=j \\ j\mu_d \cdot P_1 & , k=i \;,\; l=j-1 \\ 0 & , \text{otherwise} \end{cases}$$

$$\textit{Case 4}: \quad q_{ij,kl} = \begin{cases} (N-i)\lambda_v & , k=i+1 \;,\; l=j \\ i\mu_v & , k=i-1 \;,\; l=j \\ j\mu_d & , k=i \;,\; l=j-1 \\ 0 & , \text{otherwise} \end{cases}$$

- Case 3 : $d = C_d$, $t > C - d$



- Case 4 : $d > C_d$, $t > C - d$

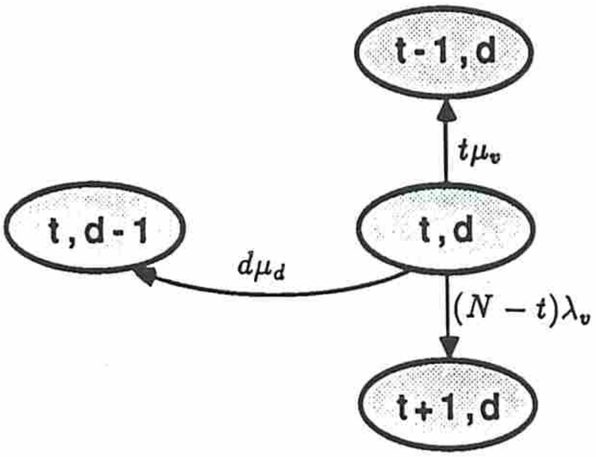
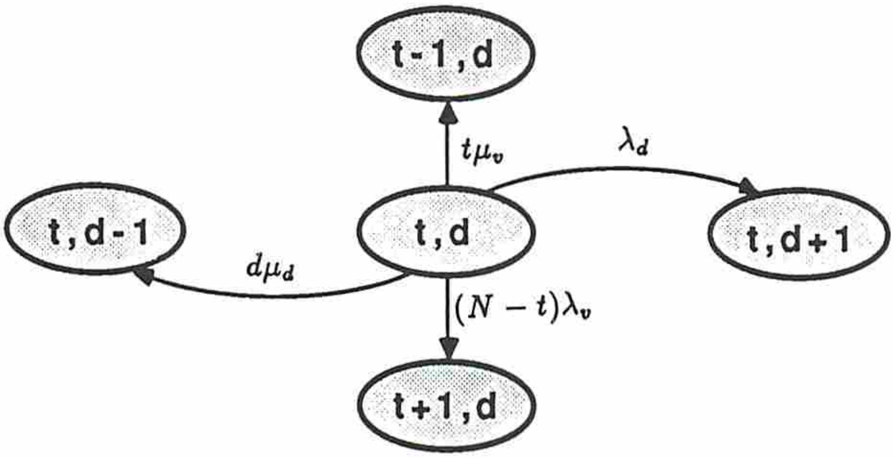
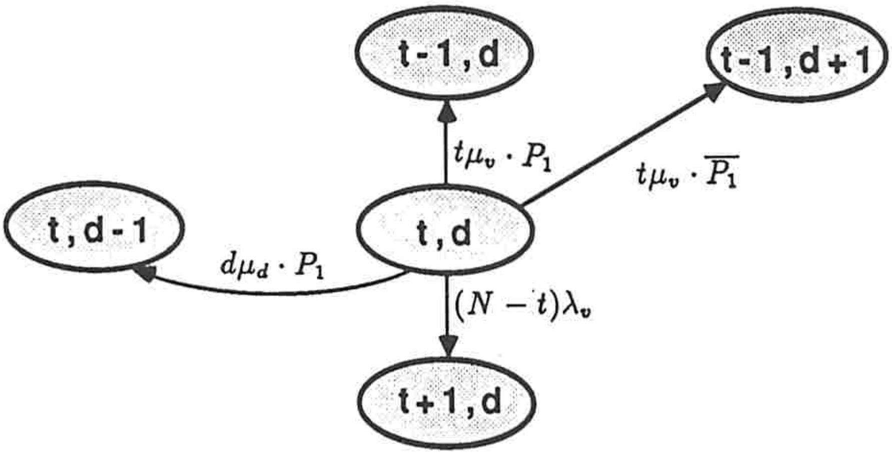


Figure 4.2: State transition of the Markov chain (t, d)

- Case 1 : $d < C_d$



- Case 2 : $d \geq C_d$, $t = C - d$



where P_1=Pr { data queue is empty }

Figure 4.2: State transition of the Markov chain (t, d)

where P_1 is the probability of the data queue being empty at the given state and $\overline{P_1} = 1 - P_1$.

After finding the transition rate matrix $\tilde{Q}$, we can solve this Markov chain by $\pi = \pi\tilde{Q}$ subject to $\sum_t \sum_d \pi_{td} = 1$ to find the equilibrium state probabilities π_{td}, $0 \leq t \leq N$ and $0 \leq d \leq C$, where $\pi = [\pi_{td}]$ is the state probability vector. The average voice cutout fraction ϕ is then given by

$$\phi = \frac{\sum_t \sum_d max\{0, t - C + d\} \cdot \pi_{td}}{\sum_t \sum_d t \cdot \pi_{td}}$$

4.2.2 The Mean Data Waiting Time W

It is more difficult to solve the second Markov chain since the system has an infinite data buffer. The method from Neuts [48] can be applied here to get the solution in a geometric matrix form. We first investigate the transitions out of a given state (d, q), which are also grouped into four cases as follows:

Case 1 : $d < C_d$.

In this case, the data queue is empty, so the process reduces to a pure birth-death process.

Case 2 : $d \geq C_d$ and $q = 0$.

Since the data queue is empty, once a new data burst arrives, it will either find an available channel or have to be buffered. We need to know how many voice users are in talkspurt to estimate the probability of an available channel existing, we call this P_2.

Case 3 : $d = C_d$ and $q > 0$.

The data queue being not empty implies that all other channels are occupied by voice. When a talkspurt completes, the released channel only can be taken over by a data burst if no voice burst is waiting for service. Hence we need to estimate the probability of the voice queue being empty - P_3.

Case 4 : $d > C_d$ and $q > 0$.

Since the data queue is not empty and some shared channels are occupied by data, once one of them is released, it can be used again by data users only if no voice bursts is queued. As in case 3, the transition depends on the probability P_3.

The transition diagrams are shown in Figure 4.3. Given a state (i, j), let $q_{ij,kl}$ be the transition rate from this state to the state (k, l), then we can summarize the transitions as follows:

$$\textit{Case 1}: \quad q_{ij,kl} = \begin{cases} \lambda_d & , k = i+1 \ , \ l = j \\ j\mu_d & , k = i-1 \ , \ l = j \\ 0 & , \text{otherwise} \end{cases}$$

$$\textit{Case 2}: \quad q_{ij,kl} = \begin{cases} \lambda_d \cdot P_2 & , k = i+1 \ , \ l = j \\ j\mu_d & , k = i-1 \ , \ l = j \\ \lambda_d \cdot \overline{P_2} & , k = i \ , \ l = j+1 \\ 0 & , \text{otherwise} \end{cases}$$

$$\textit{Case 3}: \quad q_{ij,kl} = \begin{cases} \lambda_d & , k = i \ , \ l = j+1 \\ j\mu_d & , k = i \ , \ l = j-1 \\ (C - C_d)\lambda_v \cdot P_3 & , k = C_d + 1 \ , \ l = j-1 \\ 0 & , \text{otherwise} \end{cases}$$

$$\textit{Case 4}: \quad q_{ij,kl} = \begin{cases} \lambda_d & , k = i \ , \ l = j+1 \\ j\mu_d \cdot P_3 & , k = i \ , \ l = j-1 \\ j\mu_d \cdot \overline{P_3} & , k = i-1 \ , \ l = j \\ (C - i)\lambda_v \cdot P_3 & , k = i+1 \ , \ l = j-1 \\ 0 & , \text{otherwise} \end{cases}$$

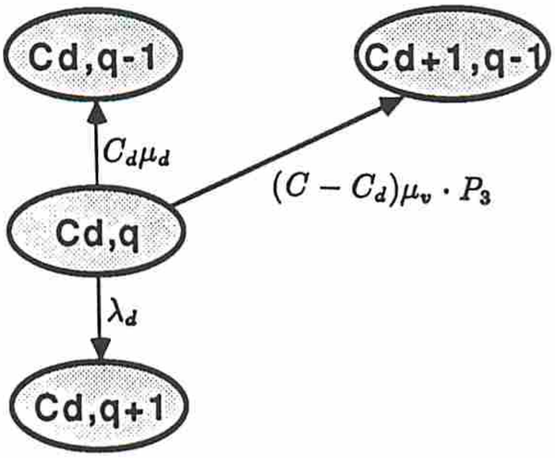
Let all states having i data bursts in the queue be at the *level* i and define π_i to be a state probability vector of level i. Then, from the state diagrams, we have

$$\pi_0 B_0 + \pi_1 B_1 = 0$$

$$\pi_0 C_0 + \pi_1 A_1 + \pi_2 A_2 = 0$$

$$\pi_{i-1} A_0 + \pi_i A_1 + \pi_{i+1} A_2 = 0 \quad , \ i > 1$$

- Case 3 : $d = C_d$, $q > 0$



where P_3=Pr { voice queue is empty }

- Case 4 : $d > C_d$, $q > 0$

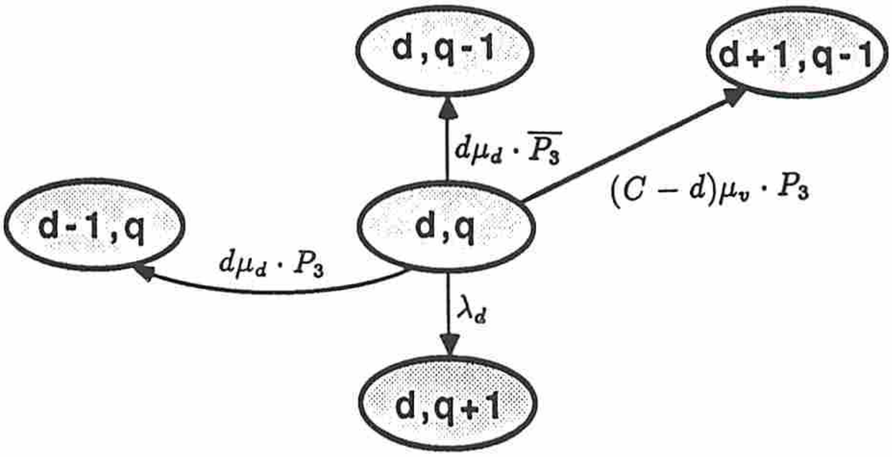
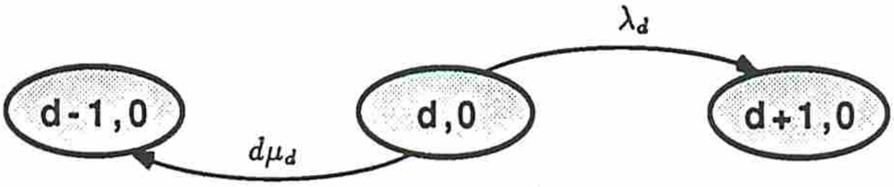
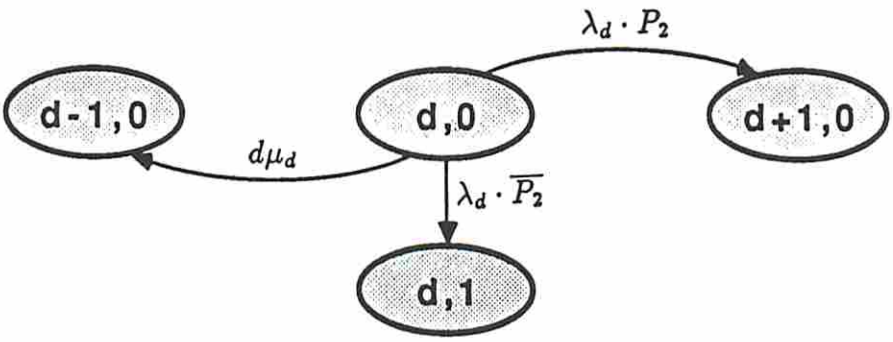


Figure 4.3: State transition of the Markov chain (d, q) (cont.)

- Case 1 : $d < C_d$



- Case 2 : $d \geq C_d$, $q = 0$



where P_2=Pr { avaliable channel(s) exist }

Figure 4.3: State transition of the Markov chain (d, q)

where

$$\begin{aligned}
B_0 &\triangleq \text{the transition rate matrix from level 0 to level 0} \\
B_1 &\triangleq \text{the transition rate matrix from level 1 to level 0} \\
C_0 &\triangleq \text{the transition rate matrix from level 0 to level 1} \\
A_0 &\triangleq \text{the transition rate matrix from level } i-1 \text{ to level } i \\
A_1 &\triangleq \text{the transition rate matrix from level } i \text{ to level } i \\
A_2 &\triangleq \text{the transition rate matrix from level } i+1 \text{ to level } i
\end{aligned}$$

Using the method from Neuts [48], we obtain a solution in geometric form

$$\pi_i = \pi_1 \cdot R^{i-1} \quad , \ \forall i \geq 1$$

where R satisfies $A_0 + RA_1 + R^2 A_2 = 0$. The state probability vectors π_0 and π_1 can be found from

$$[\pi_0, \ \pi_1] \begin{bmatrix} B_0 & C_0 \\ B_1 & A_1 + RA_2 \end{bmatrix} = [0, \ 0]$$

subject to $\pi_0 \mathbf{e} + \pi_1 (I-R)^{-1} \mathbf{e} = 1$, where $\mathbf{e} = [1, 1, \ldots, 1]^T$ and I is the identity matrix. By Little's result, it is easy to find the mean data waiting time W by

$$W = \frac{\overline{q}}{\lambda_d} = \frac{1}{\lambda_d} \pi_1 (I-R)^{-2} \mathbf{e}$$

4.3 The Probabilities P_1, P_2 and P_3

The unknown probability P_1 in the first Markov process can be estimated from the state probabilities of the second Markov process. On the other hand, based on the state probabilities of the first Markov process, we can estimate the unknown probabilities P_2 and P_3 in the second Markov process. They are dependent on each other. We select one Markov process to solve first and then use the results to solve the other. In other words, we estimate P_1 (if solving the first Markov chain first) or P_2 and P_3 (if solving the second Markov chain first) by some approximation. Correct choice of the order to solve these two Markov chains can obtain good prediction of the performance, but wrong choice will get very bad results. Since P_1 is a data statistic conditioned on voice traffic and P_2 and P_3 are voice statistics conditioned on data traffic, the decision to choose which Markov process to solve can depend on the relative frequency of fluctuations of voice and data traffic. If the fluctuation of data traffic is more frequent than that of voice traffic, we solve the first Markov process first since P_1 can be approximated by the steady state behavior of data traffic. On the other hand, i.e., when voice traffic fluctuates more rapidly, we choose the second Markov process to solve first since P_2 and P_3 can be approximated by the steady state behavior of voice traffic. The characteristics of voice traffic is fixed since it depends mainly upon the characteristics of the speech activity detector (SAD). We, therefore, consider different data traffic statistics in the following numerical examples.

4.3.1 Interactive Data Traffic

The first case that we consider is the traffic generated by interactive data users. The major characteristic of this type of traffic is that the service time is very short compared to voice talkspurt or silence periods. Therefore, we select the first Markov chain to solve first and approximate P_1 by an $M/M/K$ queueing system. Define $Q_0(K)$ as the probability that the queue is empty, from [33] we have

$$Q_0(K) = 1 - \frac{\left(\frac{(K\rho)^K}{K!}\right)\left(\frac{1}{1-\rho}\right)}{\left[\sum_{i=0}^{K-1}\frac{(K\rho)^i}{i!} + \left(\frac{(K\rho)^K}{K!}\right)\left(\frac{1}{1-\rho}\right)\right]}$$

where $\rho = \lambda_d / K\mu_d$. Given a state $S = (t, d)$, we can consider that there are d servers for data traffic, thus the probability P_1 of this state is given by

$$P_1 = \begin{cases} Q_0(d) & , \frac{\lambda_d}{d\mu_d} < 1 \\ 0 & , \frac{\lambda_d}{d\mu_d} \geq 1 \end{cases}$$

After solving this Markov chain, we find the equilibrium state probabilities π_{td}, $0 \leq t \leq N$, $0 \leq d \leq C$, which allow us to estimate P_2 and P_3 for the second Markov chain.

In the second Markov chain, when the data queue is empty and no shared channels are occupied by data bursts (i.e., $d \leq C_d$), then the number of voice users in talkspurt is unconstrained (i.e., $0 \leq t \leq N$). If some shared channels are occupied by data bursts (i.e., $d > C_d$) however, then it is very likely that the number of voice users in talkspurt is less than or equal to the number of channels that voice can use (i.e., $0 \leq t \leq C - d$), since voice has higher priority and data fluctuates very frequently. Thus, for a given state (d, q), the probability P_2 can be approximated by

$$P_2 = \begin{cases} \sum_{t=0}^{C-C_d-1} \pi_{td} & , d = C_d \\ \sum_{t=0}^{C-d-1} \pi_{td} / \sum_{t=0}^{C-d} \pi_{td} & , d > C_d \end{cases}$$

When the data queue is not empty, the number of voice users in talkspurt must be greater than the number of channels occupied by voice bursts (i.e., $t \geq C - d$), so we approximate P_3 for a given state (d, q), $q > 0$, by

$$P_3 = \frac{\pi_{C-d,d}}{\sum\limits_{t=C-d}^{N} \pi_{td}}$$

4.3.2 File Transfer Data Traffic

This traffic type corresponds to file transfers, which usually have a longer service time than a voice talkspurt or silence period. Since the voice traffic has the dominant fluctuations, we solve the second Markov chain first, where P_2 and P_3 are approximated by the steady state behavior of voice traffic. Let η be the activity factor of a voice node, defined by

$$\eta = \frac{1/\mu_v}{1/\lambda_v + 1/\mu_v}$$

The probability of k voice users in talkspurt, $T(k)$, is then given by

$$T(k) = \begin{pmatrix} N \\ k \end{pmatrix} \eta^k (1 - \eta)^{N-k}$$

For the state (d, q), $q = 0$, the probability that a channel is available is equal to the probability that the remaining $C - d$ channels are not all occupied. In other words, the number of voice users in talkspurt is less than $C - d$. Hence, P_2 is approximated by

$$P_2 = \sum_{t=0}^{C-d-1} T(t)$$

If the data queue is not empty (i.e., $q > 0$), which implies that all channels are occupied and there may also be voice bursts waiting for service, then for a given state (d, q), the probability that no voice burst is waiting is equal to the probability that only $C - d$ voice users are in talkspurt. Thus, we approximate P_3 by

$$P_3 = \frac{T(C-d)}{\sum\limits_{t=C-d}^{N} T(t)}$$

Similarly, after obtaining the equilibrium state probabilities π_{dq} of the second Markov chain, we can estimate P_1 for a given state (t, d) by

$$P_1 = \pi_{d0}$$

and then solve the first Markov chain.

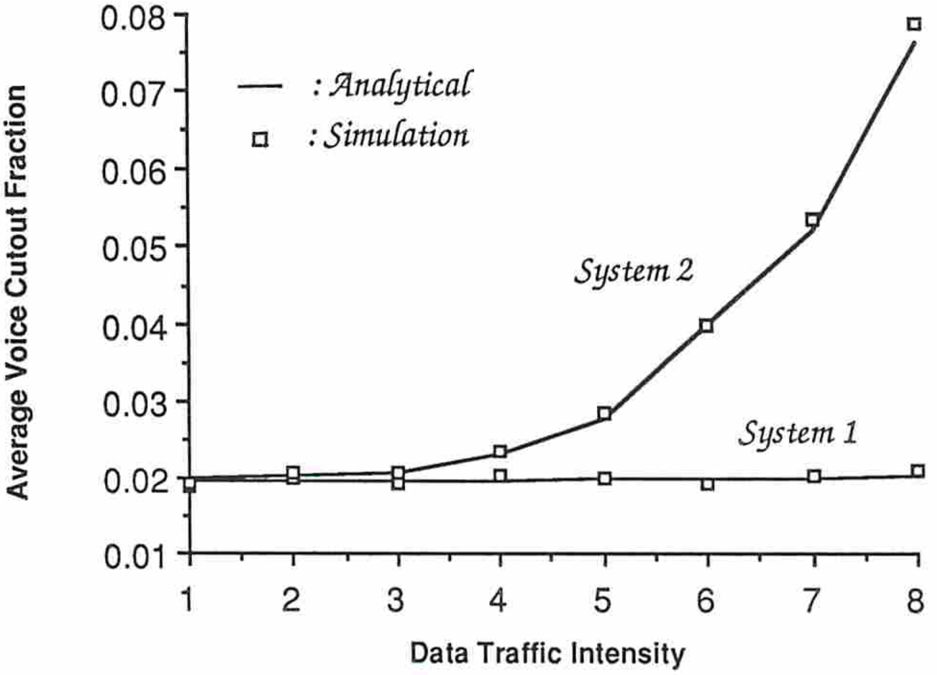


Figure 4.4: Average voice cutoff fraction vs. Data traffic intensity

4.4 Numerical Results

In this section, we investigate some specific scenarios and validate the model by simulation. A TDM link having 24 digital transmission channels (i.e., T1 trunk) is considered. Six channels are reserved for data and the others are shared by voice and data. Talkspurts are of average length 1.36 seconds and silence periods are of mean length 1.8 seconds [5]. To keep the average voice cutout fraction at a reasonable level, we limit the number of voice users to be twice the number of shared channels, i.e., $N = 36$. Two numerical examples corresponding to the two types of data traffic discussed in the last section are considered. System 1 has interactive data traffic with mean data service time $\mu_d^{-1} = 20ms$ and system 2 has file transfer data traffic with mean data service time $\mu_d^{-1} = 10s$. In Figure 4.4, we plot the average voice cutout fraction versus the data traffic intensity (λ_d/μ_d). System 1 has an almost flat curve due to the very small holding time on each shared channel occupied by a data burst. On the other hand, a longer data holding time reduces the immediate availability of shared channels and thus increases the clipping of waiting talkspurts, as is seen in the curve. Simulation results are also shown in this figure to compare with the analytical results. We note good agreement. Figure 4.5 shows the mean data waiting time versus the data traffic intensity. Again, the analytical results are validated by simulation showing good accuracy.

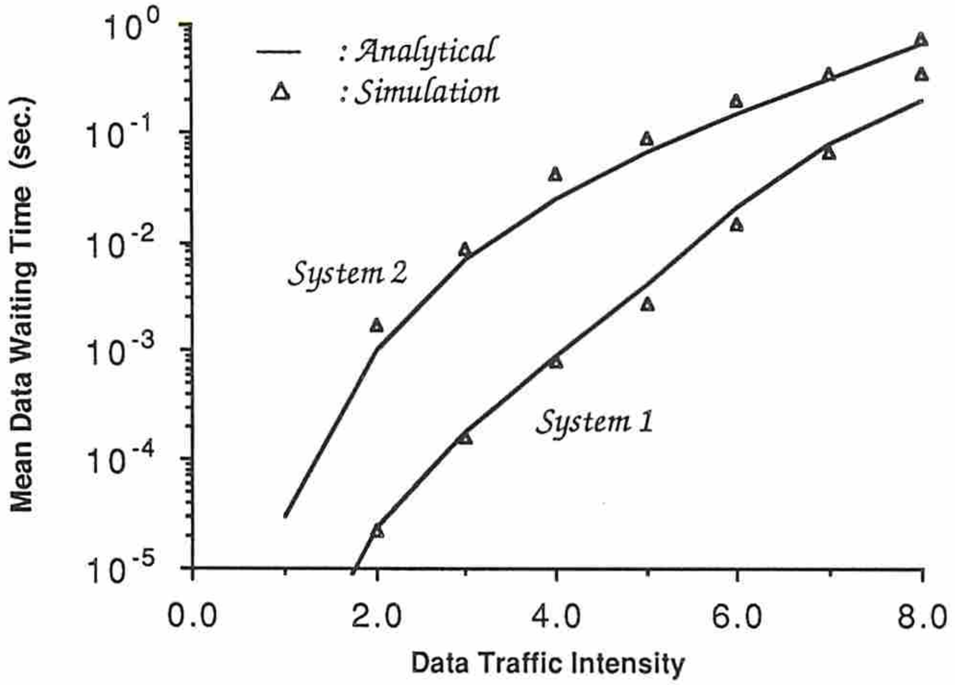


Figure 4.5: Mean data waiting time vs. Data traffic intensity

4.5 Summary

In this chapter, we propose an approximate model for burst-switched integrated voice/data traffic. We decompose the original three-dimensional Markov process into two two-dimensional Markov process, where the link between the processes is represented by a probability on each transition to simplify the analysis. Each two-dimensional Markov process corresponds to one performance measure of interest. The relative frequency of fluctuations between voice and data traffic determines the order in which these two Markov processes are solved with an approximate solution being used for the process solved first. Numerical examples corresponding to interactive data traffic and file transfer data traffic are given and validated by simulation. The good agreement found shows that this model has the capability to predict the system performance well.

However, we did not consider the case where the voice fluctuation is comparable to the data fluctuation. In this case, the approximations for P_1, P_2 and P_3 described do not work well since the steady state assumption does not exist. One way to solve this problem is to use "*iteration*", i.e., we initialize P_1 (or P_2, P_3) first to solve the first (second) Markov chain, and use the result to obtain P_2 and P_3 (P_1) to solve the second (first) Markov chain. Then the result can be used to recompute P_1 (P_2 and P_3) and the process repeated until the result converges. This iteration method can also be applied to the two extreme cases described above but the result will be similar, since the steady state assumption is valid.

Chapter 5

An Integrated Voice/Data Protocol for Local Area Networks

Serving integrated voice/data traffic over a local area network has received much attention in recent years. Although many multiple access protocols have been developed for data transmission, they are typically not suitable for voice due to the different traffic characteristics and service requirements. In this chapter, we develop a protocol for integrated voice/data traffic on a contention based bus-type local area network. Speech Activity Detectors (SAD) and a movable boundary mechanism are used to achieve high channel utilization. Voice and data performance are also studied by an analytical model.

5.1 Introduction

Basically, most local area networks are designed for data communications. In a bus-type distributed local area network, many multiple access protocols for data transmission have been developed [1,53,58,35,59,34,9]. However, they are not suitable for integrated voice/data traffic since voice and data have different traffic characteristics and service requirements. Recall that voice users generate stream-liked traffic in which packets have fixed interarrival times and they require bounded delay although some corruption or loss of received signals is tolerable. On the other hand, data users generate aperiodic, bursty traffic and, although they can tolerate some delay, correct and complete transmission is required. In the set of contention-based protocols, CSMA[58] (or CSMA/CD[59]) has shown its capability to provide high efficiency for data transmission, but it is not good for voice transmission since collisions and collision-resolution are too time consuming and unpredictable to satisfy the delay constraint of voice traffic. Demand assignment protocols (e.g. MSAP[34], BRAM[9]) may be appropriate but they need to know the exact number of users to make assignment which introduces overhead. Thus, to serve integrated voice/data traffic on a local area bus network, development of new protocols to satisfy the above requirements is needed.

Nutt and Bayer [49] and DeTreville [12] studied combined voice/data loads on Ethernet via simulation. They demonstrate the capability of a standard Ethernet to support voice conversations under various data loads. The results show that CSMA/CD networks are appropriate for transmitting integrated traffic, provided that the network is not too heavily loaded. Meditch and Zhao [46] proposed a framed TDMA/CSMA protocol in which voice call requests and data packets access the channel via slotted, nonpersistent CSMA and each successful call request is assigned a slot in consecutive frames to transmit its packets. This slot is assigned until the end of the call. They analyzed the channel utilization by varying the maximum number of voice circuits the channel can accommodate. Chlamtac and Eisinger [8] considered a system in which voice and data share slots in each cycle with a fixed boundary between the two classes of traffic. Voice calls are transmitted based on TDM slot allocation and data packets are served using the CSMA/CD protocol. The effects of cycle length and the percentage of bandwidth dedicated to voice transmissions on data delay are investigated. Goel and Elhakeem [22] suggested a FARA/CSMA-CD protocol (FARA:Frame Adaptable Reservation Aloha). In their design, voice users follow a modified version of Reservation Aloha, but CSMA-CD is the channel access policy adopted by data users. Four different pilot tones are used to achieve an adaptive bandwidth allocation strategy, which allows data users to recover the bandwidth unused by the voice users. Of course, more hardware is needed to handle the pilot tones. Voice and data performance are studied and the results show an efficient utilization of the channel bandwidth.

All the work described above has been done under the assumption that voice traffic is served on a *call* basis, which means that voice users are served as in a circuit switched system, once a call request is accepted it can be completed without interruption since dedicated resources are allocated. However, typical speech alternates between talkspurt and silence periods, where the talkspurt periods occupy only about 45% of a call [5]. Thus, at least 50% bandwidth is wasted during the silence periods. To overcome this disadvantage, voice nodes are usually equipped with a *Speech Activity Detector* (SAD) which monitors the output of the encoder such that packets are only generated during talkspurt periods. It is expected that using SAD can double the maximum channel utilization.

In this chapter, we develop a protocol suitable for integrated voice/data traffic on a contention-bus local area network with a SAD in each node. The presentation is organized as follows. The channel structure and voice/data protocol are described in section 5.2. In section 5.3, we analyze voice throughput and delay by solving a two dimensional Markov chain. Based on these results, we evaluate data performance and investigate the effects of voice on data. Numerical results are obtained and discussed in section 5.4, and some concluding remarks are made in section 5.5.

5.2 System Environment

We consider a bus-type packet-switched local area network and assume there are N voice nodes and a large number of data nodes. There are several important considerations for integrated voice/data traffic networks. First, voice should have priority over data since it must be transmitted within its time constraint, otherwise it should be dropped, whereas data can wait. Second, some bandwidth should be reserved for data to prevent infinite delay due to heavy voice traffic. Since the voice packets during a talkspurt are generated on a regular basis, i.e., they have a constant interarrival time, it does not make sense to let each individual packet in a talkspurt contend for a transmission slot. It is more efficient to treat a talkspurt as a whole, which means that once a packet succeeds in a transmission slot, we should guarantee that all the following packets of this talkspurt also get service. Thus, some reservation mechanism is needed. To satisfy these considerations, we developed the following protocol on top of a corresponding channel structure.

5.2.1 Channel Structure

The channel structure is shown in Figure 5.1 and described below:

1) The channel is organized into frames of fixed duration F, where F is the time between two consecutive voice packets from some source. A portion of the frame, denoted by F_d, is reserved for data; the rest, denoted by F_v, is allocated to voice.

2) A frame is divided into slots of fixed length R_s. There are M_v slots in F_v and M_d slots in F_d.

3) A slot is again subdivided into minislots adequate for a collision to be detected within one minislot.

4) In a voice slot, the first minislot is for reservation of the slot by the user who occupied the corresponding slot in the previous frame. The second minislot is used to indicate that there are voice competitors, i.e., new voice users transmit in this minislot. The use of remaining minislots depends on the status of these two minislots.

5) If the first minislot is set, this slot is used for continuous voice transmission only, otherwise, the operation is determined by the second minislot:

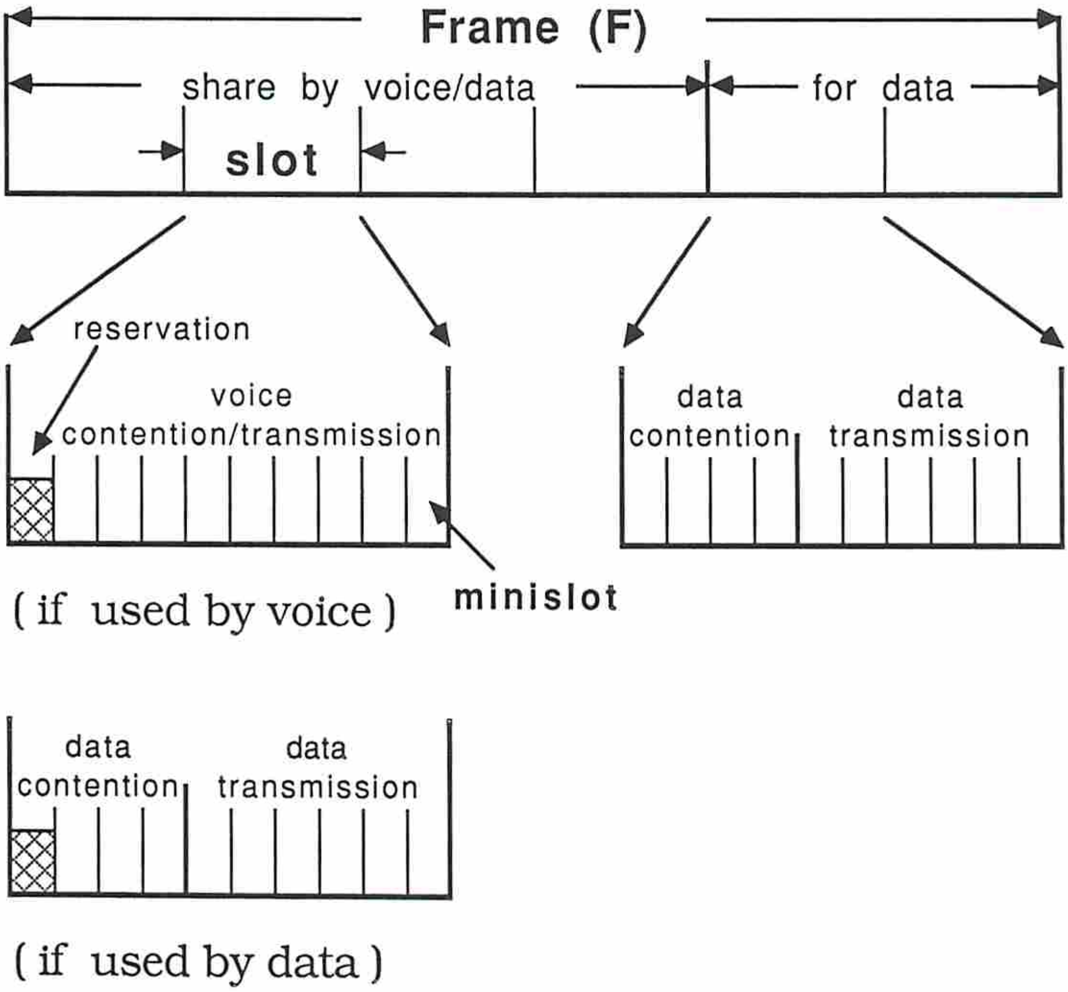


Figure 5.1: The channel structure for the protocol

- If the second minislot is not empty, i.e., at least one voice competitor exists, all the remaining minislots are reserved for voice competitors to compete for the transmission right of this slot time.
- If it is quiet, i.e., no voice competitors, the slot is scheduled for data users. In the first m_d minislots of the remaining minislots, they make an announcement, then the successful one, if any, transmits its packet.

6) In a data-slot, the first $m_d + 2$ minislots are used for contention, the rest of the slot is used for the data packet if there is a success.

From the above description, three things can be observed. First, we maintain a minimum bandwidth F_d in a frame for data users even under heavy voice traffic. Second, voice has priority over data since it always competes first in voice slots. Third, voice packets are slightly larger than data packets. For example, in a one kilometer bus-type local area network (i.e., each minislot has a minimum duration of 5 μs), if we let the slot length be 1 millisecond and take m_d to be 10 minislots, then 998 minislots could be used to transmit a voice packet but only 988 minislots for a data packet.

5.2.2 Protocol

The protocols for voice and data are as follows:
When a voice node has a new message, it waits until the beginning of the next frame and starts to compete for a slot.

1. In a slot, it first senses the reservation minislot *busy* or *idle.*
2. If the slot is *busy*, i.e., reserved, it waits until the next slot.
3. If the slot is *idle*, it sets the second minislot to make a voice reservation. Then, it randomly chooses a minislot to send a request signal and listens to the channel.
4. if a *collision* occurs, the node waits until the next slot in this frame and repeats the process.
5. if no collision is detected, the node keeps sending the request signal in each of the following minislots to jam other backlogged nodes. This guarantees that it is the only victor in competition.
6. If it does not succeed in this frame, the node drops the first packet in the input buffer and competes again in the next frame. Thus, in the worst case, the queueing delay of a transmitted packet will not exceed two frames, which satisfies the voice traffic time constraint. Overloading will lead to "front-end freeze-out" [41].

7. Once the node has competed successfully, it makes a reservation in the first minislot of the corresponding slot of each succeeding frame until the buffer is empty.

When a data node has a packet to transmit (or retransmit), it waits until the beginning of the next slot.

1. If this slot is in the subframe F_v, then it checks the first minislot. If this is set, the node waits a random time and starts again. If it is not set, the node senses the second minislot.
2. If the second minislot is busy, the node delays as it did for a busy first minislot. Otherwise, it attempts to acquire the slot by competing in the first m_d minislots. If the node succeeds, it transmits its packet. Otherwise, it delays a random time.
3. If the user falls into a data slot, it competes in the first $m_d + 2$ minislots, rescheduling if unsuccessful.

5.3 Analysis

To study the behavior of this system, we make several assumptions:

1) The voice nodes are always active and alternate between silence and talkspurt periods. Both periods are exponentially distributed with mean α and β seconds, respectively.

2) Each talkspurt is decoded and assembled into a stream of packets, a *voice message.*

3) Arrival of data packets is governed by a Poisson process.

4) Each node (voice/data) has a capability to detect a collision in which it is involved within one minislot. We assume that the length of a minislot is equal to the maximum propagation delay in this system.

5.3.1 Voice Performance Analysis

Since data users only use the *unused* slots in the voice subframe (in addition to the slots in the data subframe), they do not affect the performance of voice traffic. Therefore, we can analyze the voice traffic without considering the data traffic.

Under the above assumptions, we see that each voice user is in one of three states - *idle, backlogged* or *transmitting.* Idle users just do not have new arrivals. Backlogged users are trying to acquire a slot to transmit their message. Transmitting users are sending their messages as well as getting new packets. The behavior of voice traffic can be represented by a bivariate Markov process $(\mathcal{B}, \mathcal{T})$, $0 \leq \mathcal{B} \leq M_v$, $0 \leq \mathcal{T} \leq N$, where $\mathcal{B}$ is the number of backlogged users and $\mathcal{T}$ is the number of transmitting users at the beginning of a frame. To compute the transition probabilities, we need to know how many users are in each state. Define

$$\begin{aligned} N_e &\triangleq \text{ number of } \textit{idle} \text{ voice users} \\ N_b &\triangleq \text{ number of } \textit{backlogged} \text{ voice users} \\ N_t &\triangleq \text{ number of } \textit{transmitting} \text{ voice users} \end{aligned}$$

We assume that each user can make at most one state transition during a frame. Thus, at the end of this frame, an idle user can be either a backlogged user or still in the idle state; a transmitting user can become an idle user or remain in the transmitting state; but a backlogged user could switch into any of these three states. A silent user generates an arrival in a frame with probability g and let q be the probability that a user in talkspurt terminates its talkspurt period within the frame. For exponentially distributed silence and talkspurt periods, g and q can be easily found from the mean value of the talkspurt and silence periods, respectively, by

$$g = 1 - e^{-\lambda F}$$

$$q = 1 - e^{-\mu F}$$

Define

$$\begin{aligned} K_{eb} &\triangleq \text{ the number of users that switch from } \textit{idle} \text{ to } \textit{backlogged} \\ K_{bt} &\triangleq \text{ the number of users that switch from } \textit{backlogged} \text{ to } \textit{transmitting} \\ K_{be} &\triangleq \text{ the number of users that switch from } \textit{backlogged} \text{ to } \textit{idle} \\ K_{te} &\triangleq \text{ the number of users that switch from } \textit{transmitting} \text{ to } \textit{idle} \end{aligned}$$

and their corresponding density functions: $\psi_{K_{eb}}$, $\psi_{K_{bt}}$, $\psi_{K_{be}}$ and $\psi_{K_{te}}$, respectively. Then, for a given set $(N_b,\ N_t,\ N_e)$, we have

$$\begin{aligned} \psi_{K_{eb}}(k) &= C(N_e, k) \cdot g^k (1-g)^{N_e - k} && , 0 \leq k \leq N_e \\ \psi_{K_{te}}(k) &= C(N_t, k) \cdot q^k (1-q)^{N_t - k} && , 0 \leq k \leq N_t \\ \psi_{K_{be}|K_{bt}}(k) &= C(N_b - K_{bt}, k) \cdot q^k (1-q)^{N_b - K_{bt} - k} && , 0 \leq k \leq N_b - K_{bt} \end{aligned}$$

where $C(x, y)$ is the number of ways of selecting y objects from x objects and is equal to $\dfrac{x!}{(x-y)!y!}$.

The density function $\psi_{K_{bt}}(\cdot)$ is hard to obtain because of the complicated competition protocol, but an approximation can be made. It is due to the fact that, in a typical local area network, the ratio of propagation delay to the transmission time is usually very small, thus we have a large number of minislots in one slot. In this environment, it can be shown (see Appendix B), that the service behavior of voice traffic is just like an asynchronous TDM system. Therefore, for a given number of reserved voice slots in this frame, $N_t^{'}$, the density function $\psi_{K_{bt}}$ is approximated by

$$\psi_{K_{bt}}(K_{bt}=k) = \begin{cases} 1 & , k = min\{N - N_t^{'},\ N_b\} \\ 0 & , \text{otherwise} \end{cases}$$

It is noted here that $N_t^{'}$ may not be equal to N_t, the number of transmitting voice users of this frame - it could be more. The reason is that each transmitting packet incurs at least one frame delay, thus the user occupying a slot in the current frame may have switched to idle in the previous frame. Now, how many such users are there? It really depends on the number of transmitting users in the previous frame. Since the frame length is small compared to a talkspurt or silence period, it is observed that, within a frame, transitions will not occur often. It is very likely that the state of current frame is the same as in previous frame, thus we can approximate $N_t^{'}$ from current state information by $N_t^{'} = N_t + \nu$, where ν is the number of idle voice users who were transmitting users in the previous frame. The density function of ν, $\psi_\nu(\cdot)$, is given by

$$\psi_\nu(l) = \frac{C(N_t, l) \times q^l (1-q)^{N_t - l}}{\sum\limits_{i=0}^{\eta} \psi_\nu(i)} \qquad , 0 \le l \le \eta$$

where $\eta = min\{N_t,\ N - N_t - N_b\}$, since ν cannot be more than the number of idle voice users in this frame.

Let $P_{SS'}$ be the transition probability from state $S = (b, t)$ to the state $S^{'} = (b^{'}, t^{'})$, then it is computed by

$$P_{SS'} = \sum_{K_{eb}} \sum_{K_{te}} \psi_{K_{eb}} \psi_{K_{te}} \left(\sum_{\nu} \psi_\nu \sum_{K_{bt}} \psi_{K_{bt}|\nu} \sum_{K_{be}} \psi_{K_{be}|K_{bt}} \right)$$

where K_{eb} , K_{te}, K_{bt} and K_{be} are governed by

$$b^{'} = b + K_{eb} - K_{bt} - K_{be}$$

$$t^{'} = t + K_{bt} - K_{te}$$

After finding all transition probabilities, we can solve this Markov chain by $\pi = \pi\tilde{\mathbf{P}}$ to compute the equilibrium state probabilities π_{bt}, $0 \le b \le N$ and $0 \le t \le M_v$, where $\tilde{\mathbf{P}} = [P_{SS'}]$ is the transition matrix and π=$[\pi_{bt}]$ is the state probability vector. Then the expected number of backlogged users, $\overline{b}$, and the average number of occupied slots per frame, $\overline{t}$, can be evaluated as

$$\overline{b} = \sum_{b=0}^{N} \sum_{t=0}^{M_v} b \cdot \pi_{bt}$$

$$\overline{t} = \sum_{b=0}^{N} \sum_{t=0}^{M_v} (t + \overline{\nu}_{bt}) \cdot \pi_{bt}$$

where $\overline{\nu}_{bt}$ is the mean value of ν for a specified state (b, t) and is given by

$$\overline{\nu}_{bt} = \sum_{l=0}^{min\{t, N-t-b\}} l \cdot \psi_{\nu}(l)$$

The packet throughput of voice traffic, defined as the number of transmitted packets per frame time, is thus given by

$$S_v = \frac{\overline{t}}{N}$$

Another interesting and important measure of voice traffic is the average packet loss probability ϕ, which is defined as the probability that a packet is discarded. From the model, it is straightforward to calculate the average number of lost packets per frame by

$$\overline{d} = \sum_{m=0}^{M_v} \sum_{n=0}^{N_v} max\{0,\ b_i - (M_v - t_i - \overline{\nu}_{mn})\} \cdot \pi_{mn}$$

then ϕ is given by

$$\phi = \frac{\overline{d}}{\overline{b} + \overline{t}}$$

5.3.2 Data Performance Analysis

The data throughput consists of two parts. One is from the subframe dedicated to voice, F_v, and the other is from the subframe reserved for data, F_d. Since the data competitors in each slot are new arrivals and the retransmissions from the previous slot, it is clear that the statistics of data transmissions in each slot of a frame are mutually independent. As discussed in the voice analysis, if there are k competitors in a slot, then the probability that a success occurs during this slot in subframe F_v and F_d are $1 - \psi(k, m_d)$ and $1 - \psi(k, m_d + 2)$, respectively, where $\psi(\cdot, \cdot)$ is defined in Appendix B. We define P_{d1} to be the probability that a data packet succeeds in a slot of F_v and P_{d2} to be the probability that a data packet succeeds in a slot of F_d. Let G_d be the offered data traffic rate (packets/slot), then we have

$$P_{d1} = \sum_{k=0}^{\infty} [1 - \psi(k, m_d)] \frac{G_d^k}{k!} e^{-G_d}$$

$$P_{d2} = \sum_{k=0}^{\infty} [1 - \psi(k, m_d + 2)] \frac{G_d^k}{k!} e^{-G_d}$$

A data packet will be scheduled into the subframe F_v with probability F_v/F and into the subframe F_d with probability F_d/F, thus the data throughput is obtained from

$$S_d = (1 - S_v) \frac{F_v}{F} P_{d1} + \frac{F_d}{F} P_{d2}$$

To analyze the data delay, it is found that the probability of scheduling a successful data packet is equal to S_d/G_d, i.e., data delay is geometrically distributed with mean G_d/S_d. During each scheduling, one of the following three cases will happen:

case 1 : the packet sensed the slot busy in F_v.

case 2 : the packet failed in competition.

case 3 : the packet got through.

According to the data protocol, an arriving data packet must wait until the next slot, hence it will experience half of a slot delay. The incurred average delays for these three cases are given by

$$\begin{aligned} d_1 &= \frac{R_s}{2} + \Delta + X \\ d_2 &= \frac{R_s}{2} + [\frac{F_v}{F} m_d + \frac{F_d}{F}(m_d + 2)]\Delta + X \\ d_3 &= \frac{R_s}{2} + R_s \end{aligned}$$

the average number of schedulings of these cases can be computed by

$$\begin{aligned} n_1 &= \frac{G_d}{S_d} \cdot \frac{F_v}{F} S_v \\ n_2 &= \frac{G_d}{S_d}[\frac{F_d}{F} + \frac{F_v}{F}(1 - S_v)] - 1 \end{aligned}$$

and the average delay of a data packet is simply obtained from

$$D_d = \sum_{i=1}^{3} n_i \cdot d_i$$

5.4 Numerical Results

In the following numerical examples, we consider a one kilometer local area network with nodes (voice/data) sharing a 1 Mbps bandwidth channel. The maximum propagation delay in this network is thus 5 microseconds (or 5 bits length). We assume that there is a 64 Kbps digitizer in each voice node to generate packets. The frame length is chosen to be 16 milliseconds and we take fifteen slots per frame which corresponds to 1067 bits in each voice packet. By a simple computation, it is found that the number of minislots for voice competition in a slot time is 211. In the Appendix B, it is shown that even in the worst case (which occurs when there are two competitors) the success probability is about 0.9953, which is very close to 1. Thus, the TDM approximation used here to find the transition probabilities for voice traffic is quite reasonable.

Figure 5.2 shows the voice packet throughput under different voice populations (i.e., voice load). We also vary M_d - the number of slots reserved for data traffic in each frame.

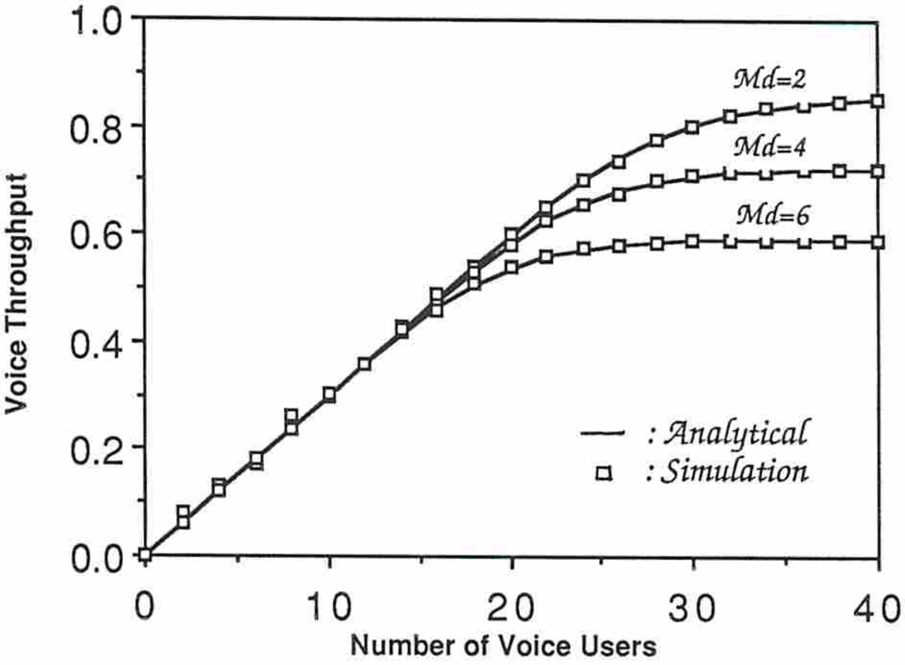


Figure 5.2: Voice packet throughput vs. Number of active voice users

The three curves correspond to M_d=2, 4 and 6, respectively. When the voice load is light, all three curves are consistent, since, under this circumstance, the number of available slots is always greater than the number of arrivals. As the load grows, these three curves saturate to the maximum throughput that the system can provide, which is 0.867, 0.733 and 0.6, respectively.

In Figure 5.3, we demonstrate the average packet loss probability ϕ versus the number of voice users in system. We use the same value for M_d as above, which implies that M_v = 13, 11 and 9. Since the acceptable packet loss probability in a voice transmission system is less than or equal to 2% [28], it is observed, from the figure, that the maximum number of voice users the system can support in each case is almost twice the number of slots in a frame. A simulation program was written to validate the performance model and the results are also shown in Figure 5.2 and 5.3 to compare with the analytic results. It is shown that our approximate model can precisely predict the system performance.

Figures 5.4 and 5.5 illustrate the effect of voice traffic on data traffic.

We take m_d, the number of minislots for data competition in a voice slot, to be 10 and plot data throughput and data delay under various voice loads. The three curves correspond to different values of M_d as above. It is observed that the data throughput goes up as the voice load decreases, which means that the boundary of allocated slots for voice and data in a frame is dynamically adjusted by the variation of the voice load, thus the channel bandwidth is efficiently utilized. We define the channel utilization as

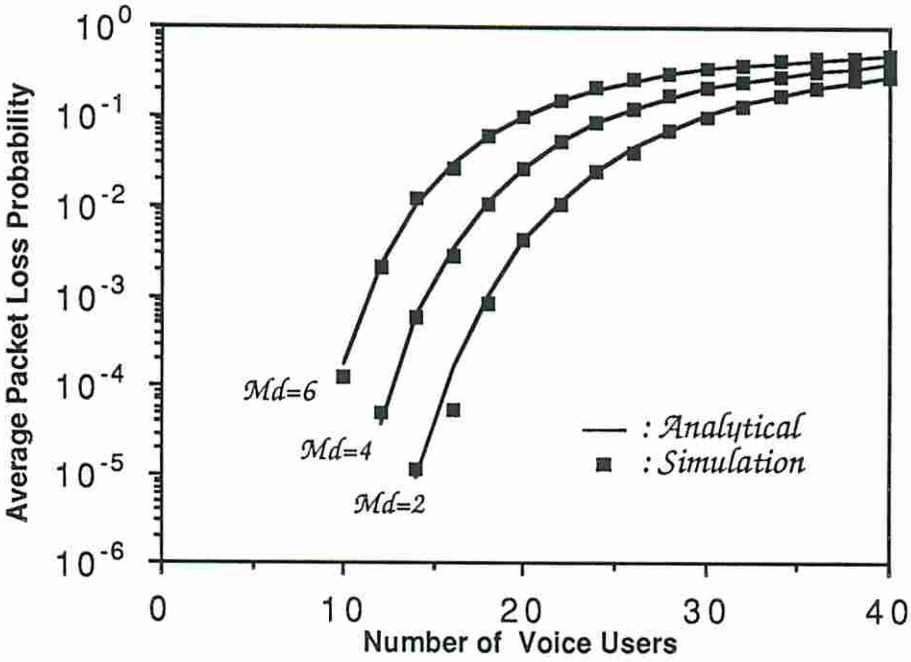


Figure 5.3: Average packet loss probability vs. Number of active voice users

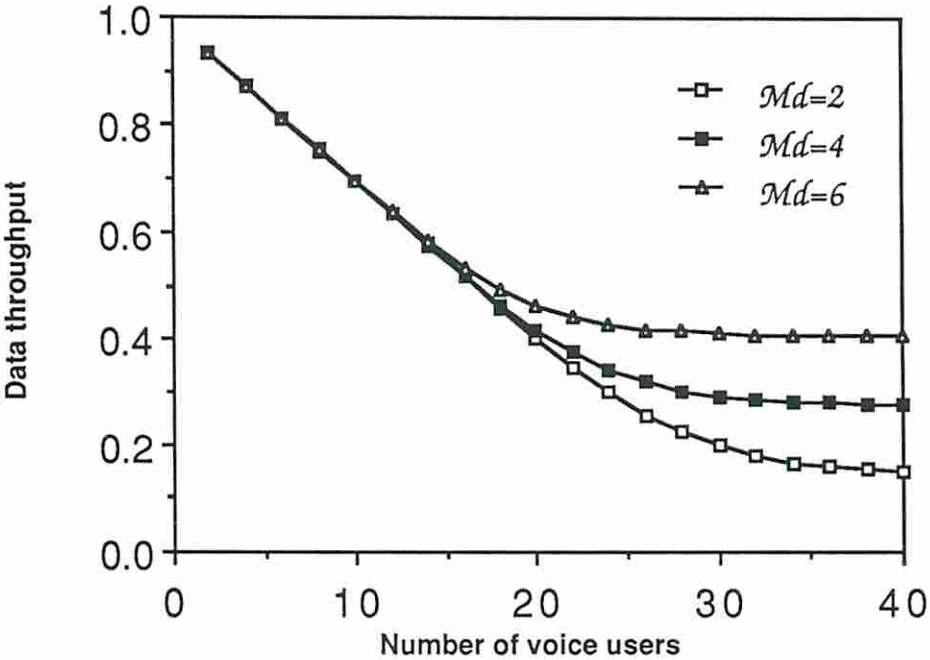


Figure 5.4: Data packet throughput vs. Number of active voice users (m_d=10, G_d=10 packets/slot)

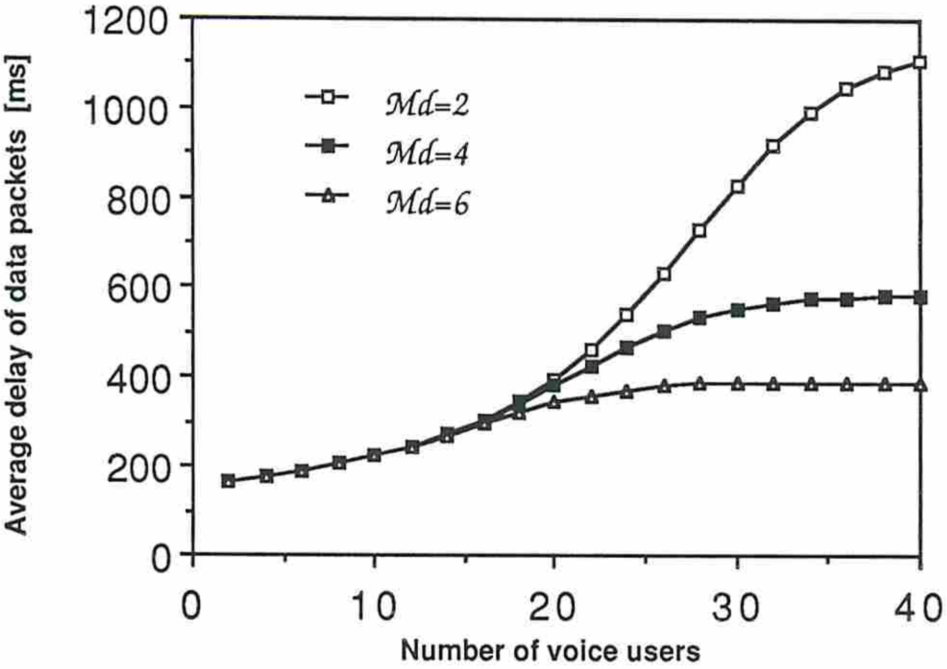


Figure 5.5: Average data packet delay vs. Number of active voice users (m_d=10, G_d=10 packets/slot)

the fraction of of minislots used for transmission in a frame. In Figure 5.6, we see that the utilization is almost flat no matter what the voice load. From the protocol, we know that there are two minislots overhead for transmitting a voice packet and twelve minislots overhead for transmitting a data packet, thus more data transmissions lead to higher overhead. This explains why the utilization is a little lower when the voice load decreases or the bandwidth reserved for voice shrinks.

Now, we set M_d=4, fix the value of total number of voice users to be 22 (the maximum voice population the system can support in this case), and examine the data performance under different data loading. The results are shown in Figures 5.7 and 5.8.

We see that the shapes of throughput and delay curves are very much like CSMA, this is no surprise since the data protocol is basically a variation of CSMA/CD.

Finally, we study the effect of varying the number of contention minislots in a slot, m_d, on data performance. Since a different value of m_d means a different data packet size, in order to be able to compare them in a unified manner we redefine data throughput to be the number of bits transmitted in a slot and data traffic load to be the number of bits arriving in a slot. We consider m_d = 10, 20 and 30 and show the results in Figure 5.9. It is seen that as m_d increases the system can achieve better throughput under heavier load but the maximum throughput becomes smaller, since a larger m_d implies more overhead.

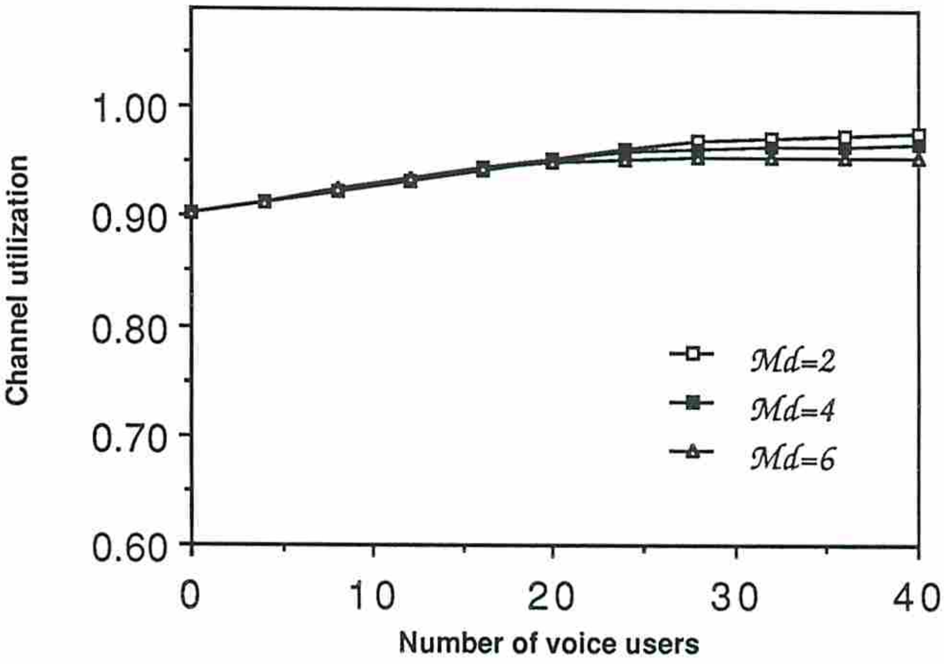


Figure 5.6: Channel utilization vs. Number of active voice users

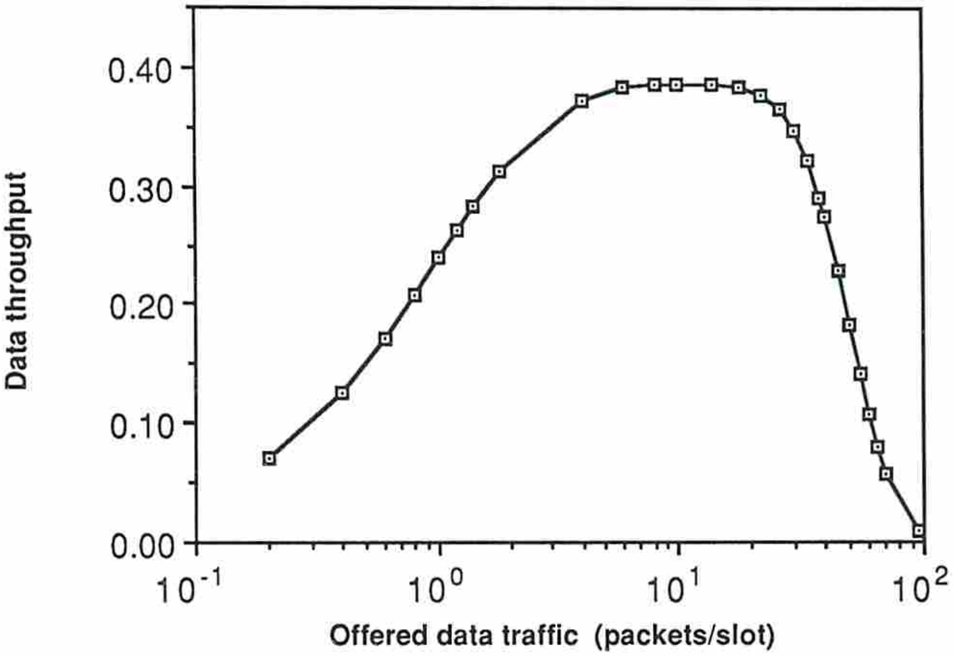


Figure 5.7: Data throughput vs. Offered data traffic (M_d=4, Number of voice users=22)

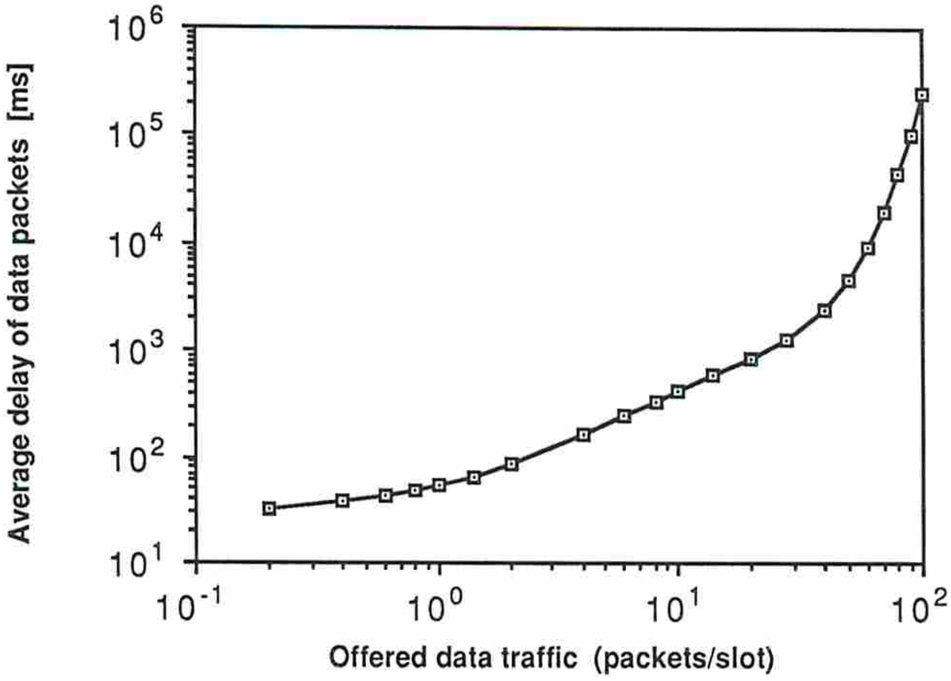


Figure 5.8: Average data packet delay vs. Offered data traffic (M_d=4, Number of voice users=22)

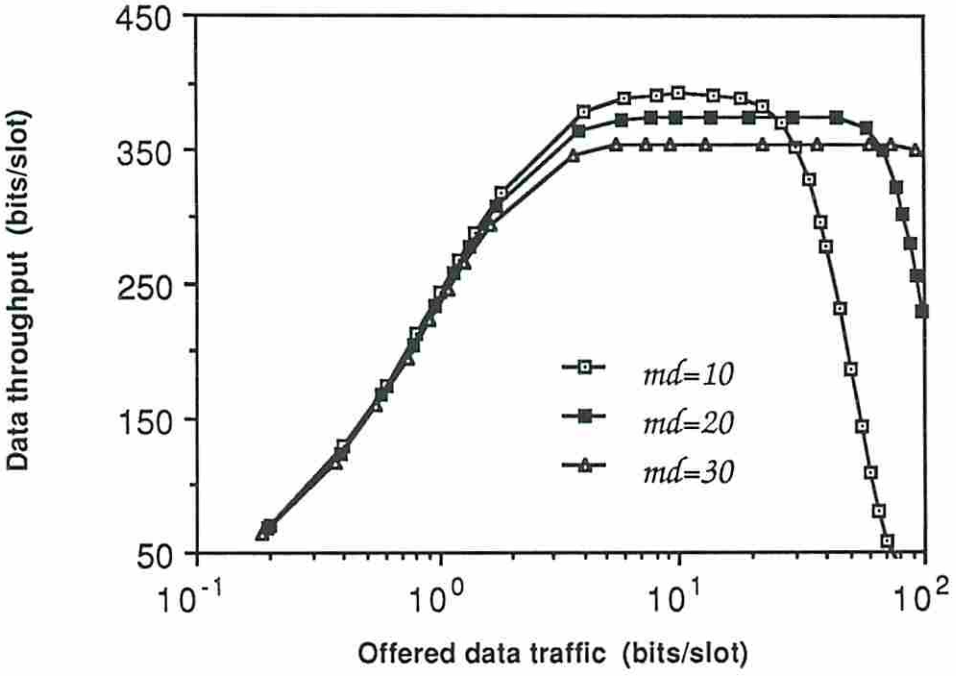


Figure 5.9: Data bits throughput vs. Offered data bits traffic (M_d=4, Number of voice users=22)

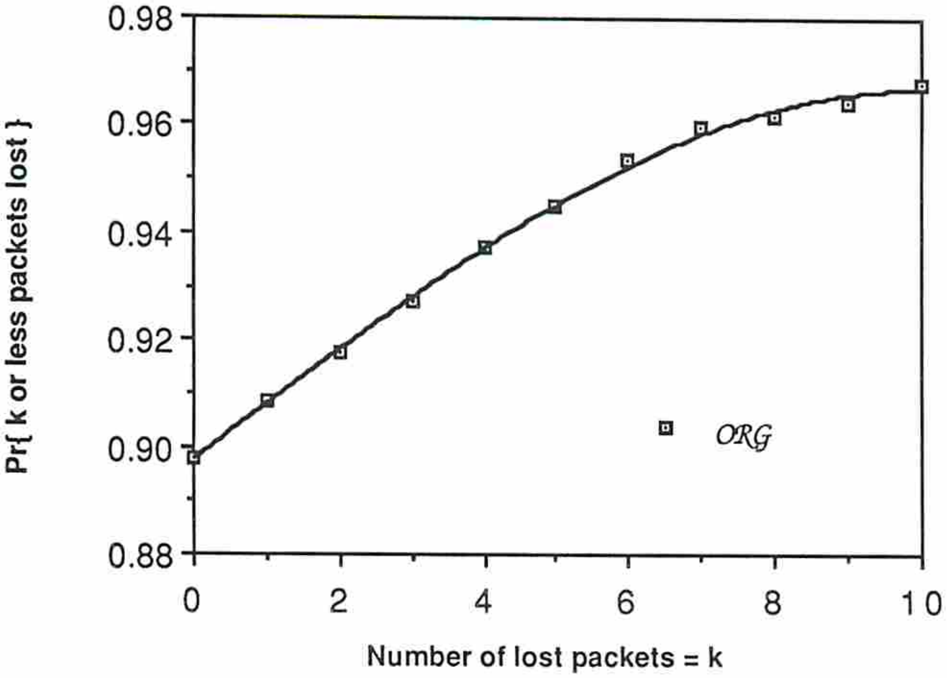


Figure 5.10: The original discarding algorithm (M_d = 2, number of voice users=22, $\phi = 0.01$)

5.5 Fairness Discarding Algorithm for Voice

The discarding algorithm used in the protocol can be summarized as follows: During a frame, a voice node in *backlogged* state competes for voice slots while it finds unoccupied slots and it drops the packet if it does not capture a slot by the end of this frame. It is very likely that a particular node loses its packet again and again since a talkspurt period is much longer than a frame interval, i.e., one user may experience terrible voice quality although the average packet loss is small. In Figure 5.10, we plot the percentage of talkspurts which discard k or fewer packets over all arrival talkspurts as a measure of fairness. We choose $M_d = 2$ and take the number of active voice users to be 22, which corresponds to 1% packet loss on the average. The results show that about 90% of talkspurts can get through without any packet loss but about 5.5% of talkspurts experience five or more packets discarded and about 3% of talkspurts have ten or more lost packets.. This situation can be improved by distributing the discarded packets over *all active* voice users in the current frame.

The basic idea of our fairness algorithm is that packets lost during a frame are not only distributed over the nodes in *backlogged* state but also distributed over those nodes who already have a slot (i.e., are in *transmitting* state). Each *backlogged* node gets an additional opportunity to take over an occupied slot as well as to compete for available slots during a frame.

The voice protocol is slightly modified such that it is adapted for those algorithms. For a voice user in *backlogged* state:

- At the beginning of a frame, it selects a voice-slot to play.
- If the slot is idle (i.e., the first minislot is not set), it competes as usual (and loses the chance to take over others in this frame).
- If that slot is occupied (i.e., the first minislot is set), then it sets the second minislot to attempt to take over the slot from its current occupant.
- If nobody else sets this minislot, then the node gets the right to transmit a packet in this slot.

For a voice user in *transmitting* state:

- It not only sets the first minislot but also controls the status of the second minislot.
- It sets the second minislot if it is not prepared to share this slot with others, otherwise, it keeps silent and senses that minislot.
- If the second minislot is idle or a collision, then the slot still belongs to him.
- If one and only one other node sets the second minislot, then the slot is taken over by that node and the original owner drops its packet.

5.5.1 The Fairness Algorithm

There are two questions remaining. One is how a *backlogged* node selects a slot to play, the other is when should a node allow its slot to be taken over. We introduce three options for our fairness algorithm. First, the way to select a slot to play could be either *random* or *round-robin*. In *random* selection, the node randomly chooses a slot in a frame. In *round-robin* selection, the node initially selects a slot randomly, then advances in a round-robin fashion from frame to frame until it captures a slot. Second, should the algorithms be used in any frame or only in frames with all slots occupied. The latter sounds more reasonable since transmitting nodes should have higher priority over others if there are still slots available in a frame. The problem is that the nodes does not know the status of the current frame in advance, but it is very likely that the status is the same as the previous frame, since the frame is very small compared to the system's fluctuation period. Third, a *threshold* for discarded packets can be set in each transmitting node such that no one can take over its slots if the number of lost packets has reached the threshold value. An infinite value for this threshold corresponds to no threshold, i.e., any number of dedicated slots can be taken over. In summary, we consider eight different schemes for the fairness algorithm:

- **RAN:** Random selection only.
- **RF:** Random selection with frame-usage detection.
- **RR:** Round-Robin selection only.
- **RRF:** Round-Robin selection with frame-usage detection.
- **RT:** Random selection with threshold.
- **RFT:** Random selection with frame-usage detection and threshold.
- **RRT:** Round-Robin selection with threshold.
- **RRFT:** Round-Robin selection with frame-usage detection and threshold.

A simulation program was written to study these schemes. The performance we measure is the distribution function of the number of dropped packets in each talkspurt, for a given average packet loss probability.

5.5.2 Numerical Example

We follow the example used in the previous section, i.e., 1 Mbps channel over one kilometer length cable, two slots for data per frame and 1% mean packet loss. In Figure 5.11, we consider the no threshold cases: *RAN*, *RF*, *RR*, *RRF* and compare them with the original discarding algorithm. We find that the *RRF* scheme is the best. It is also seen that the fairness schemes have a lower probability of losing 6 or more packets. This results in higher overall system performance if the system can tolerate 5 or fewer losses. In Figure 5.12, we plot the other four schemes: (i.e., *RT*, *RFT*, *RRT* and *RRFT*) and compare them with the original algorithm. The threshold we use is to not allow transmitting nodes to lose more than one packet. It is observed that the fairness schemes always do better than the original discarding scheme and that *RRFT* is the best of them. Next, we investigate the impact of the threshold on the distribution function. We take *RRFT* scheme as an example and vary the threshold from one to six. The results are shown in Figure 5.13. It is concluded that the higher the number of dropped packets (threshold) allowed, the better the performance. For example, if we allow five or fewer packets to be discarded (i.e., threshold=5), only 1.1% of talkspurts fail, compared to the original scheme which has 5.5% of talkspurts failing. For a given maximum number of lost packets and percentage of failing, Figure 5.14 depicts the maximum number of talking users that the system can support for *RRFT* (threshold = 5) and the original algorithm. It is shown that the fairness discarding algorithm allows more users to be supported.

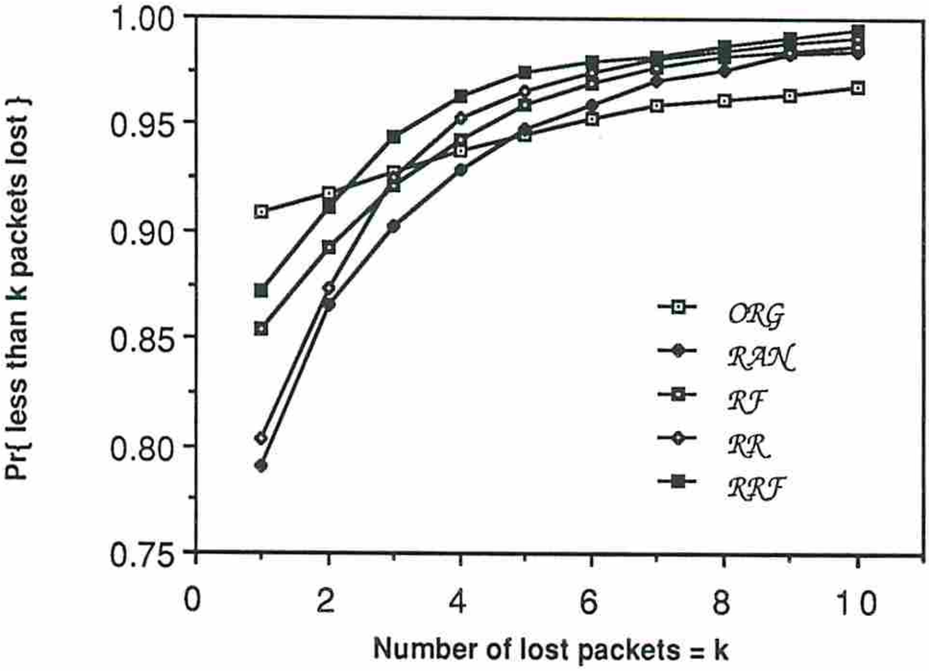


Figure 5.11: Fairness discarding Algorithms without threshold ($M_d = 2$, number of voice users=22, $\phi = 0.01$)

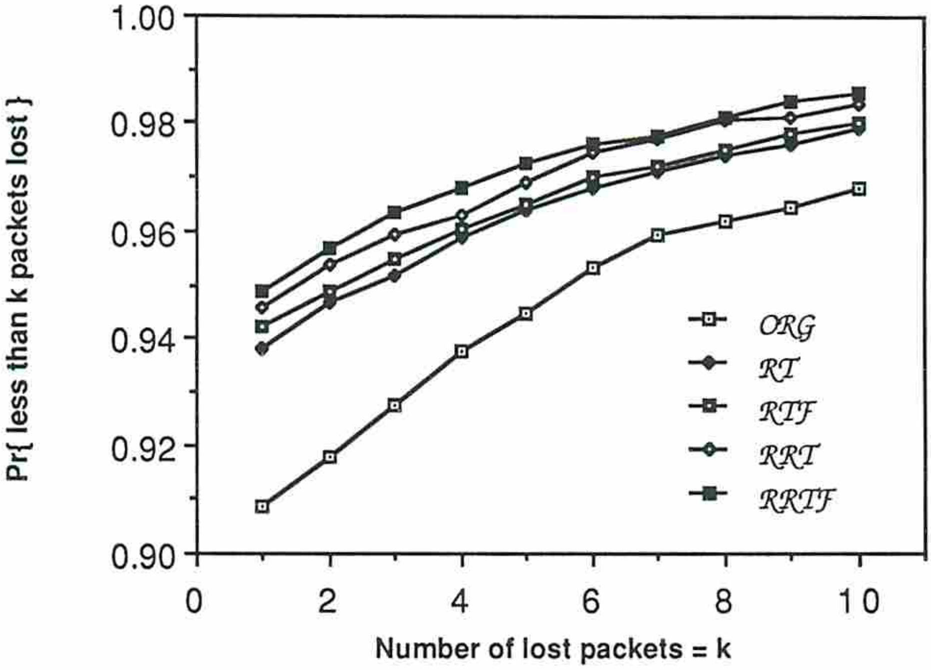


Figure 5.12: Fairness discarding Algorithms with threshold ($M_d = 2$, number of voice users=22, threshold=1, $\phi = 0.01$)

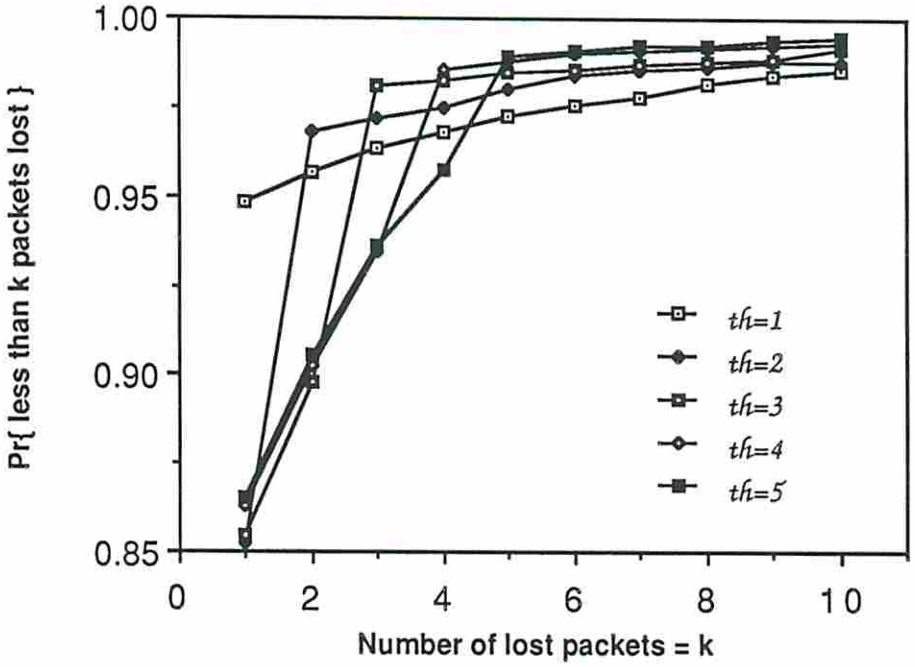


Figure 5.13: The effect of threshold on RRFT scheme ($M_d = 2$, number of voice users=22, $\phi = 0.01$)

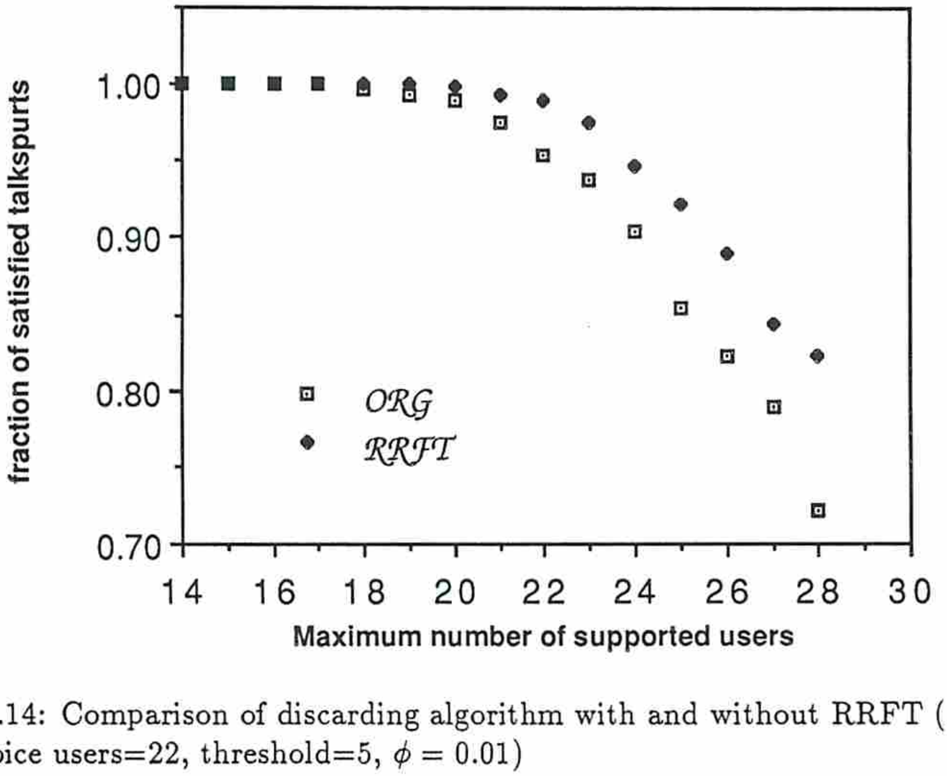


Figure 5.14: Comparison of discarding algorithm with and without RRFT ($M_d = 2$, number of voice users=22, threshold=5, $\phi = 0.01$)

5.6 Summary

In this chapter, we discuss a distributed protocol for integrated voice/data traffic on a local area bus network. The protocol uses announcement and reservation techniques. A two dimensional Markov model is used to estimate the performance of both traffic types. The performance of voice traffic is much like a centrally-controlled TDM system due to the short propagation delay properties of local area networks, and the data traffic behavior is similar to a CSMA-type system, since the data protocol is basically a variation of CSMA/CD. The interactive effects between voice and data traffic are also investigated. The protocol exhibits several desirable properties. It fulfills the requirements of both types of traffic. In addition, the unified voice and data access scheme simplifies the interface. The use of a dynamic boundary between voice and data bandwidth allocation achieves an efficient usage of the channel, and the protocol performs well for all values of traffic intensity. Simulation results are used to validate the performance model and show good consistency with analytic results. Fairness discarding algorithms for voice traffic are also studied to make discarding fair. By modifying the algorithm to distribute dropped packets over all currently active voice users the system can support additional users.

Chapter 6

Conclusions and Future Research

6.1 Conclusions

In this dissertation, we have focused on two topics. One is performance modeling of integrated voice and data traffic for various switching systems; the other is protocol design for integrated voice and data services.

In the first topic, we have developed two models for packet switching: one for delay-constrained voice traffic and one for integrated voice and data traffic. We also developed an approximate model for burst-switched integrated voice and data traffic.

The contributions are summarized as follows:

- An analytical model for delay-constrained voice traffic in a packet switching system has been developed. The model utilizes the frame structure, which preserves the Markov property and also allows the transient discarding behavior to be studied. Based on this model, we analyze the impact of the delay constraint, the packet size, and the activity detecting threshold on performance. We also proposed an algorithm to discard delayed packets in a fair manner.
- An approximate model to analyze data performance in a packet-switched integrated voice and data system has been developed. This one-dimensional approximate Markov model allows large systems to be studied with accurate results. Furthermore, it is easily extended to study the influence of *delay-constrained* voice traffic on data performance.
- An approximate model for analyzing the integrated voice and data traffic in a burst-switched node has been developed. The decomposition from the exact three-dimensional Markov into two two-dimensional Markov chains simplifies the model structure and analysis. Comparing with simulation results, the model shows good consistency.

In the second topic, we have developed an integrated voice and data protocol for a bus-type local area network, and also proposed an analytical model to measure its performance. The contribution of this work is:

- The protocol fulfills the requirements of both types of traffic and exhibits high efficiency in channel usage over a wide range of traffic intensities. A fairness discarding algorithm for voice traffic which balances dropped packets over all active users to improve system performance is introduced.

6.2 Future Research

Within the broad scope of this work, we can identify related issues for future research.

1. Our work basically focuses on the performance modeling of integrated voice and data traffic in a single switching node. However, when we consider the whole network, the model should becomes more complicated. What assumptions can be made and what approximations can be used to simplify the model for end-to-end performance analysis. Some work on a simple topology (a two-node tandem link) has been done [37,44]. However, for a more general network, this problem is still open.
2. Future communication networks are expected to provide services for various types of traffic, not only voice and data but also video and facsimile. To handle so much information through a network, the transmission media must be fibre-optic with several Gb/s bandwidth. Protocol design for these multiple traffic types in high speed networks and models to measure their performance remain to be explored.

Appendix A

In the following, we analyze the probability density functions of scheme 1 and scheme 2 mentioned in section 2.2. Before the analysis, we introduce three basic functions which are used frequently in the sequel.

F1) The ways to select n objects from m objects, denoted by C_n^m. It is simply given by

$$C_n^m = \frac{m!}{(m-n)!n!}$$

F2) The ways to put m objects into n boxes, denoted by $\Omega(m,n)$. It is obtained by

$$\Omega(m,n) = C_m^{m+n-1}$$

F3) The feasible solutions, $(X_1, X_2, \cdots, X_n)$, of the following equation set: given n non-negative integers X_i, $1 \leq i \leq n$, and

$$X_1 + X_2 + \cdots + X_n = m$$

$$X_1 + X_2 + \cdots + X_l \leq B + l, \quad 1 \leq l \leq n-1$$

where B is a non-negative integer and l, m and n are all natural numbers. It can be shown that the number of feasible solutions is

$$\Lambda(m,n,B) = \begin{cases} n & , m = 1 \\ \sum\limits_{i_1=0}^{\alpha(1)} \sum\limits_{i_2=0}^{\alpha(2)} \cdots \sum\limits_{i_{n-1}=0}^{\alpha(n-1)} 1 & , m > 1 \end{cases}$$

where

$$\alpha(k) = \begin{cases} min(1+B,m) & , k = 1 \\ min(k+B,m) - \sum\limits_{j=1}^{k-1} i_j & , 2 \leq k \leq n-1 \end{cases}$$

In addition, we assume that we are at the beginning of some arbitrary frame (length γ slots) in state $S = (t, b)$, which implies that t arrivals will occur in this frame and there are b packets already in the buffer at the beginning of the frame. Since the arrival time of a packet may occur at any instant of the interval, we assume that it is uniformly distributed over the frame. In the following, we define the arrival pattern in the frame to be a γ-tuple vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\gamma)$, where α_i be the number of arrivals that occur in the ith slot and $\alpha_1 + \alpha_2 + \ldots + \alpha_\gamma = t$.

A.1 The density function $\Psi_{\mathcal{R}}$

Recall that the random variable $\mathcal{R}$, $0 \leq \mathcal{R} \leq \gamma - b$, is the number of new arrivals that are served in a fashion as described in case 2 of both schemes. We want to find its density function $\Psi_{\mathcal{R}}(r)$. Since $b < \gamma$ in this case, the first b slots are reserved for the b packets which are already in buffer and the remaining slots can be used by t arriving packets.

We first look at the probability that less than r arrivals are served in the interval. We order the arriving packets from 1 to t and define a_i, $1 \leq i \leq t$, to be the arrival time of the ith packet. Let τ_j, $1 \leq j \leq \gamma$, be the beginning moment of jth slot. Then, the possible cases where less than r arrivals are served are:

(1) $\{a_1 \geq \tau_{\gamma-r+1}\}$

(2) $\{a_1 < \tau_{\gamma-r+1}\} \cap \{a_2 \geq \tau_{\gamma-r+2}\}$

(3) $\{a_1 < \tau_{\gamma-r+1}\} \cap \{a_2 < \tau_{\gamma-r+2}\} \cap \{a_3 \geq \tau_{\gamma-r+3}\}$

....

(r) $\{a_1 < \tau_{\gamma-r+1}\} \cap \{a_2 < \tau_{\gamma-r+2}\} \cap \cdots \cap \{a_{r-1} < \tau_{\gamma-1}\} \cap \{a_r \geq \tau_\gamma\}$

In general, we can express these r cases as:

$$\begin{array}{ll} \{a_1 \geq \tau_{\gamma-r+1}\} & , \; k = 1 \\ \bigcap_{i=1}^{k-1} \{a_i < \tau_{\gamma-r+i}\} \cap \{a_k \geq \tau_{\gamma-r+k}\} & , \; 1 < k \leq r \end{array}$$

The probability that less than r arrivals are served in the frame is equal to the ratio of the number of arrival patterns corresponding to these r cases to the total number of possible arrival patterns.

In case 1, all arrivals occur during the last r slots, so the number of arrival patterns is equivalent to the ways to put t objects into r boxes. It is given by $\Omega(t,r)$. For case k, $1 < k \leq r$, we split the arrivals into two groups: the first $k-1$ arrivals occur in the first $\gamma - r + k - 2$ slots, subject to the condition that the ith arrival occurs before the $(\gamma - r + i - 1)$th slot; the remainings $t - k + 1$ occur in the last $r - k + 1$ slots. Let $\eta = \gamma - r + k - 2$, we can restate the conditions for the first group by the following mathematical form:

$$X_1 + X_2 + \cdots + X_\eta = k - 1$$

$$X_{\eta-l} + X_{\eta-l+1} + \cdots + X_\eta \leq l + 1 \quad , \; 0 \leq l \leq k-2$$

where X_i is the number of arrivals during the ith slot. This is the same form as F3. For the second group, we must put the $t-k+1$ arrivals into $r-k+1$ slots. Thus, the number of possible patterns corresponding to these r cases is obtained by

$$\begin{array}{lcl} \textit{Case} \;\; 1 & : & \Omega(t,r) \\ \textit{Case} \;\; k & : & \Lambda(k-1, \gamma - r + k - 2, 0)\Omega(t-k+1, r-k+1) \quad , \; 1 < k \leq r \end{array} \tag{A.1}$$

Similarly, the total number of possible arrival patterns can be found by $\Omega(t,\gamma)$, then we have

$$Pr\{\mathcal{R} < r\} = \frac{\Omega(t,r) + \sum_{i=1}^{r-1} \Lambda(i, \gamma - r + i - 1, 0)\Omega(t-i, r-i)}{\Omega(t,\gamma)} \tag{A.2}$$

The probability of exactly r arrivals being served in the interval, $\Psi_{\mathcal{R}}(r)$, is thus given by

$$\Psi_{\mathcal{R}}(r) = Pr\{\mathcal{R} < r+1\} - Pr\{\mathcal{R} < r\} \tag{A.3}$$

A.2 The density function $\Psi_{\mathcal{D}_1}$

We have defined $\mathcal{D}_1$ to be the number of packets that are discarded in the fashion described in case 3 of scheme 1. The way to find its density function, $\Psi_{\mathcal{D}_1}(d)$, is very similar to $\Psi_{\mathcal{R}}$ discussed above. Let u $(= M - b)$ be the number of available spaces at the beginning of the frame, then only u of t packets can definitely get space in buffer, the remaining $t - u$ packets may or may not be discarded depending on the buffer status at their arrival instants.

Instead of finding the probability of discarding packets we compute the probability of successful packets from the remaining $t - u$ arrivals that also get into the buffer during this frame. Let $\mathcal{S}$ denote the number of arrivals having these properties, $1 \leq \mathcal{S} \leq \min\{t-u, \gamma\}$. Define a_i, $1 \leq i \leq t$, and τ_j, $1 \leq j \leq \gamma$, as before, then the possible cases for less than s successes are:

(1) $\{a_t < \tau_s\}$

(2) $\{a_t \geq \tau_s\} \cap \{a_{t-1} < \tau_{s-1}\}$

(3) $\{a_t \geq \tau_s\} \cap \{a_{t-1} \geq \tau_{s-1}\} \cap \{a_{t-2} < \tau_{s-2}\}$

....

(s-1) $\{a_t \geq \tau_s\} \cap \{a_{t-1} \geq \tau_{s-1}\} \cap \cdots \cap \{a_{t-s+3} \geq \tau_3\} \cap \{a_{t-s+2} < \tau_2\}$

Using the functions $\Omega(m,n)$ and $\Lambda(m,n,B)$ defined above, we find

$$Pr\{\mathcal{S} < s\} = \frac{\Omega(t, s-1) + \sum_{i=1}^{s-2} \Lambda(i, \gamma - s + i)\Omega(t-i\ , s-i-1)}{\Omega(t,\gamma)} \tag{A.4}$$

then the probability of exactly s successes can be obtained by

$$Pr\{\mathcal{S} = s\} = Pr\{\mathcal{S} < s+1\} - Pr\{\mathcal{S} < s\} \tag{A.5}$$

and $\Psi_{\mathcal{D}_1}(d)$, the probability of exactly d packets being discarded, is given by

$$\Psi_{\mathcal{D}_1}(d) = Pr\{\mathcal{S} = t - u - d\} \tag{A.6}$$

A.3 The density function $\Theta_{\mathcal{R},\mathcal{D}_1}$

Let $\mathcal{R}$ and $\mathcal{D}$ be defined as before. Given a pair (r,d), where $r \in \mathcal{R}$ and $d \in \mathcal{D}_1$, we combine the ideas of finding $\Psi_{\mathcal{R}}(r)$ and $\Psi_{\mathcal{D}_1}(d)$ to find $\Theta_{\mathcal{R},\mathcal{D}_1}(r,d)$, Since the arrivals getting service always occur before the arrivals being discarded, we compute the probability of the r arrivals by $\Psi_{\mathcal{R}}(r)$ (i.e. without considering d). Thus

$$\Theta_{\mathcal{R},\mathcal{D}_1}(r,d) = \Psi_{\mathcal{R}}(r)\Psi_{\mathcal{D}_1|\mathcal{R}}(d|r) \tag{A.7}$$

where $\Psi_{\mathcal{D}_1|\mathcal{R}}(d|r)$ is the probability of d arrivals being discarded conditioned on the first r arrivals getting service in this frame. In the following, all the events have this condition. Let $\nu(i|r)$ denote the probability that the rth arrival occurs in the ith slot and $\chi(d|i,r)$ denote the probability that d arrivals are discarded while the remaining $t-r$ packets arrive in the remaining $\gamma - i + 1$ slots with $M - (b-i)^+$ buffer spaces at the beginning of the ith slot, then we have

$$\Psi_{\mathcal{D}_1|\mathcal{R}}(d|r) = \sum_{j=1}^{\gamma} \nu(i|r)\chi(d|i,r) \tag{A.8}$$

To obtain $\nu(i|r)$, we first compute $N(i,r)$, which is defined to be the number of arrival patterns in which the rth arrival occurs in the ith slot. It is obvious, for $0 \leq i \leq \gamma - r$, that

$$N(i,r) = \Omega(r-1,i)\Omega(t-r,\gamma-i+1)$$

For $\gamma - r < i \leq \gamma$, it is not quite so straightforward, we first find the number of patterns with less than r arrivals getting service, $N^-(i,r)$, by using the same idea as in A.1.

$$\begin{aligned} N^-(i,r) &= \left[\Omega(r,\kappa) + \sum_{j=1}^{\kappa-1} \Lambda(j,\gamma-r+j-1)\Omega(r-j,\kappa-j)\right] \\ &\quad \cdot\Omega(t-r,\gamma-i+1) \end{aligned}$$

where $\kappa = i - \gamma + r + 1$. Similarly, we get $N^-(i,r+1)$ and then

$$N(i,r) = N^-(i,r+1) - N^-(i,r) \tag{A.9}$$

The probability $\nu(i|r)$ is simply obtained by dividing $N(i,r)$ by $\Omega(t,r)$ - the total number of arrival patterns.

To evaluate $\chi(d|i,r)$, we recall that $\Psi_{\mathcal{D}_1}(d)$ is the probability that d arrivals are discarded while t packets arrive during γ slots with $M-b$ buffer spaces at the beginning. By comparing with the definition of $\chi(d|i,r)$ and using the same arguments as in A.2, we obtain

$$\chi(d|i,r) = Pr\{\mathcal{S} = t - (r+u+i+d)\} \tag{A.10}$$

where $\mathcal{S}$ and u have been defined before. The probability $Pr\{\mathcal{S} = s\}$ is computed by (A.5) and

$$Pr\{\mathcal{S} < s\} = \frac{\Omega(t-r,s-1) + \sum_{j=1}^{s-2} \Lambda(j,\gamma-i-\zeta,0)\Omega(t-r-j,\zeta)}{\Omega(t-r,\gamma-i+1)} \tag{A.11}$$

where $\zeta = s - j + 1$. After obtaining $\nu(i|r)$ and $\chi(d|i,r)$, we can compute $\Psi_{\mathcal{D}_1|\mathcal{R}}(d|r)$ and then $\Theta_{\mathcal{R},\mathcal{D}_1}(r,d)$.

A.4 The density function $\Psi_{\mathcal{D}_2}$

The random variable $\mathcal{D}_2$ was defined to be the number of arrivals which should be discarded to get a valid buffer in scheme 2. To compute its density function $\Psi_{\mathcal{D}_2}(d)$, we first compute the number of valid arrival patterns for a given number of arrivals and number of slots. The state (t, b) implies that there are $M - b$ spaces in the buffer available at the beginning of this frame. If $t \leq M - b$ then $\mathcal{D}_2 = \emptyset$ since all arrivals can get into valid buffer. In the following, we analyze the case of $t > M - b$. Since $b > \gamma$ in this case, exactly i packets have left the buffer by the ith slot, thus, for a valid arrival pattern, the total number of arrivals by the end of the ith slot cannot exceed $M - b + i$. This problem can be mapped into a mathematical form as in F3, i.e.

$$X_1 + X_2 + \cdots + X_\gamma = t$$

subject to

$$X_1 + X_2 + \cdots + X_l \leq (M - b) + l, \quad 1 \leq l \leq \gamma - 1$$

where X_l is the number of arrivals during the lth slot. We need to know the number of feasible solutions for the problem. Therefore, given t, b, γ and M, the number of valid arrival patterns without discard is simply $\Lambda(t, \gamma, M - b)$ if $(M - b) + \gamma \geq t$; for $(M - b) + \gamma < t$, there is no feasible pattern since at least $t - (M - b) - \gamma$ arrivals must be discarded.

Now let us consider the general case when a valid buffer is not obtained until enough discards are made. We find the probability of a particular $\mathcal{D}_2 = d$, where $(t - (M - b + \gamma))^+ \leq d \leq t - (M - b + 1)$. Obviously, if $(M - b) + \gamma < t - d$, there are no valid patterns (d is too small). For $(M - b) + \gamma \geq t - d$, there are feasible solutions. We first determine the number of ways to delete d arrivals such that the resulting pattern is valid N_d^+, which includes patterns that were already valid before the dth deletion. This can be computed by multiplying the number of resulting valid patterns by the number of ways that d (deleted) arrivals could be put back into those patterns. Since the number of resulting valid patterns is $\Lambda(t - d, \gamma, M - b)$, N_d^+ is given by

$$N_d^+ = \Lambda(t - d, \gamma, M - b)\ [\Omega(d, \gamma + t - d) \cdot d!]$$

Next, we find N_d, the number of ways that the dth deletion causes the pattern to become valid. This is given by:

$$N_d = N_d^+ - N_{d-1}^+ \cdot C_1^{t-d+1} \tag{A.12}$$

where C_1^{t-d+1} is just the number of ways to make dth deletion on each pattern which is already valid. $\Psi_{\mathcal{D}_2}(d)$ can then be calculated by

$$\Psi_{\mathcal{D}_2}(d) = \frac{N_d}{\Omega(t, \gamma) \cdot C_d^t \cdot d!} \tag{A.13}$$

where the denominator is the total number of ways to delete d arrivals from all possible arrival patterns.

A.5 The density function $\Theta_{\mathcal{R},\mathcal{D}_2}$

The definitions of $\mathcal{R}$ and $\mathcal{D}_2$ can be found in section 2. The technique used to find $\Theta_{\mathcal{R},\mathcal{D}_1}$ can be applied here to obtain $\Theta_{\mathcal{R},\mathcal{D}_2}(r,d)$, i.e. we split the density function into two parts:

$$\Theta_{\mathcal{R},\mathcal{D}_2}(r,d) = \Psi_{\mathcal{R}}(r)\Psi_{\mathcal{D}_2|\mathcal{R}}(d|r)$$

where $\Psi_{\mathcal{D}_2|\mathcal{R}}(d|r)$ is the probability that d arrivals are discarded conditioned on the event that the first r arrivals get service in this frame. $\Psi_{\mathcal{D}_2|\mathcal{R}}(d|r)$ can be expressed as in equation (A.8) except that now $\chi(d|i,r)$ is computed by

$$\chi(d|i,r) = \frac{N'_d}{\Omega(t-r,\gamma-i+1)\cdot C_d^{t-r}\cdot d!} \tag{A.14}$$

instead of from (A.10), (A.5) and (A.11), where N'_d has the same form as (A.12) except that it operates in an environment where $t-\gamma$ arrivals occur in $\gamma-i+1$ slots with $(b-i)^+$ initial available spaces. $\Psi_{\mathcal{R}}(r)$ has been discussed before and thus $\Theta_{\mathcal{R},\mathcal{D}_2}(r,d)$ is obtained.

Appendix B

In the voice protocol in chapter 5, when competing for a slot, a competitor randomly chooses a minislot to send a request signal. Assume there are m minislots in a slot for voice competition, then $m = \lfloor T/\Delta \rfloor - 2$, where T is the slot length and Δ is the duration of a minislot. We subtract 2 to account for the first two minislots (used for reservation and announcement). Let n be the number of competitors involved in this competition, then we need to find the probability of at least one cell having exactly one object when n objects are distributed over m cells, denoted by $\varphi(n, m)$. From Feller [17],

$$\varphi(n,m) = 1 - \frac{m!n!}{m^n} \sum_{j=0}^{min\{m,n\}} (-1)^j \frac{(m-j)^{n-j}}{j!(m-j)!(n-j)!}$$

The values of $\varphi(n, m)$ obtained by varying n and m are shown in Figure B.1. It is observed that $\varphi(n, m) \approx 1$ if m is large enough, i.e. there is always a success no matter how many competitors there are. If we assume n is always less than m, the worst case occurs at $n = 2$.

Let us further examine the service behavior of voice traffic in this system. Assume the system is in a state S at the beginning of a frame with b backlogged users and t' slots already reserved. These b backlogged users will compete during the frame (new arrivals have to wait for the next frame), Once a user successfully competes in some slot the number of competitors for the next slot will be decreased by one. To investigate the competition during this frame, we represent the status of the non-reserved slots as a bit pattern, where a 0 represents a slot without success and a 1 represents a slot with a success. We break the bit string into sub-patterns consisting of 0's followed by 1 and define $p(n, r)$ as the probability of a sub-pattern with $r - 1$ 0's when there are n competitors involved, (in other words, this is the probability that the first success with n competitors occurs after r slots' competitions), then

$$p(n,r) = (1 - q(n))^{r-1} q(n)$$

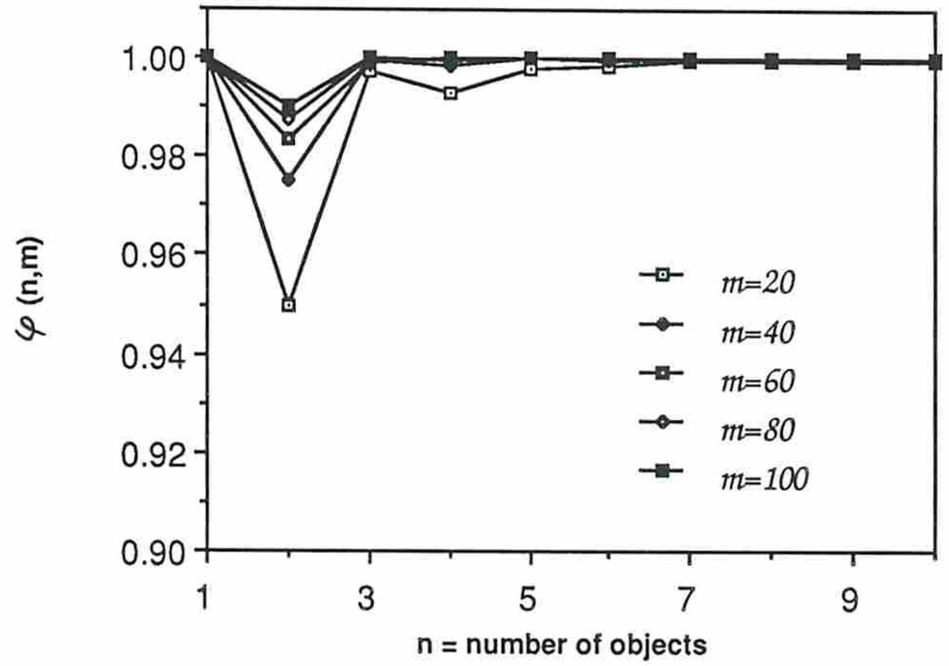


Figure B.1: The probability density function $\varphi(n,m)$

where $q(n)$ is the probability a success occurs in a slot given n competitors. Let us define the set of all $k+1$-tuples corresponding to patterns which have exactly k successes to be $\Gamma(v,k) = \{(\gamma_1, \gamma_2, \ldots, \gamma_k, \gamma_{k+1}) \; : \; \sum_{i=1}^{k+1} \gamma_i = v\}$ where γ_i is the number of slots in the subpattern associated with the ith success. (note: γ_{k+1} is the number of non-successful slots (0's) at the end of the frame). Then $\psi_{K_{bt}}(K_{bt} = k)$ (probability of k users switching from backlogged to transmitting is given by:

$$\psi_{K_{bt}}(K_{bt} = k) = \sum_{\gamma \in \Gamma(N-t',k)} \left(\prod_{i=0}^{k} p(b-i, \gamma_i) \right) (1 - q(b-k))^{\gamma_{k+1}}$$

Now, $q(n) = \varphi(n,m)$, where m is the number of minislots (per slot) that voice users can compete in, and for large m it was shown that $\psi(n,m) \approx 1$, thus

$$p(n,r) \cong \begin{cases} 1 & , r = 1 \\ 0 & , \text{otherwise} \end{cases}$$

which means that every slot having a nonzero number of competitors has a success. Then $\psi_{K_{bt}}(\cdot)$ is simply:

$$\psi_{K_{bt}}(K_{bt} = k) = \begin{cases} 1 & , k = min\{N - t', b\} \\ 0 & , \text{otherwise} \end{cases}$$

i.e. if the number of backlogged users is less than the number of available slots, all will succeed; whereas if the number of backlogged users is greater than the number of available slots, then all available slots will be occupied by successes. This is just the same behavior as an asynchronous TDM system where all the arrivals get an available slot until the system is saturated.

Bibliography

[1] N. Abramson. "The ALOHA system - another alternative for computer communications". In *Proc. Fall Joint Computer Conference*, 1970.

[2] J. P. Agrawal and V. M. Patel. "An advanced reservation multiple access (ARMA) protocol for integrated local networks". In *Proc. IEEE GLOBECOM 1986*, 1986, pp. 1-5.

[3] S. R. Amstutz. "Burst switching - an introduction". *IEEE Communication Magzine*, vol. 21, no. 8, pp. 36-42, Nov. 1983.

[4] E. Arthur and B. W. Stuck. "A theoretical traffic performance analysis of an integrated voice-data virtual circuit packet switch". *IEEE Trans. Commun*, vol. COM-27, pp. 1104-1111, July 1979.

[5] P. T. Brady. "A statistical analysis of on-off patterns in 16 conversations". *Bell Syst. Tech. J.*, vol. 47, pp. 73-91, Jan. 1968.

[6] P. T. Brady. "A model for generating on-off speech patterns in two-way conversations". *Bell Syst. Tech. J.*, vol. 48, pp. 2445-2472, Sept. 1969.

[7] J. I. Capetanakis. "Generalized TDMA : The multi-accessing tree protocol". *IEEE Trans. Commun.*, vol. COM-27, Oct. 1979, pp. 1476-1484.

[8] I. Chlamtac and M. Eisinger. "Performance of integrated services (voice/data) CSMA/CD networks". *ACM*, pp. 87-93, 1985.

[9] I. Chlamtac, W. Franta, and K. D. Levin. "BRAM : broadcast recognizing access method". *IEEE Trans. Commun.*, vol. COM-27, August, 1979, pp. 1497-1517.

[10] G. J. Coviello. "Comparative discussion of circuit- vs. packet-switched voice". *IEEE Trans. Commun*, vol. COM-27, pp. 1153-1160, Aug. 1979.

[11] J. N. Daigle and J. D. Langford. "Models for analysis of packet voice communication systems". *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 847-855, Sept. 1986.

[12] J. DeTreville. "A simulation-based comparison of voice transmission on CSMA/CD networks and on token buses". *Bell Syst. Tech. J.*, vol. 63, no. 1, pp. 33-35, Jan. 1984.

[13] A. Leon-Garcia et al. "Performance evaulation methods for an integrated voice/data link". *IEEE Trans. Commun*, vol. COM-30, pp. 1848-1858, Aug. 1982.

[14] C. J. Weinstein et al. "Data traffic performance of an integrated circuit- and packet-switched multiplex structure". *IEEE Trans. Commun*, vol. COM-28, pp. 873-878, Oct. 1980.

[15] H. B. Kekre et al. "Buffer behavior for mixed arrivals and single server with random interruptions". *IEEE Trans. Commun*, vol. COM-28, pp. 59-64, Jan. 1980.

[16] K. Sriram et al. "Discrete-time analysis of integrated voice-data multiplexer with and without speech activity detectors". *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 1124-1132, Dec. 1983.

[17] W. Feller. *Introduction to Probability Theory and Its Application*, volume I. Wiely, New York, 1957.

[18] M. Fine and F. A. Tobagi. "Packet voice on a local area network with round robin service". *IEEE Trans. Commun.*, vol. COM-34, no. 9, pp. 906-915, Sept. 1986.

[19] M. J. Fisher and T. C. Harris. "A model for evaluating the performance of an integrated circuit- and packet-switched multiplex structure". *IEEE Trans. Commun*, vol. COM-24, pp. 195-202, Feb. 1976.

[20] E. Friedman and C. Ziegler. "Real-time voice communications over a token-passing ring local area network". In *ACM SIGCOMM '86 Symposium*, volume pp. 52-57, 1986.

[21] I. Gitman and *et al.* "Analysis and design of hybrid switching networks". *IEEE Trans. Commun.*, vol. COM-29, No. 9, pp. 1290-1300, 1981.

[22] R. K. Goel and A. K. Elhakeen. "A hybrid FARA/CSMA-CD protocol for voice/data integration". *Computer Networks ISDN*, vol. 9, no. 3, pp. 223-240, 1985.

[23] J. G. Gruber. "A comparison of measured and calculated speech temporal parameters relevant to speech activity detection". *IEEE Trans. Commun*, vol. COM-30, no. 4, pp. 728-738, April, 1982.

[24] J. G. Gruber. "Delay related issues in integrated voice and data networks". *IEEE Trans. Commun*, vol. COM-29, no. 6, pp. 786-800, June 1981.

[25] J. F. Haughney. "Application of burst-switching technology to the defense communication system". *IEEE Communication Magzine*, vol. 22, no. 10, pp. 15-21, Oct. 1984.

[26] H. Heggestard. "A overview of packet-switching communications". *IEEE Communication Magzine*, vol. 22, pp. 24-31, Nov. 1984.

[27] R. A. Howard. *"Dynamic Probabilistic Systems, Vol. 1: Markov Models"*. Wiley, New York, 1971.

[28] N. S. Jayant and S. W. Christensen. "Effect of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure". *IEEE Trans. Commun.*, vol. COM-29, Feb. 1981.

[29] Y. C. Jenq. "Approximation for packetized voice traffic in statistical multiplexer". In *Proc. IEEE INFOCOM*, San Francisco, Nov. 1984, pp. 256-259.

[30] A. Joeland Jr. "Circuit-switching fundamentals". In J. McDonald, editor, *Fundamentals of Digital Switching.* Plenum, New York, 1983.

[31] B. G. Kim. "Characterization of arrival statistics of multiplexed voice packets". *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 1133-1139, Dec. 1983.

[32] B. G. Kim. "Two adaptive token ring strategies for real-time traffic". In *Proc. IEEE Computer Networking Symposium*, Dec. 1983, pp. 119-121.

[33] L. Kleinrock. *"Queueing Systems"*. John Wiley & Sons, New York, 1975.

[34] L. Kleinrock and M. Scholl. "packet switching in radio channels: New conflict-free multiple access schemes for a small number of data users". In *ICC Conference*, June, 1977.

[35] L. Kleinrock and Y. Yemini. "An optimal adaptive scheme for multiple access broadcast communication". In *Proc. ICC 1978*, 1978, pp. 326-332.

[36] D. E. Knuth. *"The Art of Computer Programming"*. Addison-Wesley, London, 1973.

[37] A. Leon-Garcia and O. S. Aboul-Magd. "Performance analysis of integrated voice and data tandem link network". In *Proc. IEEE GLOBECOM '86*, 1986, pp. 944-948.

[38] S. Li and J. C. Majithia. "Performance analysis of a DTDMA local area network for voice and data". *Computer Networks*, vol. 8, no. 2, pp. 81-91, 1984.

[39] S. Q. Li. "Performance of voice/data integration on a tdm system". *IEEE Trans. Commun*, vol. COM-33, no. 12, pp. 1265-1273, Dec. 1985.

[40] S. Q. Li. "A new measurement for voice transmission in burst and packet switching". In *Proc. IEEE INFOCOM*, San Francisco, March 1987, pp. 782-791.

[41] Y. Lim. "Preformance analysis of an integrated voice and data TDM link with digital speech interpolation". In *Proc. IEEE GLOBECOM 1986*, 1986, pp. 960-964.

[42] M. T. Liu. "Distributed loop computer networks". In M. C. Yovits, editor, *Advances in Computers*. Academic Press, New York, 1978.

[43] B. N. Ma and J. W. Mark. "Performance analysis of burst switching for integrated voice/data services". *IEEE Trans. Commun.*, vol. COM-36, pp. 282-297, March 1988.

[44] B. N. W. Ma and J. W. Mark. "End-to-end cutoff fraction of voice transmission in tandem link networks". In *Proc. IEEE ICC '88*, 1988, pp. 290-294.

[45] N. F. Maxemchuk. "A variation on CSMA/CD that yields movable TDM slots in integrated voice/data local networks". *Bell Syst. Tech. J.*, vol. 61, pp. 1527-1550, Sept. 1982.

[46] J. S. Meditch and Y. Zhao. "Framed TDMA/CSMA for integrated voice/data local networks". In *Proc. IEEE INFOCOM*, 1985, pp. 10-17.

[47] R. M. Needham and A. J. Herbert. *"The Cambridge Distributed Computing System"*. Addison-Wesley, London, 1982.

[48] M. F. Neuts. *"Matrix-geometric solutions in stochastic models : an algorithm approach"*. The Johns Hopkins University Press, Baltimore, MD, 1981.

[49] G. J. Nutt and D. L. Bayer. "Performance of CSMA/CD networks under combined voice and data loads". *IEEE Trans. Commun.*, vol. COM-30, pp. 1-11, Jan. 1982.

[50] P. O'Reilly. "Performance analysis of data in burst switching". *IEEE Trans. Commun.*, vol. COM-34, pp. 1259-1263, Dec. 1986.

[51] P. O'Reilly and S. Ghani. "Data performance in burst switching when the voice silence period have a hyperexponential distribution". *IEEE Trans. Commun.*, vol. COM-35, No. 10, pp. 1109-1112, Oct. 1987.

[52] B. K. Penney and A. A. Baghdadi. "Survey of computer communication loop networks : Part 1 and Part 2". *Computer Communications*, vol. 2, pp. 165-180, 224-241, 1979.

[53] L. G. Roberts. "ALOHA packet system with and without slots and capture". In *Computer Communications Review*, April, 1975.

[54] T. Saydam and A. S. Sethi. "Performance evaluation of voice-data token ring LANs with random priorities". In *Proc. IEEE INFOCOM 1985*, 1985, pp. 326-332.

[55] W. Stallings. *"Data and Computer Communications"*. MacMillan, 2nd Edition, New York, 1988.

[56] T. E. Stern. "A queueing analysis of packet voice". In *Proc. IEEE GLOBECOM '83*, San Diego, Dec. 1983, pp. 71-76.

[57] A. Tanenbaum. *"Computer Networks"*. Prentice-Hall, New York, 1981.

[58] F. A. Tobagi. *"Random Access technique for data transmission over packet switched radio networks"*. PhD thesis, Computer Sci.Dept., Univ. of California, Los Angeles, Dec. 1974.

[59] F. A. Tobagi and V. B. Hunt. "performance analysis of carrier sense multiple access with collision detection". *Computer Networks*, vol. 4, Oct.-Nov., 1980, pp. 245-259.

[60] C. J. Weinstein. "Fractional speech loss and talker activity model for TASI and for packet-switched speech". *IEEE Trans. Commun.*, vol. COM-26, no. 8, pp. 1253-1257, Aug. 1978.

[61] G. F. Williams and A. Leon-Garcia. "Performance analysis of integrated voice and data hybrid-switched links". *IEEE Trans. Commun*, vol. COM-32, pp. 695-706, June 1984.

[62] J. W. Wong and P. M. Gopal. "Analysis of a token ring protocol for voice transmission". In *Proc. IEEE Computer Networking Symposium*, Dec. 1983, pp. 113-117.

[63] Y. Yemini. *"On Channel Sharing in Discrete-time Packet Switched, Multiaccess Broadcast Communication"*. PhD thesis, Computer Sci.Dept., Univ. of California, Los Angeles, Dec. 1980.