Performance Analysis of a Voice Multiplexer with Bit Dropping [*]

Marta Barría [†] *Reinaldo Vallejos* [‡] *Luiz Fernando G. Soares* [§]

Abstract

This paper presents the performance evaluation of a multiplexer with a bit dropping algorithm for voice signal transmission. A new analytical approach is proposed to evaluate different performance measures. One of these measures is the distribution function of the packet residence time which is very important to determine the control parameters of the multiplexer. For some other measures, the analysis is done for congestion and non-congestion periods of the multiplexer. It is very important to separate the analysis in this way, since the quality of the transmitted voice is degraded only during congestion periods. To characterize the multiplexer behavior during congestion, we evaluate the distribution function of the congestion period length, the number of cell losses during congestion, and the length of a non-congestion period. Additionally, the mean values of the referred measures are evaluated.

Keywords: Performance Evaluation, Voice Multiplexer, Congestion Control, Bit Dropping Algorithm.

1 Introduction

Nowadays a great effort is being made to develop computer networks that are able to transmit audio, video and computer data information, simultaneously. As part of this effort, there is an increasing interest to develop performance models for these integrated networks, since, different from the data networks, there are still many phenomena not well understood in them. Particularly, this is the case of voice transmission [6], which is the subject of this paper.

Voice signals are made up of information talkspurts interlaced by silence periods, as shown in Figure 1. During talkspurts, voice is digitized, in the source, at a constant rate. On the other hand, during silence periods voice information may be not transmitted, in order to save bandwidth. Of course, the silence periods must be restored at the receiver.

To transmit voice through the network, a packet, consisting of a header and a number of voice samples, is assembled at the source. As a consequence of the non-transmission of silence periods and because talkspurts duration is variable, the voice packets generation is an asynchronous process.

In order to make an efficient use of the channel bandwidth, high speed networks, like IACS [9] and B-ISDN [10], statistically multiplex several voice sources, as shown in Figure 2. As each source generates packets asynchronously, the packet arrival to the multiplexer is also an asynchronous process. The multiplexer manages the traffic by means of a store and forward

---

[*] This paper was partially financed by a grant from HyperProp-PROTEM project of CNPq, Brazil

[†] Computer Science Department, Telemedia Lab., PUC-RJ, Brasil, e-mail: marta@inf.puc-rio.br

[‡] Electronic Department, Technical University Federico Santa María, Chile, e-mail: reinaldo@elo.utfsm.cl

[§] Computer Science Department, Telemedia Lab., PUC-RJ, Brasil, e-mail: lfgs@inf.puc-rio.br

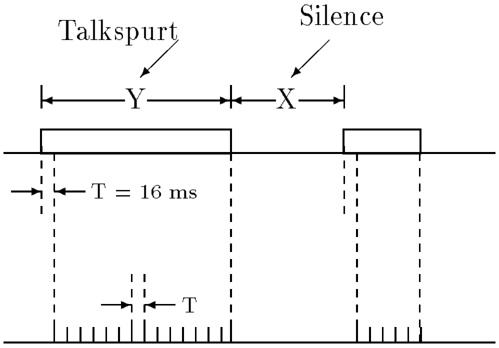

Figure 1: Voice Signal

process and, since the packet arrival is a random process and the buffer is finite, eventually congestion situations may occur.

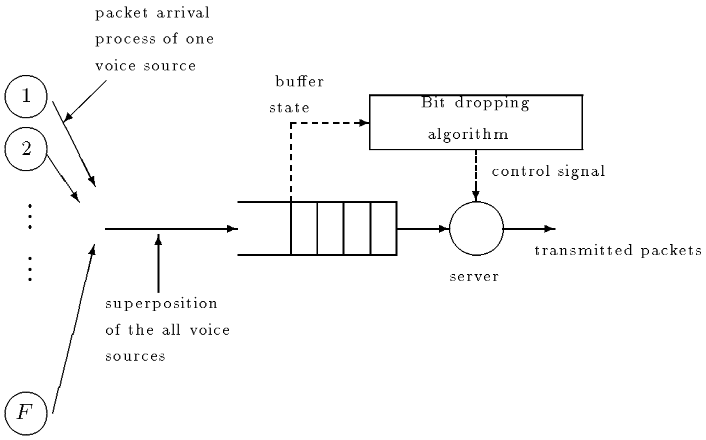

Figure 2: Voice Multiplexer

An important requirement in the transmission of voice packets is the transfer delay. This delay must be small (around a few hundred milliseconds), to maintain the interactivity of the communication. For this reason, it is not possible to increase the multiplexer buffer beyond a certain limit, since the time spent by the packet in the multiplexer queue could exceed the maximum delay allowed. As the arrival process is random and the buffer is limited, the superposition of talkspurts of several sources may deteriorate the quality of service offered by the network, increasing the transfer delays, or even worse, causing losses of packets.

Several proposals have been suggested in the literature to control congestion situations. One of them, used in the IACS network from the AT&T [9], discards the least significant bits of the voice samples when the multiplexer queue length (that is, the number of packet to transmit) exceeds a certain pre-defined limiar. This congestion control scheme, named *bit dropping*, has as goal "to speed up" the transmission of voice packets (through the bit dropping) when the multiplexer approaches congestion. The dropping of some bits is feasible, due to redundant information present in voice signals [1]. In the algorithm, losses occur when a voice packet is partially transmitted (due the drop mechanism) or when a complete packet is discard (if it arrives when the multiplexer buffer is completely full).

The intuitive idea behind the bit dropping mechanism is that it is better to lose the least significant bits of some voice samples than losing the entire samples. Since, with bit dropping, the possibility of the memory becomes full is reduced, the loss of entire packets is also reduced. The success or failure of this congestion control policy depends on the right choice of control parameters. In order to determine the best parameters this paper presents an analysis of its behavior under different conditions.

Although it is well known that, in general, the superposition of voice packets from different sources is not a Poisson process, it was demonstrated [12] that when a bit dropping mechanism is used, the Poisson process is a good model. Based in this assumption, Sriram [11] studied the voice multiplexer used in the IACS network of AT&T. The measures were evaluated only for the voice packets departure instants. In order to have a deep knowledge of the voice multiplexer behavior, this paper presents the evaluation of the same measures, but for all instants of time. Also we present the evaluation of new measures, as for example, the distribution function of the residence time of packets in the multiplexer. This measure is very important to determine the parameters of the bit dropping mechanism.

Defining the congestion period as the time interval elapsed during the sequential dropping of cells, and the non-congestion period as the time interval during which there is no cell losses, it is easy to see that, during non-congestion periods, the real time requirements are accomplished. On the other hand, during congestion periods the behavior of the multiplexer is degraded. Thus, other evaluated measures presented in this paper will allow the characterization of the multiplexer behavior during congestion periods. Among these measures we find the distribution function of the congestion period length, the distribution of the number of bits dropped by the algorithm during congestion periods, and the distribution of the non-congestion period length. The mean values of all these measures are also presented. These are very important results since, as a time-critical traffic, the voice distortion incurred is strongly dependent on the multiplexer behavior during congestion periods.

Still another result presented in the paper is a new analytical approach to evaluate the mean value and distribution functions of several performance measures during congestion periods. The analysis will greatly help the understanding of the multiplexer behavior, and so, its design.

The remaining of this paper is organized as follows. Section 2 describes the multiplexer operation and reviews the bit dropping algorithm. Section 3 presents the general model used to evaluate different performance measures and its mathematical solution. Section 4 shows some numerical examples. Finally, section 5 presents the conclusions.

2 Voice Packet Assembly and the Bit Dropping Mechanism

In this section we briefly describe how voice packets are assembled in the sources and how the bit dropping algorithm is used in IACS network of the AT&T. Reference [11] presents a detailed description of the algorithms.

As shown in Figures 1 and 3, in the voice packetization process, each source is sampled (only during talkspurts) at a constant rate of 8 KHz and coded using an embedded ADPCM scheme (using 4 bits per sample). Packets are made up of 128 samples gathered in a 16 ms interval. The

128 samples are arranged in one packet that have a header and 4 blocks, each of them having one bit from each of the 128 samples. The least significant bits from the 128 samples are placed sequentially in block #1 and the most significant ones in block #4. Blocks #3 and #2 contain the second and third most significant bits of the 128 samples, respectively. The packet header has 10 bytes and stores control information. The voice packets are thus made up of 74 bytes (see Figure 3): 10 bytes for the header and four blocks of 16 bytes as payload.

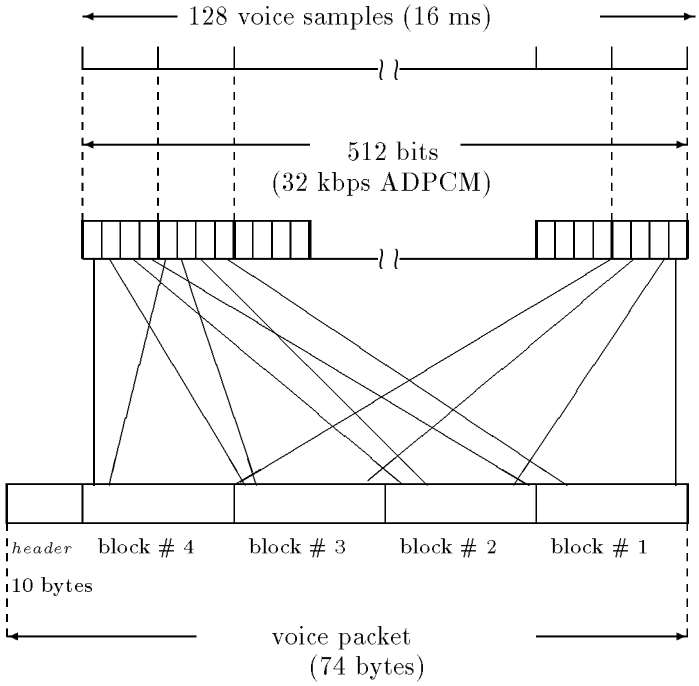

Figure 3: Voice packet structure

The bit dropping scheme is shown in Figure 4, for the special case of a T_1 channel. Let K be the maximum number of packets that can be stored in the multiplexer buffer. If, at the beginning of the transmission of the head of line packet, there are at most L_1 packets in the multiplexer memory, then the complete packet is transmitted. If the memory contains more than L_1 and less than or equal to L_2 packets, the packet transmission is "speed up" by transmitting only the header and the three most significant blocks. Similarly, if the occupation of the buffer is greater than L_2 and the buffer is not completely full, the transmission packet rate is speed up, by dropping the two less significant blocks. Finally, packets arriving when the buffer is full are totally discarded.

3 Performance Measures Evaluation

3.1 Multiplexer model

The multiplexer model used in this paper is the same used by Sriram [11]. With the purpose of completeness and as the basis for new analysis presented in this paper, we reproduce here the

Buffer State e_j	Congestion Control Action	Transmitted Packet Size (bytes)	Transmission Time in a T_1 channel (ms)
$0 \le j \le L_1$	no dropping	74	D_4=0.385
$L_1 < j \le L_2$	drops block # 1	58	D_3=0.302
$L_2 < j \le K-1$	drops blocks # 1 and # 2	42	D_2=0.219

Figure 4: Bit dropping algorithm description

model and some basic analysis presented in [11].

Let λ_f be the mean arrival packet rate from a specific source f, and let F be the number of voice sources (see Figure 2). Assuming that all the sources have the same characteristics, the packet arrival rate to the multiplexer is $\lambda = F\lambda_f$. The packet arrival process to the multiplexer is modeled as a Poisson process with parameter λ. Let e_j be the multiplexer state in any instant, where j is the number of packets stored in the buffer, including the one being transmitted. We define the instants where a packet transmission ends as being the embedded points in time. Let δ_i be a cycle of type i, that corresponds to the time interval ranging from an embedded point where the multiplexer state is e_i to the next embedded point; and let s_i be the time needed to transmit a packet in a δ_i cycle. Then, as shown in Figure 4, the packet service time is deterministic, but load dependent, due to the bit dropping algorithm:

$$s_i = \begin{cases} D_4, & i = 0, 1, 2, \ldots, L_1 \\ D_3, & i = L_1 + 1, \ldots, L_2 \\ D_2, & i = L_2 + 1, \ldots, K - 1 \end{cases} \tag{1}$$

For a T_1 channel, in the example of Figure 4, we have: $D_4 = 0.385$ ms, $D_3 = \frac{58}{74}D_4$ and $D_2 = \frac{42}{74}D_4$.

Let a_k, b_k and c_k be the probabilities of k voice packets arrivals at the multiplexer in an interval of size D_4, D_3 and D_2, respectively. Then, because the arrival process has a Poisson distribution, it follows that:

$$\begin{aligned} a_k &= \frac{e^{-\lambda D_4}(\lambda D_4)^k}{k!}, & k \ge 0 \\ b_k &= \frac{e^{-\lambda D_3}(\lambda D_3)^k}{k!}, & k \ge 0 \\ c_k &= \frac{e^{-\lambda D_2}(\lambda D_2)^k}{k!}, & k \ge 0 \end{aligned} \tag{2}$$

Let χ_n be the discrete time Markov chain, defined at the embedded points, whose states correspond to the number of stored packets in the multiplexer. Let $\mathbf{P}$ be the transition probabilities matrix among χ_n states. It is easy to prove that $\mathbf{P}$ is given by the matrix of the Figure 5.

As $\mathbf{P}$ is a stochastic matrix, the term $(1 - \sum)$ at the right of each $\mathbf{P}$ row corresponds to 1 minus the sum of all the other terms of the row.

$$
\begin{array}{c|ccccccccccccccc|}
 & 0 & 1 & 2 & \ldots & L_1{-}1 & L_1 & L_1{+}1 & \ldots & L_2{-}1 & L_2 & L_2{+}1 & \ldots & K{-}2 & K{-}1 \\
0 & a_0 & a_1 & a_2 & & . & . & . & & . & . & . & & & 1-\sum \\
1 & a_0 & a_1 & a_2 & & . & . & . & & . & . & . & & & 1-\sum \\
2 & 0 & a_0 & a_1 & & . & . & . & & . & . & . & & & 1-\sum \\
. & & & & & & & & & & & & & & \\
. & & & & & & & & & & & & & & \\
. & & & & & & & & & & & & & & \\
L_1{-}1 & & & & a_0 & a_1 & a_2 & a_3 & & . & . & . & & & . \\
L_1 & & & & & a_0 & a_1 & a_2 & & . & . & . & & & . \\
L_1{+}1 & & & & & & b_0 & b_1 & & . & . & . & & & . \\
. & & & & & & & & & & & & & & \\
. & & & & & & & & & & & & & & \\
. & & & & & & & & & & & & & & \\
L_2{-}1 & & & & & & & & b_0 & b_1 & b_2 & b_3 & & & . \\
L_2 & & & & & & & & & b_0 & b_1 & b_2 & & & . \\
L_2{+}1 & & & & & & & & & & c_0 & c_1 & & & . \\
. & & & & & & & & & & & & & & \\
. & & & & & & & & & & & & & & \\
. & & & & & & & & & & & & & & \\
K{-}2 & & & & & & & & & & & & c_0 & c_1 & 1-\sum \\
K{-}1 & & & & & & & & & & & & & c_0 & 1-\sum
\end{array}
$$

Figure 5: Matrix $\mathbf{P}$

3.2 Buffer State at the Embedded Points

Let $\boldsymbol{\pi} = \langle \pi_0, \pi_1, \pi_2, \ldots, \pi_{K-1} \rangle$ be the left eigenvector of $\mathbf{P}$. The vector $\boldsymbol{\pi}$ is commonly evaluated by resolving the equation $\pi = \pi \mathbf{P}$. However, as the transitions of $\mathbf{P}$ from a specific state e_i, $1 \leq i < K$, can reach only states e_j, with $j \geq i-1$, $\boldsymbol{\pi}$ can be evaluated more efficiently, as follows [7].

Let X_i be a non-normalized state of π_i, $i = 0, 1, 2, 3, \ldots K-1$; that is, $\sum_{i=0}^{K-1} X_i \neq 1$, and $\frac{X_i}{X_j} = \frac{\pi_i}{\pi_j}$. We arbitrarily assume that:

$$X_0 = 1 \tag{3}$$

Now, applying the global balance equation to the state e_0, but using X_0 and X_1 instead of π_0 and π_1, respectively, we obtain:

$$X_1 = X_0 \frac{(1-a_0)}{a_0} \tag{4}$$

Analogously, we can obtain recurrent equations to evaluate X_{i+1}, $2 \leq i+1 \leq K-1$, by applying global balance equations to the states e_i, $1 \leq i \leq K-2$. Then, we have:

$$\underset{2\leq i+1\leq L_1}{X_{i+1}} = \frac{1}{a_0}\left\{X_i(1-a_1) - X_0 a_i - \sum_{k=1}^{i-1} X_k a_{i-k+1}\right\} \tag{5}$$

$$X_{L_1+1} = \frac{1}{b_0}\left\{X_{L_1}(1-a_1) - X_0 a_{L_1} - \sum_{k=1}^{L_1-1} X_k a_{L_1-k+1}\right\} \tag{6}$$

$$\underset{L_1+2\leq i+1\leq L_2}{X_{i+1}} = \frac{1}{b_0}\left\{X_i(1-b_1) - X_0 a_i - \sum_{k=1}^{L_1} X_k a_{i-k+1} - \sum_{k=L_1+1}^{i-1} X_k b_{i-k+1}\right\} \tag{7}$$

$$X_{L_2+1} = \frac{1}{c_0}\left\{X_{L_2}(1-b_1) - X_0 a_{L_2} - \sum_{k=1}^{L_1} X_k a_{L_2-k+1} - \sum_{k=L_1+1}^{L_2-1} X_k b_{L_2-k+1}\right\} \tag{8}$$

$$\underset{L_2+2\leq i+1\leq K-1}{X_{i+1}} = \frac{1}{c_0}\left\{X_i(1-c_1) - X_0 a_i - \sum_{k=1}^{L_1} X_k a_{i-k+1} - \sum_{k=L_1+1}^{L_2} X_k b_{i-k+1} - \sum_{k=L_2+1}^{i-1} X_k c_{i-k+1}\right\} \tag{9}$$

If $S = \sum_{i=0}^{K-1} X_i$, it can be easily proved that:

$$\pi_i = \frac{X_i}{S}; \quad 0 \leq i \leq K-1 \tag{10}$$

Thus, finally, the set of equations 3 to 10 gives π, more efficiently than using equation $\pi = \pi\mathbf{P}$.

3.3 Mean Number of Packets in the Multiplexer

Let N be the mean number of voice packets in the multiplexer, N_i be the mean number of packets in the multiplexer during a δ_i cycle, and ω_i be the steady state probability of being in a δ_i cycle. Then, we have:

$$N = \sum_{i=0}^{K-1} \omega_i N_i \tag{11}$$

By renewal theory we know that [8]:

$$\omega_i = \frac{\pi_i B_i}{\sum_{j=0}^{K-1} \pi_j B_j} \tag{12}$$

where B_i is the mean duration of δ_i cycle.

To evaluate B_i we must distinguish two cases. The first one corresponds to $i = 0$, where there are no packets in the multiplexer at the beginning of the cycle. In this case, B_0 has a duration equal to the mean time a packet takes to reach the multiplexer plus the service time of the first arriving packet, that is, $B_0 = 1/\lambda + D_4$. The second case corresponds to e_i, $i > 0$, in which B_i depends on the packet service time, as shown in equation 1. Summarizing we have:

$$B_i = \begin{cases} \frac{1}{\lambda} + D_4; & i = 0 \\ D_4; & 1 \leq i \leq L_1 \\ D_3; & L_1 < i \leq L_2 \\ D_2; & L_2 < i < K \end{cases} \tag{13}$$

The evaluation of N_i requires a more laborious development. Let us attribute, a reward speed i to the multiplexer, when it is in state e_i. This means that, when the multiplexer has i packets stored in its memory, including the one being served, it gains a reward equal to i, by each time unit under such state. In other words, each packet stored in the multiplexer's buffer contributes with exactly one reward unit, each time unit. This interpretation will allow us to calculate the total reward, gained by the multiplexer, adding contributions made by each of its packet.

Using the previous reward rate, we have that N_i corresponds to the mean reward gained during a δ_i cycle, divided by the mean size of such cycle. More precisely, if we define R_i as the mean reward accumulated by the multiplexer in a δ_i cycle, we have that:

$$N_i = \frac{R_i}{B_i} \tag{14}$$

To compute R_i, we condition on the number of arrivals occurring during δ_i cycle. Let us then define $R_{i|j}$ as the mean reward accumulated in the δ_i cycle, when j packets arrive during this interval. Then, we have:

$$R_i = \sum_{j=0}^{K-i-1} Z_i(j) R_{i|j} + \sum_{j=K-i}^{\infty} Z_i(j) R_{i|j} \tag{15}$$

where $Z_i(j)$ corresponds to the probability of j packets arrivals during the δ_i cycle. As the arrival process has Poisson distribution with parameter λ, the equation (15) can be rewritten as follows:

$$R_i = \sum_{j=0}^{K-i-1} e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} R_{i|j} + \sum_{j=K-i}^{\infty} e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} R_{i|j} \tag{16}$$

where s_i is given by equation (1).

The first sum of equation (16) corresponds to the cases where the j arrival packets are not enough to fill up the multiplexer, while the second sum corresponds to all cases where the j arrivals fill up the multiplexer, during the δ_i cycle. This will allows to easily calculate $R_{i|j}$ as follows.

The value of $R_{i|j}$ in the first sum of (16) corresponds to the area under the curve of Figure 6. As shown in that figure, the total accumulated reward in the δ_i cycle has two components: one of them is due to the number of packets in the multiplexer at the beginning of the cycle, and the other one depends on the number of packets arriving at the multiplexer during s_i. For $i > 0$, the total accumulated reward due to the first component is just is_i. To calculate the reward gained due to the second component we recall that the packets arrival instants have Poisson distribution. To simplify the mathematical analysis, we use the theorem: "the arrival instants of j events, occurring with Poisson distribution in a s_i interval, have the same joint distribution as the order statistics of j independent events, uniformly distributed in such interval" [3]. Then, we have that the reward gained by the j packets arriving during s_i corresponds to the sum of the mean residence time (in the δ_i cycle) of each of these j packets. Moreover, due to the uniform distribution and independence of packet arrivals, it is easy to see that the mean residence time of each packet, during the δ_i cycle, is equal to the half of the interval. This implies that the mean reward gained by the j packets arriving in s_i is simply $j\frac{s_i}{2}$. Finally, by adding the two components, we obtain:

$$R_{i|j} = is_i + j\frac{s_i}{2}; \qquad 0 < i,\ (i+j) < K \tag{17}$$

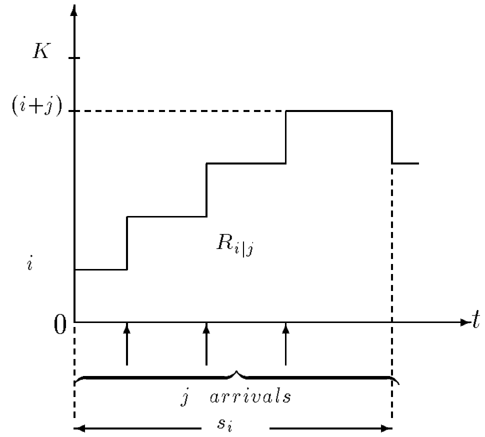

Figure 6: Accumulated reward in the δ_i cycle ($i > 0$) when the multiplexer's buffer is not full.

To calculate ($R_{i|j}$) in the second sum of 16, with $i > 0$, we use the fact that j arrivals occur during interval $[0, s_i]$, $i + j \geq K$, as shown in Figure 7. In this case, we first note that packets arriving at the multiplexer in the δ_i cycle, when the memory is full, are lost. Then, let the random variable τ be the instant that the memory fills up (considering the beginning of δ_i as time 0). Let $A'_{i|j}(\tau)$ be the total reward gained by the multiplexer in the interval $[0, \tau)$ of the δ_i cycle, and let $A''(\tau)_{i|j}$ be the total reward gained by the multiplexer in the interval $[\tau, s_i]$ of the same cycle (see Figure 7). Applying the total probability theorem, we have that the total reward accumulated by the system during the δ_i cycle is:

$$R_{i|j} = \int_{\tau=0}^{\infty} \left(A'_{i|j}(\tau) + A''_{i|j}(\tau)\right) f(\tau)d\tau, \qquad i > 0,\ \ i+j \geq K \tag{18}$$

where $f(\tau)$ corresponds to probability density function of τ.

To calculate $A'_{i|j}(\tau)$, we proceed as in the previous case, in which the buffer is not filled up (see Figures 6 and 7), with the difference that here we must replace s_i, from equation 17, by τ, once now, this is the time the buffer takes to fill up. Thus, noting that $K - i - 1$ packets arrive in $[0, \tau)$, we have:

$$A'_{i|j}(\tau) = \frac{K-1+i}{2}\tau; \qquad i > 0,\ \ i+j \geq K \tag{19}$$

To evaluate, $A''_{i|j}(\tau)$ we take in account that it simply corresponds to the total reward gained by the multiplexer due to its K resident packets during the $[\tau, s_i]$ interval. Then:

$$A''_{i|j}(\tau) = K\,(s_i - \tau)\,; \qquad i > 0,\;\; i + j \geq K \tag{20}$$

Using (19) and (20) in (18), we have:

$$\underset{i>0,\;\; i+j\geq K}{R_{i|j}} = \int_{\tau=0}^{s_i} \left\{ \frac{K-1+i}{2}\tau + K(s_i - \tau) \right\} \cdot f(\tau) d\tau \tag{21}$$

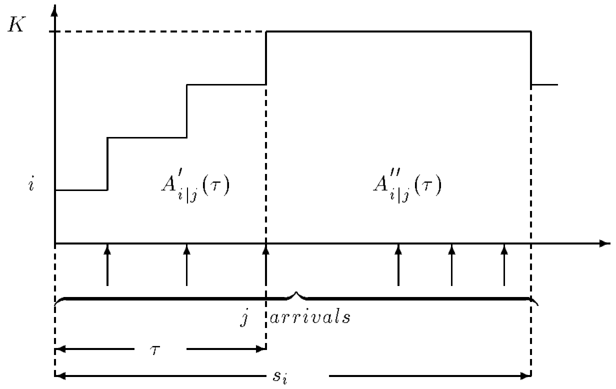

Figure 7: Accumulated reward in the δ_i cycle ($i > 0$), when the multiplexer's buffer is full

To calculate $f(\tau)$ we use the fact that τ corresponds to the $(K - i)$-th order statistics of j independent random variables uniformly distributed in the $[0, s_i]$ interval [3]. Thus:

$$f(\tau) = \binom{j}{K-i-1} \left[\frac{\tau}{s_i}\right]^{K-i-1} (j - K + i + 1)\,\frac{1}{s_i} \left[\frac{s_i - \tau}{s_i}\right]^{j-K+i} \tag{22}$$

Replacing (22) in (21):

$$\underset{i>0,\;\; i+j\geq K}{R_{i|j}} = s_i^{-j}\,(j - K + i + 1) \int_{\tau=0}^{s_i} \left\{ \frac{K-1+i}{2}\tau + K(s_i - \tau) \right\} \binom{j}{K-i-1} \tau^{K-i-1}\,(s_i - \tau)^{j-K+i}\, d\tau$$

Applying the binomial theorem for the expression $(s_i - \tau)^{j-K+i}$, exchanging the order of the integral with the sum, and simplifying, we have:

$$\begin{aligned} \underset{i>0,\;\; i+j\geq K}{R_{i|j}} &= s_i \binom{j}{K-i+1} (j - K + i + 1) \\ &\sum_{l=0}^{j-K+i} \binom{j-K+i}{l} (-1)^l \left\{ \frac{K}{l + (K - i)} - \frac{\frac{(K-i+1)}{2}}{l + (K - i + 1)} \right\} \end{aligned} \tag{23}$$

Now, as $\sum_{l=0}^{m} \frac{\binom{m}{l}(-1)^l}{(l+L)} = \left\{ L \binom{L+m}{L} \right\}^{-1}$, from (23), and simplifying, we have:

$$\underset{i>0,\ i+j\geq K}{R_{i|j}} = s_i \left(\frac{K-i}{j+1}\right) \cdot \left(\frac{K-1+i}{2}\right) + s_i \left(\frac{j-K+i+1}{j+1}\right) K \tag{24}$$

Equation (24) has an interesting interpretation. Observe that the mean value of the base of $A'_{i|j}(\tau)$ (see Figure 7) is given by $s_i \frac{(K-i)}{(j+1)}$, and that the mean value of the height of $A'_{i|j}(\tau)$ is given by $\frac{K-1+i}{2}$, (this can be rigorously demonstrated). Analogously, $s_i \left(\frac{j-K+i+1}{j+1}\right)$ corresponds to the mean size of the $A''_{i|j}(\tau)$ base, and the mean height of $A''_{i|j}(\tau)$ clearly is K.

Now, to obtain R_i, $i > 0$, we replace (17) and (24) in (16), from where we have:

$$\begin{aligned} \underset{i>0}{R_i} &= \sum_{j=0}^{K-i-1} \left[i s_i + j \frac{s_i}{2} \right] e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} \\ &+ \sum_{j=K-i}^{\infty} s_i \left[\left(\frac{K-i}{j+1}\right) \left(\frac{(K-1+i)}{2}\right) + K \left(\frac{j-K+i+1}{j+1}\right) \right] e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} \end{aligned} \tag{25}$$

To obtain R_0, we see that in the δ_0 cycle, the multiplexer only starts gaining reward when the first packet arrives. From then on, the system behavior is the same as that of the δ_1 cycle . Assuming $L_1 \geq 1$, then $R_0 = R_1$. Finally, replacing (13) and (14) in (11), we conclude that the mean number of packets in the multiplexer is:

$$\begin{aligned} N &= \omega_0 \frac{D_4}{D_4 + \frac{1}{\lambda}} \left\{ \sum_{j=0}^{K-2} \left[1 + \frac{j}{2} \right] e^{-\lambda D_4} \frac{[\lambda D_4]^j}{j!} \right. \\ &\quad \left. + \sum_{j=K-1}^{\infty} \left[\left(\frac{K-1}{j+1}\right) \frac{K}{2} + K \left(\frac{j-K+2}{j+1}\right) \right] e^{-\lambda D_4} \frac{[\lambda D_4]^j}{j!} \right\} \\ &+ \sum_{i=1}^{K-1} \omega_i \left\{ \sum_{j=0}^{K-i-1} \left[i + \frac{j}{2} \right] e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} \right. \\ &\quad \left. + \sum_{j=K-i}^{\infty} \left[\left(\frac{K-i}{j+1}\right) \left(\frac{K-1+i}{2}\right) + K \left(\frac{j-K+i+1}{j+1}\right) \right] e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} \right\} \end{aligned} \tag{26}$$

Where ω_i, $0 \leq i < K$, is given by equation (12).

3.4 Mean Residence Time of a Packet in the Multiplexer

As the multiplexer memory is finite, not all the arriving packets enter into the queue. The throughput of the system is given by $\mu(1 - p_0)$, where p_0 is the probability that the multiplexer is empty at an arbitrary time instant, and μ is the mean packet transmission rate, when the multiplexer is not idle.

To calculate p_0, we note that instants during which the system is empty are renewal points of the stochastic behavior of the system. Starting from a renewal point, the system remains idle for a time interval, defined as *idle period*. From the arrival of the first packet, the system remains busy for a time interval, defined as *busy period*, where it serves all the packets that arrive before it empties again. Let I be the mean value of an idle period, and B be the mean value of a busy period. From renewal theory we know that $p_0 = I/(I+B)$. Moreover, as the packet arrival is characterized by a Poisson process with rate λ, we have that $I = 1/\lambda$. On the other hand, we also know from renewal theory that $B = (1/\pi_0)(1/\mu)$, where $1/\pi_0$ corresponds to the mean number of packets transmitted during a busy period, and $1/\mu$ is the mean time for packet transmission. By definition, we have that $(1/\mu) = \sum_{i=0}^{K-1} \pi_i s_i$, that is:

$$\mu = \left[D_4 \sum_{i=0}^{L_1} \pi_i + D_3 \sum_{i=L_1+1}^{L_2} \pi_i + D_2 \sum_{i=L_2+1}^{K-1} \pi_i \right]^{-1} \tag{27}$$

From the last paragraph:

$$p_0 = \frac{\pi_0}{\pi_0 + \rho} \tag{28}$$

where $\rho = \lambda/\mu$

To calculate T, the packet mean residence time of the multiplexer, we use Little's law, by which:

$$T = \frac{N}{\mu(1 - p_0)} \tag{29}$$

Replacing (27) and (28) in (29), we conclude that T is given by:

$$T = \left(\frac{\pi_0 + \rho}{\rho} \right) N \left[D_4 \sum_{i=0}^{L_1} \pi_i + D_3 \sum_{i=L_1+1}^{L_2} \pi_i + D_2 \sum_{i=L_2+1}^{K-1} \pi_i \right] \tag{30}$$

3.5 Mean Residence Time of a Packet in the Multiplexer's Queue

Let W be the mean time that a packet spends in the multiplexer's queue. We have that:

$$W = T - \frac{1}{\mu} \tag{31}$$

and using (27) and (30) in (31), we get:

$$W = \left[D_4 \sum_{i=0}^{L_1} \pi_i + D_3 \sum_{i=L_1+1}^{L_2} \pi_i + D_2 \sum_{i=Q_2+1}^{K-1} \pi_i \right] \left(\frac{\pi_0 + \rho}{\rho} N - 1 \right) \tag{32}$$

3.6 Mean Number of Packets in the Queue

Let L_q be the mean number of packets in the queue. By applying Little's law, $L_q = \mu(1-p_0)W$. Then, using (26), (28) and (32), we have:

$$L_q = N - \frac{\rho}{\pi_0 + \rho} \tag{33}$$

3.7 Congestion Analysis

We have defined a congestion period as the time interval elapsed during sequential transmission of packets with bit dropping. In [4], San-Qi Li demonstrated that the evaluation of measures averaged over all instants of time is not enough to characterize the behavior of a voice multiplexer, since it alternates between congestion and non-congestion periods, and the system performance is only degraded during the first one. The evaluation of performance measures that characterize system behavior under the influence of the congestion intervals is then important and is the objective of this section.

To characterize a congestion period we will use a methodology extracted from the area of performability [5]. We will define a particular performability problem and then we will apply its solution to the evaluation of some performance measures that characterize the system behavior with respect to congestion.

To state the performability problem of interest, consider a homogeneous discrete time Markov chain in which each state transition is associated with a specific reward. In other words, each time the system carries out a certain transition, it receives a (fixed) reward associated with this transition.

As we will see, for the analysis of the multiplexer, it would be useful to assign rewards to states, instead of assigning rewards to transitions. However, a brief reasoning shows that the assignment of rewards to states is a special case of the assignment of rewards to transitions. To be more general, we will consider this last case.

To specify the performability problem in formal terms, consider a discrete time homogeneous Markov chain, $\mathcal{Y}_n = \{Y(n), n \geq 0\}$, that describes the behavior of the system at the embedded points. Let $E = \{e_j, j = 1, \ldots, |E|\}$ be the finite state space associated to the model, where $|E|$ corresponds to the total number of states of E. Let $\mathbf{Z}$ be the transition probabilities matrix between states of E, and z_{ij} the probability that the Markov chain transits, in one step, from state e_i to state e_j. Let $T \subset E \times E$ be the set of transitions between the states of the Markov chain. The transition from state $e_i \in E$ to state $e_f \in E$ is written t_{if}. Let each transition $t_{if} \in T$ be associated with a specific reward, defined as $\eta(i, f)$, which corresponds to the reward gained by the system each time it moves from state e_i to state e_f.

Let $\Theta = \{\theta_k, k = 1, \ldots, \theta_{|\Theta|}\}$ be the set of different rewards associated to the transitions of T. In general, $|\Theta| \leq |E|^2$, because different transitions may be associated to a same reward, and some transitions between states may be impossible to occur. Let $RTAD(n)$ be the total reward gained by the system in the first n transitions of the Markov chain, in the case that rewards gain occurs at the transitions, according to a function η previously defined. Due to the fact that during the interval of observation the Markov chain may move along different paths, $RTAD(n)$ is a random variable. Usually, although not necessarily, it is assumed that $RTAD(0) = 0$. Let $G(n, r)$ be the probability distribution function of $RTAD(n)$:

$$G(n,r) = P[RTAD(n) \leq r] \tag{34}$$

To evaluate $G(n,r)$, we will use the same procedure of [2], where a similar problem was solved, but for a continuos time Markov chain. Let us define $G_f(n,r)$ as the probability that $RTAD(n)$ be less or equal to r, and define the state of the Markov chain immediately after n transitions as e_f. Then, considering all the different possible cases, we have:

$$G(n,r) = \sum_{e_f \in E} G_f(n,r) \tag{35}$$

To evaluate $G_f(n,r)$ we proceed as follows. Let $RTAD_f(n)$ correspond to the random variable $RTAD(n)$ in the special case where the Markov chain is in state e_f after n transitions, and let $\mathcal{R}_f^n$ be the set of all different possible values of $RTAD_f(n)$. Let w be one of the values belonging to $\mathcal{R}_f^n$. Obviously, the value of w depends on the sample path followed by the Markov chain, however, the same value of w may correspond to different paths. Let $r_f^n(r)$ be the subset of $\mathcal{R}_f^n$ consisting of all the rewards $w \in \mathcal{R}_f^n$, such that $w \leq r$. Then, considering all the different cases, we have:

$$G_f(n,r) = \sum_{w \in r_f^n(r)} P\left[RTAD_f(n) = w\right] \tag{36}$$

The probability $P\left[RTAD_f(n) = w\right]$ can be evaluated recurrently on the value of n. For $n = 0$, assume that $\mathcal{R}_f^n = \{0\}$, then:

$$P\left[RTAD_f(0) = w\right] = \alpha_f(0) I\left[w = 0\right] \tag{37}$$

where $\alpha_f(0)$ is the probability that the system starts its operation in the state $e_f \in E$.

The set of probabilities $\alpha_f(0)$, $e_f \in E$, is a system input parameter or must be evaluated specifically for each particular problem. On the other hand, $I\left[event\right]$ is the indicator function of the *event*. In order to evaluate $P\left[RTAD_f(n) = w\right]$ for $n \geq 1$, we condition on the last transition of the Markov chain, from where we have:

$$P\left[RTAD_f(n) = w\right] = \sum_{e_i \in E} z_{if} P\left[RTAD_i(n-1) = w - \eta(i,f)\right] \tag{38}$$

Equations (35) to (38) represents an efficient solution to the evaluation of $G(n,r)$. Reference [2] presents a discussion of its computational complexity.

In what follows we will use the solution of $G(n,r)$ to evaluate different measures of a congestion period.

3.8 Probability Distribution Function of the Congestion Period Length

To evaluate the probability distribution function of the congestion period duration, we use the Markov chain with transition probabilities matrix $\mathbf{Z}_1$ of the Figure 3.8. The state e_f, $L_1 < f < K$, of $\mathbf{Z}_1$, corresponds to the number of packets in the multiplexer at an embedded point, during a congestion period. The transition probabilities c_i, and b_i, $i \geq 0$, were defined in equation (10). The state e_a is an absorbent state used to model the end of the congestion period. The system enters in the state e_a when it goes for the first time from state e_{L1+1} to state e_{L1} (with probability b_0, as can be seen in the matrix Z_1).

$$
\begin{array}{c|cccccccccc|}
 & a & L_1{+}1 & L_1{+}2 & \ldots & L_2 & L_2{+}1 & L_2{+}2 & \ldots & K{-}2 & K{-}1 \\
a & 1 & & & & & & & & & \\
L_1{+}1 & b_0 & b_1 & b_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1-\sum \\
L_1{+}2 & & b_0 & b_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1-\sum \\
\cdot & & & & & \cdot & \cdot & \cdot & \cdot & \cdot & \\
\cdot & & & & & & & & & & \cdot \\
\cdot & & & & & & & & & & \cdot \\
L_2 & & & & b_0 & b_1 & b_2 & b_3 & \cdot & \cdot & 1-\sum \\
L_2{+}1 & & & & & c_0 & c_1 & c_2 & \cdot & \cdot & 1-\sum \\
L_2{+}2 & & & & & & c_0 & c_1 & \cdot & \cdot & 1-\sum \\
\cdot & & & & & & & & & & \cdot \\
\cdot & & & & & & & & & & \cdot \\
\cdot & & & & & & & & & & \cdot \\
K{-}2 & & & & & & & & c_0 & c_1 & 1-\sum \\
K{-}1 & & & & & & & & & c_0 & 1-c_0
\end{array}
$$

Figure 8: Transition probabilities matrix $\mathbf{Z_1}$ of Markov chain that models the system behavior during a congestion period

Now, consider the following function of reward assignments to the transitions t_{if} of $\mathbf{Z_1}$:

$$\eta_1(i,f) = \begin{cases} D_3; & L_1 < i \leq L_2 \\ D_2; & L_2 < i < K \\ 0; & i = a \end{cases} \tag{39}$$

The assignment of reward of (39) depends only on which state the Markov chain is, before the transition. This illustrates, as previously mentioned, that the assignment of rewards to states is a special case of the assignment of rewards to transitions.

The key observation is that, with the reward assignment of (39), the reward accumulated by the system, after a certain number of transitions, is equivalent to the duration of the congestion period, since each time the system visits the state e_j, $L_1 < j < K$, it gains a reward equal to the time it spends in that state, and when it falls into the absorbent state e_a, the system stops of earning rewards.

Let $n_a = \lceil \frac{r}{D_2} \rceil$. Then, after n_a transitions, the reward accumulated by the system in any path that ends in some state e_f, $L_1 < f < K$, is greater than r. The reward accumulated by the system can be less than or equal to r after n_a transitions, only when in some transition before the n_a-th, the chain falls into the absorbent state. Let $P[CP \leq r]$ be the probability that the duration of the congestion period be less than or equal to r time units. Then, the previous observation implies that:

$$P[CP \leq r] = G_a(n_a, r) \tag{40}$$

where $G_a(n_a, r)$ is recurrently evaluated using equations 36 to 38.

To evaluate $P[CP \leq r]$ it remains to know the probability distribution of in which state the system starts the congestion period, that is, $\alpha_f(0)$, $L_1 < f < K$. This is done noting that all congestion periods start soon after a non-congestion state. Then, conditioning on this state, applying the total probability theorem, and using matrix $\mathbf{P}$ of Section 3.1, we have:

$$\alpha_f(0) = \frac{\pi_0 a_f + \sum_{i=1}^{L_1} \pi_i a_{f-i+1}}{\pi_0 \sum_{k=L_1+1}^{K-1} a_k + \sum_{i=1}^{L_1} \pi_i \sum_{k=L_1+1}^{K-1} a_{k-i+1}}; \quad L_1 < f < K \tag{41}$$

Equations (36), (37), (38) and (41) allow the recursive evaluation of $P[CP \leq r]$.

3.9 Probability Distribution Function of the Bits Dropped During a Congestion Period

To evaluate the probability distribution function of the number of bits dropped during a congestion period, we model the system, during this period, by the Markov chain of the Figure 3.8, as before.

The appropriated reward function related to the transition of $\mathbf{Z_1}$, in order to model the number of bits dropped in each state, is:

$$\eta_2(i,f) = \begin{cases} 1; & L_1 < i \leq L_2 \\ 2; & L_2 < i < K \\ 0; & i = a \end{cases} \tag{42}$$

In this function, each time the system visits the state e_i, $L_1 < i \leq L_2$, it gains a reward equals to 1, modeling, thus, the one bit dropped per sample each time that the system remains in one of these states. When the system visits a state e_i, $L_2 < i < K$, it gains a reward equal to 2, modeling the two bits dropped in these states.

Let $P\left[BD \leq r\right]$ be the probability that the bits dropped during a congestion period be less than or equal to a certain number r, and $n_b = \lceil r \rceil$. Then, similarly to the calculation of $P[CP \leq r]$, we have:

$$P\left[BD \leq r\right] = G_a(n_b, r) \tag{43}$$

The evaluation of $G_a(n_b, r)$ is accomplished using equations (36) to (38), and the set of probabilities α_f, $L_1 < f < K$, given by equation (41).

3.10 Probability Distribution Function of the Non-Congestion Period Length

Using the same method of the Section 3.8 we can evaluate the distribution of a non-congestion period size. In Figure 3.10 we show the transition probabilities matrix Z_2 of the Markov chain that models the system behavior during a non-congestion period. In this chain, the state e_i, $0 < i \leq L_1$, corresponds to the number of packets in the multiplexer at the embedded points. The state e_b is an absorbent state used to model the end of a non-congestion period.

The reward assignment function for this case is:

$$\eta_3(i,f) = \begin{cases} D_4; & 0 < i \leq L_1 \\ 0; & i = b \end{cases} \tag{44}$$

$$
\begin{array}{c|ccccccc|}
 & 0 & 1 & 2 & \ldots & L_1-1 & L_1 & b \\
0 & a_0 & a_1 & a_2 & & . & . & 1-\sum \\
1 & a_0 & a_1 & a_2 & & . & . & 1-\sum \\
2 & 0 & a_0 & a_1 & & . & . & 1-\sum \\
. & & & & & & & 1-\sum \\
. & & & & & & & 1-\sum \\
. & & & & & & & 1-\sum \\
L_1-1 & & & & a_0 & a_1 & a_2 & 1-\sum \\
L_1 & & & & & a_0 & a_1 & 1-\sum \\
b & & & & & & & 1
\end{array}
$$

Figure 9: Transition probabilities matrix Z_2 of the Markov chain modelling the system behavior during non-congestion period

This reward assignment function states that the system gains a reward equal to D_4 in each transition that starts in a non-congested state, and gains a reward equal to 0 after the Markov chain falls into the absorbent state e_b. Let $P[NCP \leq r]$ be the probability that a non-congestion period be less than or equal to r time units, and $n_c = \lceil \frac{r}{D_4} \rceil$. Then, we have:

$$P[NCP \leq r] = G_b(n_c, r) \tag{45}$$

where $G_b(n_c, r)$ is evaluated using the method of Section 3.8. Additionally, because all the non-congestion intervals start in state e_{L_1}, we have:

$$\alpha_i = \begin{cases} 0; & 0 \leq i < L_1 \\ 1; & i = L_1 \end{cases} \tag{46}$$

3.11 Mean Values Characterization

So far we have arrived at a detailed characterization of congestion periods by means of the distribution functions. Nevertheless, it is useful to know the mean values of measures, that can be evaluated using their distribution functions. However, it is convenient to find a more efficient method that allows a direct evaluation. This is the objective of this section.

To state the problem, we use again the generic reward assignment function $\eta(i,j)$ of (42). Let R be the mean reward gained by the system during a congestion period. Let R_i be the mean reward gained by the system from an initial instant, at which the Markov chain is in the state e_i, $L_1 < i < K$, to the instant in which the Markov chain transits, for the first time, to the state e_{i-1} (where, for notational convenience, we consider state a being equivalent to e_{L_1}). Let $R_{i,j}$, $L_1 < j \leq i < K$ be the mean reward gained by the system in a congestion period starting in state e_i and ending when the Markov chain transits, for the first time, to the state e_{j-1}.

Starting in the state e_i, the system gains a mean reward equal to R_i until it transits for the first time to the state e_{i-1}. Then, it gains R_{i-1}, until it transits for the first time to the state e_{i-2}, and so on. Finally, when the system transits for the first time to the state e_{j-1}, it gains

R_j. Thus, by definition of R_i and $R_{i,j}$, we have:

$$R_{i,j} = \sum_{l=i}^{j} R_l \tag{47}$$

As before, $\alpha_j(0)$ is the probability that the Markov chain starts a congestion period in the state e_j, $L_1 < j < K$. Conditioning on the state in which the system is in the beginning of the congestion period, we have:

$$R \; = \; \sum_{i=L_1+1}^{K-1} \alpha_i(0) R_{i,L_1+1} \tag{48}$$

Applying (47) and interchanging the order of the summation:

$$R = \left(\sum_{l=L_1+1}^{K-1} R_l \right) \sum_{i=l}^{K-1} \alpha_i(0) \tag{49}$$

where $\alpha_i(0)$, $L_1 < i < K$, is given by equation (41).

To evaluate R, it remains to find an expression for R_l, $L_1 < l < K$. This is recurrently done on the value of l, starting at $L = K-1$, until L_1+1, as follow. To evaluate R_{K-1}, we condition on the first transition of the markovian chain $\mathbf{MCZ_1}$, from where:

$$R_{K-1} = \eta(K-1, K-1) + (1-c_0) R_{K-1} \tag{50}$$

From 50, we have:

$$R_{K-1} = \frac{\eta(K-1, K-1)}{c_0} \tag{51}$$

Now, for $L_2 < l < K$, we condition again on the first transition of the Markov chain:

$$R_l = \eta(l,l) + \sum_{m=l}^{K-2} c_{m-l+1} R_{m,l} + R_{K-1,l} \left(1 - c_0 - \sum_{m=l}^{K-2} c_{m-l+1} \right)$$

Using (47) and after some algebraic manipulation, we arrive to:

$$R_l = \frac{1}{c_0} \left\{ \eta(l,l) + \sum_{n=l+1}^{K-1} R_n \left(1 - \sum_{k=0}^{n-l} c_k \right) \right\}, \qquad L_2 < l < K \tag{52}$$

For $L_1 + 1 \leq l < L_2$, we have:

$$R_l = \frac{1}{b_0} \left\{ \eta(l,l) + \sum_{n=l+1}^{K-1} R_n \left(1 - \sum_{k=0}^{n-l} b_k \right) \right\} \tag{53}$$

3.12 Mean Length of a Congestion Period

To calculate the mean value of the length of a congestion period we use the recurrent methodology, described in the section 3.11 and the reward assignment function given in equation 39.

Initial State	End State	Probability	
$(1,j)$	(c)	1	$1 \le j \le K-1$
(i,j)	$(i-1,j-1)$	a_0	$1<i\le<j\le L_1$
(i,j)	$(i-1,j-1)$	b_0	$1<i\le j,\ L_1<j\le L_2$
(i,j)	$(i-1,j-1)$	c_0	$1<i\le j,\ L_2<j\le K-1$
(i,j)	$(i-1,k)$	a_{k-j+1}	$1<i\le j\le L_1$ $j\le k\le K-1$
(i,j)	$(i-1,k)$	b_{k-j+1}	$1<i\le j,\ L_1<j\le L_2$ $j\le k\le K-1$
(i,j)	$(i-1,k)$	c_{k-j+1}	$1<i\le j,\ L_2<j\le K-1$ $j\le k\le K-1$

Figure 10: Transition probabilities of the Markov chain that models the residence time of a packet in the multiplexer

3.13 Mean Number of Bits Dropped during a Congestion

Similarly, to calculate the mean number of bits dropped during a congestion period, we use the method explained in section 3.11 and the reward assignment function defined in the equation 42.

3.14 Probability Distribution Function of the Packet Residence Time

One of the critical issues in the operation of high speed networks is its ability to comply with real time requirements. One of this is the packet transfer delay, whose evaluation is very important. For this purpose, it is not enough to evaluate the mean residence time of the packets in a multiplexer. It is necessary to know the distribution function of this time. When the multiplexer operates with the bit dropping policy, the evaluation of such distribution is a difficult task, because, as was mentioned in [11], the residence time of a packet $\mathcal{M}$ (from now on called *marked packet*) depends not only on the packets that are in the multiplexer before $\mathcal{M}$ arrives, but also on the packets that arrive after $\mathcal{M}$.

To overcome this difficulty, we will use the Markov chain, of Figure 11, with transition probabilities matrix Z_3. The state (i,j) of this Markov chain means that, at an embedded point, the queue length is j and the position of the *marked packet* is i. The state c is an absorbent state. The transition probabilities are calculated as showed in the Figure 10, where the probabilities a_k, b_k and c_k are given by equation 2.

To model the residence time of the marked packet, we associate each transition of the Markov chain Z_3 to a specific reward. The basic idea is to associate a reward, equal to the time the system remains in a state, to each departure transition out of this state. The system starts to gain reward at the moment the marked packet arrives, and stops gaining reward when this packet is transmitted. Thus, the reward gained by the system corresponds to the residence time of the marked packet. In addition, to model the transmission of the marked packet, we use the absorbent state, in which the system does not gain reward. As a consequence, the evaluation of the probability distribution function of the marked packet residence time is equivalent to the

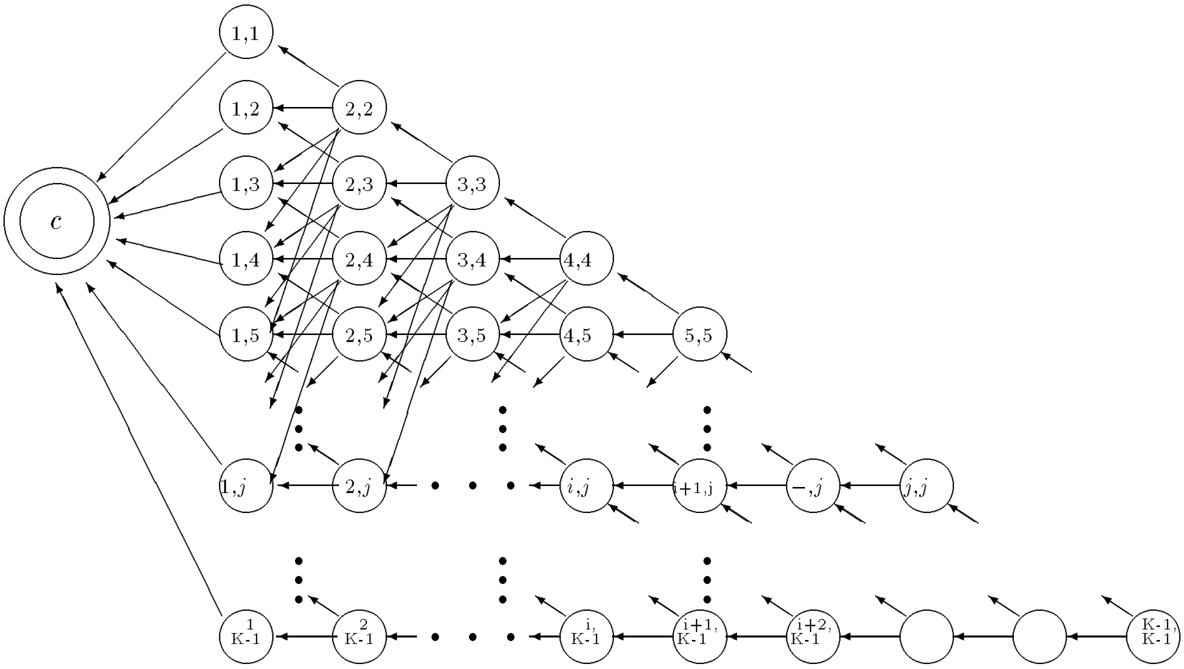

Figure 11: Markov Chain $\mathbf{Z}_3$ used to model the residence time in the multiplexer

evaluation of the distribution function of the reward earned by the multiplexer.

More precisely, we define the function η_4, which associates rewards to the output transitions of the state (i, j), $1 \leq i \leq j < K$, as follows:

$$\eta_4(i,j) = \begin{cases} D_4, & j = 0, 1, 2, \ldots, L_1 \\ D_3, & j = L_1 + 1, \ldots, L_2 \\ D_2, & j = L_2 + 1, \ldots, K - 1 \\ 0, & otherwise \end{cases} \tag{54}$$

To evaluate the probability distribution function of the marked packet residence time in the multiplexer, we use the procedure explained in Section 3.7. In particular, for this case, we define $G(r)$ as the residence time probability distribution function:

$$G(r) = P\left[\xi \leq r\right] \tag{55}$$

where ξ is the random variable that corresponds to the residence time of a marked packet in the multiplexer.

As the residence time of a packet in the multiplexer ends when the system enters into the absorbent state e_c, and as the maximum number of transitions that the marked packet may go through, until the system reach the absorbent state, is $K - 1$ (see Figure 11), we have:

$$G(r) = G_c(K - 1, r) \tag{56}$$

The function $G_c(-,r)$ is evaluated after $K-1$ transitions, because at most in $K-1$ transitions the system falls into the absorbent state e_c, modeling the end of the residence time, independent of the state of the system when the marked packet arrives.

To evaluate $G_c(K-1,r)$ we must evaluate $P[RTAD_c(K-1)=w]$, for different w values. This can be done in a recurrent procedure, as follows:

$$P\left[RTAD_{i,j}(s)=w\right]=\sum_{(k,l)} z_{(k,l)(i,j)}P\left[RTAD_{k,l}(s-1)=w-\eta_4(k,l)\right] \tag{57}$$

The initial condition $(s=0)$ is given by $\mathcal{R}^0_{i,j}=\{0\}$, so that:

$$P\left[RTAD_{i,j}(0)=w\right]=\alpha_{i,j}(0)I\left[w=0\right] \tag{58}$$

where $\alpha_{i,j}(0)$ is the probability that the system falls into state (i,j) when the marked packet arrives.

To solve $G(r)$, it remains to evaluate $\alpha_{i,j}(0)$. The exact evaluation of $\alpha_{i,j}(0)$ is not easy, principally due to the residual time of the packets in transmission when the marked packet arrives. To overcome this difficulty we evaluate an approximation of $\alpha_{i,j}(0)$. Note that during the service of one packet, the probability of only one packet arrival is greater than the probability of arriving two or more packets (because in the normal operation of the multiplexer the transmission time of one packet is lower than the packets interarrival time). Thus, assuming that when the marked packet arrives no other packet arrives in the same interval, we have:

$$\alpha_{i,j}(0)=\begin{cases}\pi_0+\pi_1 & i=j=1\\ \pi_i & i=j>1\\ 0, & i\neq j\end{cases} \tag{59}$$

Equations (54) to (58) compose an efficient solution for calculating the probability distribution function $G(r)$ of the packet residence time in the multiplexer.

We conclude this section remarking that the recurrent solution for the evaluation of G_r is made using equations (36) to (38).

4 Numerical examples

This section shows some numerical examples. For all of them , the voice activity (percentage of the talkspurt) is considered to be equal to 35%. The packet service time is given in the table of Figure 4 for a T_1 channel (1.536 Mbps), and the bit dropping thresholds are: $L_1=5$ ms, $L_2=10$ ms and $K=20$ ms, which correspond to 13, 26 and 52 packets, respectively. In case of a DS3 channel (39.744 Mbps), the limits defined for bit dropping are the same in milliseconds, but they correspond to 336, 672 and 1345 packets, respectively.

Figure 12 shows the mean number of packets in the system versus the load, which is expressed as the used bandwidth percentage of the T_1 channel, (in case that all packets were transmitted in a D_4 time interval). Note that the load may go beyond the 100 %, since, when it is high, the bit dropping algorithm transmits packets in D_3 or D_2 time units. The curve labeled D_4 corresponds to a multiplexer that transmits all its packets in a D_4 time interval, that is: a multiplexer that doesn't use the bit dropping algorithm. Analogously, the curves labeled D_3 and D_2 correspond to the cases where all packets are transmitted in a D_3 and D_2 time interval, respectively. The curve labeled *Dropping* corresponds to a multiplexer that operates with the bit dropping algorithm. It is noteworthy that the Dropping curve corresponds to an interpolation of the others. The Dropping curve accompanies the curve having service time equal to D_4 until the buffer stores approximately L_1 packets, from there on, a load increase led the bit dropping curve closer to that one with service time equal to D_3, and afterwards closer to the curve with service time D_2.

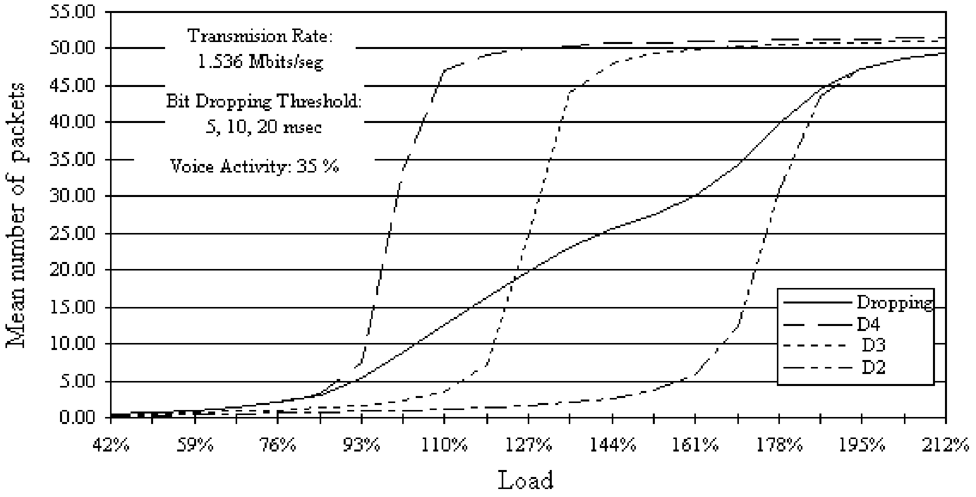

Figure 12: Mean number of packets in the system for a transmission rate of 1.536 Mbits/seg.

Figure 12 shows that the thresholds must be carefully chosen to maintain the interpolation between the curves without bit dropping smooth. It is desirable that a steep increase of the mean number of packets be avoided. To better exemplify this observation, Figure 13 shows the mean number of packets in the multiplexer for a DS3 channel, and with same thresholds of Figure 12. In the Figure, the interpolation is not smooth. To have a smooth curve, the thresholds must be lower and closer than those ones.

Nevertheless, Figure 13 is very interesting to analyze. When the load is less than 90 %, the mean number of packets in the buffer is near 0. An increase in the load to 99 % causes a fast increase in the number of packets. A load over 99 % leeds the number of packets quickly to 336. In such situation, all packets are transmitted in D_3 time units, and a load of $x\%$ will really mean $x(58/74)\%$, since only 58 out of 74 bytes of the packet are transmitted, as shown in Figure 4. Thus, it is easy to prove that approximately 126 % and 174 % of load, corresponds to a real 99 % load, when all packets are transmitted with 58 and 42 bytes, respectively. As a consequence, the behavior of the curve around these loads is analogous to the behavior with 99 % of load, but adapted to the corresponding parameters.

Another conclusion that can be extracted from Figures 12 and 13: there is a close relationship between the thresholds, the mean number of packets and the number of bits dropped. If the

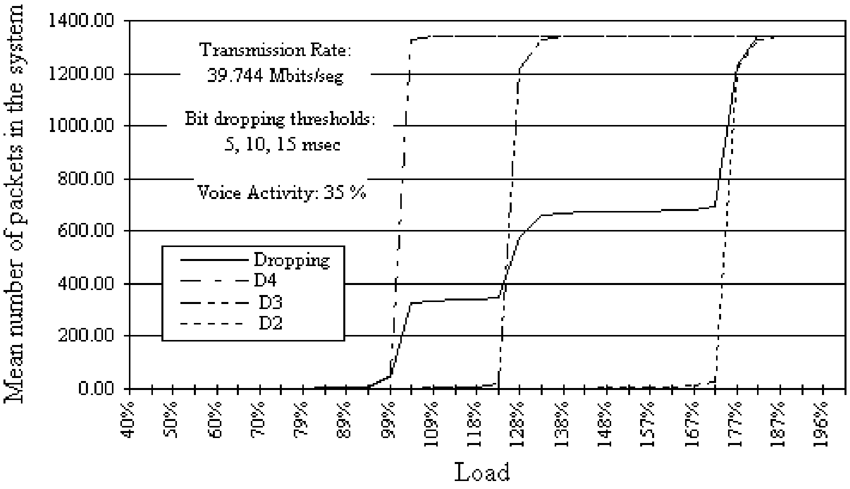

Figure 13: Mean number of packets in the system. Transmission rate 39.744 Mbits/seg

threshold decreases, the mean number of packets increase in a smoother way, and the number of bits dropped increases. Then, a trade-off between these variables clearly exists.

For another kind of experiment, Figure 14 shows the probability distribution function of the packet residence time, for a T_1 channel and with the same thresholds defined for Figure 12. As before, the time axis is expressed in milliseconds. The curves are plotted for different percentages of occupation of the T_1 channel. As can be seen, for a channel occupation less than or equal to 76% (equivalent to 90 voice sources) the packets experiment low residence time in the multiplexer with high probability. For example, with probability 0.5 the packets experiment a delay less than or equal to 0.365 ms. On the other hand, for a channel occupation of 102% (equivalent to 120 voice sources), the packets in the multiplexer experiment a higher residence time with high probability. For the case corresponding to 127% of load (150 voice sources), the packets experiment a time residence even higher. Obviously, this behavior takes place because, in spite of the bit dropping mechanism, the number of packets in the buffer increases. The total effect of the bit dropping mechanism is to smooth up the increase of the packet residence time, when the load increases.

Still in Figure 14, we can note that the curves have different characteristics for low and high load. In the case of low load (51 % or less) the curve shows significant jumps, since the probability mass is concentrated in few states. Each jump corresponds approximately to the probability that the packet finds the system in one of these states. For instance, for 51 % of load, the marked packet will find, with high probability (corresponding to the first vertical step of the curve), the buffer empty, and consequently it will stay in the system a time equal to D_4. Also, with high probability, the arriving packet will find one packet ahead it, and will stay in the

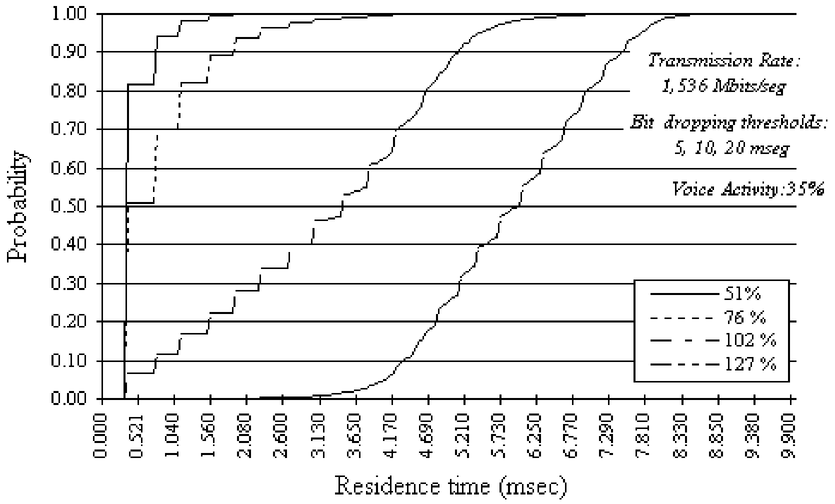

Figure 14: Residence Time Distribution

system a time equal to $2 * D_4$. More generally, due to low load, the multiplexer's buffer reach or pass over the threshold limit L_1 with low probability, which implies that the packet residence time is a multiple of D_4. This fact can be noted in the figure, where all horizontal steps are equally spaced (D_4 time units). A similar explanation can be given for the curve with 76 % of load.

For the cases of high load (102 % or 127 %), the curves are smoother and do not have any steep jump in both axes. This can be explained due to the fact that the probability mass is distributed in more states, which implies that the delay will be a linear combination of the transmission times D_2, D_3 and D_4. The only (evident) fact that we can state, is that when the marked packet arrives and finds $(n-1)$ packets in the multiplexer, its residence time will lie between $D_2 * n$ and $D_4 * n$.

5 Conclusions

This paper puts forward the performance evaluation of a packet multiplexer working with the bit dropping algorithm. Some of the results already presented in the literature were extended to be valid at any time instant instead of only at the departure instants of the packets from the multiplexer. This is the case of the mean value measures of subsection 3.3 to 3.6. Additionally, we presented a more efficient method to evaluate the steady state probabilities of the system state at the packets departure instant. These probabilities are important in the calculation of several performance measures.

The most important contribution, however, was the definition of a method through which some new measures can be evaluated, as was presented in Section 3.7. This method was used in the paper to evaluate the probability distribution function of a packet residence time in the multiplexer, a very complex task, as pointed out in [11]. The method was also applied in the evaluation of different measures during congestion and non-congestion periods. Among them we evaluate the probability distribution function of the congestion period length, the distribution function of the number of packets dropped by the bit dropping algorithm during congestion, and the distribution function of the non-congestion period length. The mean values of these measures were also evaluated.

All the evaluated measures permit to characterize the multiplexer behavior in detail, and, as a consequence, greatly help us to understand it and design it more efficiently.

ACKNOWLEDGEMENTS

The authors would like to thank to Gustavo Welkner for the implementation of the software tool used in the numerical evaluation of the equations.

References

[1] D.O. Bowker and C.A. Dvorak. Speech transmission quality of wideband packet technology. In *IEEE GLOBECOM'87*, pages 1887–1889, Nov 1987.

[2] R. Vallejos C. *Performability Models and Its Aplications to Computer Systems.* PhD thesis, Federal University of Rio de Janeiro, Brasil, January 1993. (writted in portuguese).

[3] S. Karlin and H. Taylor. *A Second Course in Stochastic Processes.* Academic Press,Inc., 1981.

[4] San-Qi Li. Study of information loss in packet voice systems. *IEEE Transactions on Communications*, 37(11):1192–1202, November 1989.

[5] J. F. Meyer. On evaluating the performability of degradable computing systems. *IEEE Transactions on Computers*, C-29(8):720–731, 1980.

[6] I. Nikolaidis and R. Onvural. A bibliography on performance issues in ATM networks. *Computer Communications Review*, pages 8–23, 1992.

[7] R. Rom and M. Sidi. *Multiple Access Protocols. Performance and Analysis.* Springer-Verlang, 1990.

[8] S. M. Ross. *Introduction to Probability Models.* Academic Press, Inc., 5th. edition, 1993.

[9] M.H. Sheriff, A. D. Malaret-Colazo, and M. C. Gruensfelder. Wideband packet technology in the integrated access and cross–connect systems (IACS). *Int. J. Satell. Commun.*, (8):437–444, 1990.

[10] L.F.G. Soares, G. Lemos, and s. Colcher. *Redes de Computadores. Das LANs, MANs e WANs às Redes ATM.* Editora Campus, Rio de Janeiro, 2nd. edition, 1995.

[11] K. Sriram and D. Lucatoni. Traffic smoothing effects of bit dropping in a packet voice multiplexer. *IEEE Transactions on Communications*, 37:703–712, July 1989.

[12] K. Sriram and W. Whitt. Characterizing superposition arrival process in packet multiplexers for voice and data. *IEEE J. Select Areas Communic.*, SAC–4:833–846, Sept. 1986.