BLOCKING IN TELETRAFFIC SYSTEMS UNDER NONSTATIONARY ARRIVAL AND SERVICE CONDITIONS

by

Muhammad Naim Yunus BSc (Adelaide)

Thesis submitted for the degree of
Doctor of Philosophy
in the
Department of Applied Mathematics
University of Adelaide
October 1985

Awarded 3RD APRIL 1986

CONTENTS

CHAPTER I : INTRODUCTION

CHAPTER II : EXACT TRANSIENT BLOCKING PROBABILITY FOR THE M/M/N LOSS SYSTEM

CHAPTER III : APPROXIMATION OF THE TRANSIENT BLOCKING PROBABILITY FOR THE M/M/N LOSS SYSTEM

CHAPTER IV : APPROXIMATION OF THE M(t)/M(t)/N LOSS SYSTEM BY THE M/M/N LOSS SYSTEM

CHAPTER V : TIME-DEPENDENT BLOCKING PROBABILITY FOR THE M(t)/M(t)/1 LOSS SYSTEM

SUMMARY

This thesis contains work on time-dependent blocking probability in both the M/M/N and the M(t)/M(t)/N loss systems, which have wide application in teletraffic studies. Research commenced in March, 1983 and was completed in October, 1985.

First the M/M/N loss system is considered. A new exact method to determine time-dependent or transient blocking probability for this system is outlined. However for large N this exact method is unsuitable due to computational difficulties and approximation methods are introduced for this case. All these approximations use the Erlang Loss Function, but with its argument, the offered traffic, being replaced with some other time-dependent functions. Comparison of these approximations with simulation and exact results, where available, is made.

Then the problem of blocking in the more general M(t)/M(t)/N loss system is considered. Firstly, this nonstationary system is approximated by a succession of M/M/N systems over time. Then the methods previously developed for the M/M/N loss system are applied to these systems to yield approximations to the time-dependent blocking probability of the M(t)/M(t)/N system. Initially the simple M(t)/M(t)/1 loss system is treated followed by the more complicated M(t)/M(t)/N system.

Finally a study is made of the expectation and the variance of traffic overflowing from a loss system experiencing nonstationary arrival streams. Approximate methods are outlined for estimating these two moments and they are found to be adequate for practical purposes. Their application in a modified Equivalent Random Method, which is important in dimensioning problems, is also given together with an illustration.

SIGNED STATEMENT

This thesis contains no material which has been accepted for the award of any other degree or diploma in any University. To the best of my knowledge and belief, the thesis contains no material previously published or written by any other person, except where due reference is made in the text. I also consent to the thesis being made available for photocopying and loan if accepted for the award of the degree.

Muhammad Naim Yunus

ACKNOWLEDGEMENTS

I would like to express my appreciation for the encouragement and guidance given by my supervisor, Dr. L.T.M. Berry, during the course of this research. I am also grateful to Andrew Coyle for his assistance in some of the computer programming and also for some helpful discussions and ideas. I would also like to thank Prof. R.B. Potts and Dr. O. Martikainen for their interest in my research.

This thesis is dedicated to my wife, Sabariah.

CHAPTER I

INTRODUCTION

1.1 Preamble

In the study of teletraffic, there are still a lot of unsolved problems, many of them with considerable practical importance. This is despite the large amount of research in this area. One basic problem in teletraffic is the problem of network dimensioning, that is to provide the optimal number of trunks in a network to carry a specified amount of traffic. This problem necessitates the study of the blocking probability of the trunks. This is the probability that all trunks are occupied for a specified number of trunks and given arrival and service processes. We'll assume that a trunk can only carry one call at a time and that rejected calls are lost. The blocking probability is important because it is used in the allocation of trunks to satisfy specified network grades-of-service.

It has been widely accepted that the arrival process is a Poisson process and the holding time follows the negative exponential distribution. For the case of the homogeneous Poisson process, a lot of research has been done. A very well-known formula for this particular process is the Erlang Loss Function,

$$E_N(a) = \frac{a^N/N!}{\sum_{k=0}^{N} a^k/k!}$$

where N is the number of trunks, and a is the offered traffic, which will be discussed later. $E_N(a)$ is the probability that all N trunks in a steady state system are occupied, assuming that the calls arrive according to a Poisson process with a constant parameter and that full-availability holds. The formula is valid for arbitrary service time distribution (Syski [24]). It has been used for dimensioning purposes but due to the assumption of the steady state, it does not give accurate results. This is simply because real-life traffic is time-dependent (Myskja [17,18,19]), Martikainen [11], Martikainen and Lehtinen [10, 12].

Therefore research has to be done in developing methods to compute time- dependent blocking probability due to the time-varying nature of traffic. Pioneering work in this area has been done by Palm [21]. He considers the problem of blocking with time-varying offered load for finite trunk groups. Khintchine [9] considers the same problem for infinite trunk groups. Palm [21] proposes his 'slow variations'model in which he assumes that if $a(t)$, the offered traffic, varies slowly with time, then its time-dependency could be ignored. Instead $a(t)$ could be replaced by a random variable with an incomplete gamma function distribution (Jagerman [8]). However, this assumption is not valid because $\frac{da(t)}{dt}$ can be very high at times.

A substantial advance in this area of research is made by Jagerman [8]. He obtains integral equations for the time-dependent blocking probability resulting from non-stationary arrival streams. Unfortunately, these integral equations are very difficult to solve. Jagerman also derives some very useful formulae to approximate the blocking probability. However these approximations are only for transient blocking in the M/M/N loss system, with constant mean service time.

This thesis represents work done on the problem of transient blocking for the M/M/N loss system and time-dependent blocking for the M(t)/M(t)/N loss system. Both these systems model calls arriving to a group of N trunks. The motivation for studying the transient blocking probability of the M/M/N loss system lies in its application to approximate time-dependent blocking probability of the M(t)/M(t)/N loss system as will be shown.

Chapters II and III contain original work dealing with the transient blocking probability for the M/M/N loss system. In this system we consider the case when a sudden, instantaneous surge of traffic held at a constant intensity is offered to an initially empty group of N trunks. We are interested in the transient blocking probability of the system. An exact, analytical method and an approximation method are given to find this probability. The exact method (Chapter II) is, however, not suitable for large N due to computational difficulties, whereas the approximate method (Chapter III) can be used for any N, regardless of its magnitude. A paper based on Chapter II has been presented at a conference (Yunus [28]).

Methods developed in Chapters II and III are then extended to the more realistic problem of blocking in the M(t)/M(t)/N loss system. Chapter IV discusses the use of a succession of the M/M/N loss system to approximate the M(t)/M(t)/N loss system. This is done by approximating both the arrival and service rates in the M(t)/M(t)/N loss system with step functions, that is we discretize them. This technique is not original, but the idea of approximating the M(t)/M(t)/N loss system with the M/M/N loss system using it is certainly novel.

We then apply the above technique to the simple M(t)/M(t)/1 loss system in Chapter V and approximate its time-dependent blocking probability. This work has been published (Yunus [27]). In Chapter VI the more complicated M(t)/M(t)/N loss system is considered. A numerical procedure, based on the above discretization, is outlined. This procedure is feasible only for small and medium N because for large N the errors are too big. This research has been accepted for publication, see [3].

In Chapter VII a few approximate methods for large N are given and discussed. One of them is a new method whereas the others are modifications of other approximations. The advantage of this new approximation lies in its simplicity and it uses very little computer time, which is important when dealing with large systems. These methods are compared with numerical results.

Chapter VIII discusses time-dependent lost traffic and variance of traffic overflowing from a loss system experiencing a nonstationary arrival stream. These moments are crucial in dimensioning problems. Their application in such problems will be illustrated.

Therefore the thesis begins with a consideration of the M/M/N loss system and then goes on to the more general M(t)/M(t)/N loss system. Finally it treats the two time-dependent moments of the overflow stream for use in dimensioning problems involving time-dependent streams.

1.2 Nonstationary Arrival Process

We are interested in traffic coming from a nonstationary stream, in particular the nonhomogeneous Poisson process. As mentioned in the previous section, this process models real-life telecommunications traffic satisfactorily. Myskja [17,18,19] made an extensive study on the process. We have

$$p_k(t, t+\Delta t) = exp\left(-\int_t^{t+\Delta t} \lambda(u)du\right) \frac{\left(\int_t^{t+\Delta t} \lambda(u)du\right)^k}{k!} \qquad (1.2.1)$$

where $k = 0, 1, 2, \ldots$, as the probability that k calls arrive in the time interval $[t, t+\Delta t]$ when the arrival rate is $\lambda(t)$. This process is similar to the normal homogeneous Poisson process, except that its parameter, $\lambda(t)$, varies with time (Khintchine [9], Gnedenko and Kovalenko [6]). This implies that the probability of k calls arriving in the time interval of length Δt depends not only on Δt, but also on the initial instant, t, of this time interval.

Let us define the functions

$$\begin{aligned} \pi_1(t, t+\Delta t) &= \sum_{k=1}^{\infty} p_k(t, t+\Delta t) \\ &= 1 - p_0(t, t+\Delta t) \\ \pi_2(t, t+\Delta t) &= \sum_{k=2}^{\infty} p_k(t, t+\Delta t) \\ &= 1 - p_0(t, t+\Delta t) - p_1(t, t+\Delta t) \end{aligned} \tag{1.2.2}$$

The first equation defines the probability that at least one call arrives in $[t, t+\Delta t]$ and the second defines the probability that at least two calls arrive in the same interval.

Now one of the assumptions of the nonhomogeneous Poisson process is orderliness, which implies that

$$\lim_{\Delta t \to 0} \frac{\pi_2(t, t+\Delta t)}{\Delta t} = 0 \tag{1.2.3}$$

Another assumption is that, for any $t \geq 0$,

$$\lim_{\Delta t \to 0} \frac{\pi_1(t, t+\Delta t)}{\Delta t} = \lambda(t) \tag{1.2.4}$$

provided that $\lambda(t)$, the instantaneous value of the parameter, exists.

It can also be shown that the instantaneous intensity of an orderly stream without after-effects is the instantaneous value of the parameter, $\lambda(t)$ (Khintchine [9]).

1.3 Nonstationary Service Process

The service process in teletraffic network is modelled by the negative exponential distribution. In the nonstationary case, the distribution is

$$p(t \leq \tau \leq t + \Delta t) = 1 - exp\left(-\int_{t}^{t+\Delta t} \mu(u)du\right) \tag{1.3.1}$$

where τ is the holding time, and $\mu(t)$ is the time-dependent service rate. Measurements show that the distribution of the holding times tends to be time-varying and hence the above distribution is used to model the real-life holding time distribution.

1.4 The Time-Dependent Offered Traffic

The concept of offered traffic is very important in teletraffic theory It is defined as the number of incoming calls during the average holding time. In the steady state case it is simply

$$a = \frac{\lambda}{\mu} \tag{1.4.1}$$

where a is the offered traffic, λ is the constant arrival rate and μ is the constant service rate.

For the nonstationary case, the expression is more complicated, namely

$$a(t) = \int_{0}^{t} \lambda(t-u)H^{c}(u)du \tag{1.4.2}$$

where $a(t)$ is the instantaneous offered traffic at time t, and $H^{c}(t)$ is the complementary service time distribution (Berry [2], Syski [24]). Hence for the transient case, that is in the M/M/N loss system, the transient offered traffic is

$$a(t) = \lambda \int_{0}^{t} H^{c}(u)du \tag{1.4.3}$$

One is tempted to simply use this time-dependent offered traffic, $a(t)$, in the Erlang Loss Function to obtain the transient or time-dependent blocking probability. This will be discussed in Chapter VI. It gives good approximations in some cases and the contrary in the other cases.

1.5 The M(t)/M(t)/N Loss System

In queueing theory, we can model a group of N trunks with stationary Poisson arrivals and stationary negative exponential service times, assuming full-availability, as the M/M/N loss system. Calls that arrive when all trunks are full are rejected and lost, whence the term 'loss system'. For the nonstationary case, that is when both arrivals and service times are time-dependent, we have the M(t)/M(t)/N loss system.

Analogous to the M/M/N loss system, the birth-and-death equations, a set of difference-differential equations, for the M(t)/M(t)/N loss system are

$$\begin{aligned}
\frac{dp_0(t,N)}{dt} &= -\lambda(t)p_0(t,N) + \mu(t)p_1(t,N) \\
\frac{dp_i(t,N)}{dt} &= \lambda(t)p_{i-1}(t,N) - (\lambda(t) + i\mu(t))\,p_i(t,N) + (i+1)\,\mu(t)p_{i+1}(t,N) \\
&\qquad 1 \le i \le N-1 \\
\frac{dp_N(t,N)}{dt} &= \lambda(t)p_{N-1}(t,N) - N\mu(t)p_N(t,N)
\end{aligned} \tag{1.5.1}$$

where $p_i(t,N)$ is the probability that i out of N trunks are busy. The blocking probability is denoted by $p_N(t,N)$, the probability that all N trunks are busy.

Equations (1.5.1) together with

$$\sum_{k=0}^{N} p_k(t,N) = 1 \tag{1.5.2}$$

describe the M(t)/M(t)/N loss system totally and a way of finding the blocking probability is by solving them. They can be expressed as a matrix differential equation, namely

$$\frac{d\underline{p}(t,N)}{dt} = A(t)\underline{p}(t,N) \qquad (1.5.3)$$

where

$$\underline{p}\,(t,N) = (\,p_0\,(t,N) \quad p_1\,(t,N) \quad \ldots \quad p_N\,(t,N)\,)^T$$

$$A = \begin{pmatrix} -\lambda & \mu & 0 & \ldots & 0 & 0 & 0 \\ \lambda & -\lambda-\mu & 2\mu & \ldots & 0 & 0 & 0 \\ 0 & \lambda & -\lambda-2\mu & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & -\lambda-(N-2)\,\mu & (N-1)\mu & 0 \\ 0 & 0 & 0 & \ldots & \lambda & -\lambda-(N-1)\,\mu & N\mu \\ 0 & 0 & 0 & \ldots & 0 & \lambda & -N\mu \end{pmatrix} \qquad (1.5.4)$$

where $\lambda = \lambda(t)$ and $\mu = \mu(t)$.

Solving equations (1.5.1) is similar to solving the matrix differential equation (1.5.3). Unfortunately, at the moment there is no analytical method to solve our particular matrix differential equation although methods are available for other special forms of $A(t)$ (Wu [26]). Therefore we shall have to resort to numerical methods or approximations.

CHAPTER II

EXACT TRANSIENT BLOCKING PROBABILITY FOR THE M/M/N LOSS SYSTEM

2.1 Introduction

In applied queueing theory it is very important that we consider cases when the system has not yet achieved (or may never achieve) the steady state. This could be due to the resumptions of service (start-ups) due to breaks, such as a breakdown. It could also be due to nonstationary, or time-dependent arrival traffic.

When traffic arrives at a system which has not achieved the steady state, stochastic models based on the assumption of transience have to be used to find blocking probability, queue length, delay probability, etc. Application of models which assume the steady state gives incorrect results. The steady state can be achieved only when traffic is stationary, that is constant over time. If traffic is nonstationary, for example the M(t)/M(t)/N loss system, then the steady state is never achieved because of the nature of the arriving traffic. However we can have the transient state with stationary traffic, i.e. before the system reaches the steady state. This arises when the arrival stream begins abruptly, like a step function, at time τ, say, and stays at a constant intensity thereafter. The time taken for the system to achieve the steady state after τ will depend on a number of factors, as shall be shown later.

This chapter considers the M/M/N loss system, i.e. with no waiting space, which has wide application in teletraffic engineering. Only Riordan [23] has given an exact method to find the transient blocking probability for the above system. However, his method uses Laplace transforms which are difficult to invert in the case of all trunks not being free initially. Surprisingly enough, bearing in mind the importance of the transient blocking probability, nobody, to the best of our knowledge, has given a method which is better than Riordan's and which is applicable for any initial state. The procedure outlined evaluates exact blocking probability for this system for any initial state.

2.2 Transient Blocking Probability

We consider the M/M/N loss system with the following arrival rate

$$\lambda(t) = \begin{cases} 0, & t \leq 0 \\ \lambda, & t > 0 \end{cases}$$

The set of difference-differential equations associated with this system (after normalizing the mean service time to make it unity) is

$$\begin{aligned} \frac{dp_0\,(t,N)}{dt} &= -\lambda p_0\,(t,N) + p_1\,(t,N) \\ \frac{dp_i\,(t,N)}{dt} &= \lambda p_{i-1}\,(t,N) - (\lambda + i)\,p_i\,(t,N) + (i+1)\,p_{i+1}\,(t,N) \\ & \qquad 1 \leq i \leq N-1 \\ \frac{dp_N\,(t,N)}{dt} &= \lambda p_{N-1}\,(t,N) - Np_N\,(t,N) \end{aligned} \tag{2.2.1}$$

where $p_i\,(t,N)$ is the transient probability that i out of N servers are busy and λ is the constant arrival rate. The transient blocking probability is given by $p_N\,(t,N)$. However to obtain this probability the above set of equations (2.2.1) must be solved and this is a very difficult exercise.

Equations (2.2.1) can be written as a matrix differential equation, that is

$$\frac{d\underline{p}\,(t,N)}{dt} = A\underline{p}\,(t,N) \tag{2.2.2}$$

where

$$\underline{p}\,(t,N) = (\,p_0\,(t,N) \quad p_1\,(t,N) \quad \ldots \quad p_N\,(t,N)\,)^T$$

$$A = \begin{pmatrix} -\lambda & 1 & 0 & \cdots & 0 & 0 & 0 \\ \lambda & -\lambda-1 & 2 & \cdots & 0 & 0 & 0 \\ 0 & \lambda & -\lambda-2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda-(N-2) & N-1 & 0 \\ 0 & 0 & 0 & \cdots & \lambda & -\lambda-(N-1) & N \\ 0 & 0 & 0 & \cdots & 0 & \lambda & -N \end{pmatrix} \tag{2.2.3}$$

Matrix equation (2.2.2) can be solved to obtain

$$\underline{p}\,(t,N) = e^{At}\underline{p}\,(0,N) \tag{2.2.4}$$

where $\underline{p}\,(0,N)$ is the probability distribution vector at $t = 0$. The required blocking probability is the last element in the vector $\underline{p}(t,N)$. The problem lies in the evaluation of e^{At}. Many procedures are available (see Moler and van Loan [16]) and the choice of which procedure to adopt depends on λ and N.

The tridiagonal matrix A can be shown to have $(N+1)$ real, nonpositive distinct eigenvalues with one of them being zero (Bellman [1]). The method to be outlined is based on the fact that every element of the matrix e^{At} is a linear combination of the exponentials of these eigenvalues, multiplied by the factor t, the time (Reid [22]). In fact Riordan's expression for the transient blocking probability is expressed as a linear combination of these exponentials. Therefore we would expect $\underline{p}\,(t,N)$ to consist of two parts: one part consisting of the coefficient vector to $e^{0.t}$ which corresponds to the steady state solution, and the other part consisting of the linear combination of the exponentials of the other negative eigenvalues and time, which corresponds to the transient part of the solution.

We therefore have, writing $\underline{p} = \underline{p}(t, N)$

$$\underline{p} = \underline{b} + \left(\underline{c}_1 e^{r_1 t} + \underline{c}_2 e^{r_2 t} + \cdots + \underline{c}_N e^{r_N t}\right) \tag{2.2.5}$$

where r_1, r_2, ...,r_N are the distinct, negative eigenvalues of A, and $\underline{b}$ is

$$\begin{aligned} \underline{b} &= \beta\left(1 \quad \lambda \quad \tfrac{\lambda^2}{2!} \quad \tfrac{\lambda^3}{3!} \quad \cdots \quad \tfrac{\lambda^N}{N!}\right)^T \\ \beta^{-1} &= \sum_{k=0}^{N} \frac{\lambda^k}{k!} \end{aligned} \tag{2.2.6}$$

Note that the last element in $\underline{b}$ is the well-known Erlang Loss Function, the steady state blocking probability. The steady state exists when $\frac{d\underline{p}(t,N)}{dt}$ in (2.2.2) is zero, i.e. $A\underline{p} = \underline{0}$, and we have $\underline{p} = \underline{b}$ as the corresponding solution. This is easily shown by verifying that

$$A\underline{p} = A\underline{b} = \underline{0} \tag{2.2.7}$$

The eigenvalues could be found for specific N and λ by using computer packages. Our next task is to find the coefficient vectors, $\underline{c}_k$. We now differentiate (2.2.5) with respect to time repeatedly to obtain

$$\begin{aligned} \underline{p}(t, N) &= \underline{b} + \underline{c}_1 e^{r_1 t} + \underline{c}_2 e^{r_2 t} + \cdots + \underline{c}_N e^{r_N t} \\ \frac{d\underline{p}(t, N)}{dt} &= \underline{c}_1 r_1 e^{r_1 t} + \underline{c}_2 r_2 e^{r_2 t} + \cdots + \underline{c}_N r_N e^{r_N t} \\ \frac{d^2\underline{p}(t, N)}{dt^2} &= \underline{c}_1 r_1^2 e^{r_1 t} + \underline{c}_2 r_2^2 e^{r_2 t} + \cdots + \underline{c}_N r_N^2 e^{r_N t} \\ &\vdots \\ \frac{d^{N-1}\underline{p}(t, N)}{dt^{N-1}} &= \underline{c}_1 r_1^{N-1} e^{r_1 t} + \underline{c}_2 r_2^{N-1} e^{r_2 t} + \cdots + \underline{c}_N r_N^{N-1} e^{r_N t} \end{aligned} \tag{2.2.8}$$

And we also have

$$\begin{aligned} \frac{d\underline{p}(t, N)}{dt} &= A\underline{p}(t, N) \\ \frac{d^2\underline{p}(t, N)}{dt^2} &= A\frac{d\underline{p}(t, N)}{dt} = A^2\underline{p}(t, N) \\ \frac{d^3\underline{p}(t, N)}{dt^3} &= A\frac{d^2\underline{p}(t, N)}{dt^2} = A^3\underline{p}(t, N) \\ &\vdots \\ \frac{d^{N-1}\underline{p}(t, N)}{dt^{N-1}} &= A^{N-1}\underline{p}(t, N) \end{aligned} \tag{2.2.9}$$

Therefore at $t = 0$ and combining (2.2.8) and (2.2.9), we have

$$\begin{aligned}
\underline{b} + \underline{c}_1 + \underline{c}_2 + \cdots + \underline{c}_N &= \underline{p}(0, N) \\
\underline{c}_1 r_1 + \underline{c}_2 r_2 + \cdots + \underline{c}_N r_N &= A\underline{p}(0, N) \\
\underline{c}_1 r_1^2 + \underline{c}_2 r_2^2 + \cdots + \underline{c}_N r_N^2 &= A^2\underline{p}(0, N) \qquad (2.2.10) \\
&\vdots \\
\underline{c}_1 r_1^{N-1} + \underline{c}_2 r_2^{N-1} + \cdots + \underline{c}_N r_N^{N-1} &= A^{N-1}\underline{p}(0, N)
\end{aligned}$$

In matrix form the above set of equations (2.2.10) can be written as

$$V\underline{c} = \underline{d} \qquad (2.2.11)$$

where

$$\begin{aligned}
V &= \begin{pmatrix} 1 & 1 & \cdots & 1 \\ r_1 & r_2 & \cdots & r_N \\ \vdots & \vdots & \ddots & \vdots \\ r_1^{N-1} & r_2^{N-1} & \cdots & r_N^{N-1} \end{pmatrix} \\
\underline{c} &= (\, \underline{c}_1 \quad \underline{c}_2 \quad \ldots \quad \underline{c}_N \,)^T \qquad (2.2.12) \\
\underline{d} &= (\, \underline{p}(0, N) - \underline{b} \quad A\underline{p}(0, N) \quad \ldots \quad A^{N-1}\underline{p}(0, N) \,)^T \\
&= (\, \underline{d}_1 \quad \underline{d}_2 \quad \ldots \quad \underline{d}_N \,)^T
\end{aligned}$$

Now V is a well-known matrix, the Vandermonde matrix. Our particular V, made up of the distinct eigenvalues of A, is invertible and therefore

$$\underline{c} = W\underline{d} \qquad (2.2.13)$$

where $W = V^{-1}$.

Writing

$$\underline{e} = (\, e^{r_1 t} \quad e^{r_2 t} \quad \ldots \quad e^{r_N t} \,)^T$$

we have

$$\begin{aligned}
\underline{p}(t, N) &= \underline{b} + \underline{c}^T \underline{e} \\
&= \underline{b} + \underline{d}^T W^T \underline{e}
\end{aligned} \qquad (2.2.14)$$

The above equation is the exact expression for the transient blocking probability vector for any initial condition of a system with constant arrival rate and negative exponential service time of mean one. The initial condition is contained in the vector $\underline{p}(0, N)$. But for large λ and N matrix V becomes almost singular and thus cannot be inverted accurately by the computer. However there exists an analytical method to find the inverse of the Vandermonde matrix (Csaki [4]), but this method becomes cumbersome when N is large, regardless of the magnitude of λ.

If w_{ij} is the element of W, then

$$w_{ij} = \frac{L_j(r_i)}{D_i(r_i)} \tag{2.2.15}$$

where

$$\begin{aligned} L_j(r_i) &= r_i^{N-j} + \sum_{m=j}^{N-1} k_m r_i^{m-j} \\ D_i(r_i) &= \prod_{\substack{m=1 \\ m\neq i}}^{N} (r_i - r_m) \end{aligned} \tag{2.2.16}$$

and k_m are the coefficients of the polynomial

$$\begin{aligned} \chi(r) &= r^N + k_{N-1} r^{N-1} + \cdots + k_1 r + k_0 \\ &= (r - r_1)(r - r_2)\ldots(r - r_m) \end{aligned} \tag{2.2.17}$$

Using $\underline{c}^T = \underline{d}^T W^T$, we have

$$\underline{c}_k = \frac{1}{D_k(r_k)} \sum_{s=1}^{N} L_s(r_k)\underline{d}_s \qquad k = 1, 2, ..., N \tag{2.2.18}$$

Therefore

$$\begin{aligned} \underline{p}(t, N) &= \underline{b} + \sum_{k=1}^{N} \underline{c}_k e^{r_k t} \\ &= \underline{b} + \sum_{k=1}^{N} \frac{1}{D_k(r_k)} \sum_{s=1}^{N} L_s(r_k)\underline{d}_s e^{r_k t} \\ &= \underline{b} + \sum_{k=1}^{N} \left(\sum_{s=1}^{N} \frac{L_s(r_k)}{D_k(r_k)} \underline{d}_s \right) e^{r_k t} \end{aligned} \tag{2.2.19}$$

from which the transient blocking probability, $p_N(t, N)$, can be obtained.

2.3 The M/M/2 Loss System

As an illustration let us consider the M/M/2 loss system. Our polynomial (2.2.17) is now

$$\begin{aligned} \chi(r) &= r^2 + r(-r_1 - r_2) + r_1 r_2 \\ \Rightarrow k_1 &= -r_1 - r_2 \qquad\qquad (2.3.1) \\ k_2 &= r_1 r_2 \end{aligned}$$

Therefore we have

$$\begin{aligned} L_1(r_1) &= r_1 + k_1 = -r_2 \\ L_1(r_2) &= r_2 + k_1 = -r_1 \\ L_2(r_1) &= L_2(r_2) = 1 \qquad\qquad (2.3.2) \\ D_1(r_1) &= r_1 - r_2 \\ D_2(r_2) &= r_2 - r_1 \end{aligned}$$

Hence, using (2.2.19), we obtain

$$\begin{aligned} \underline{p}(t,2) &= \underline{b} + \left(\underline{p}(0,2) - \underline{b}\right)\left(-\frac{r_2}{r_1 - r_2}e^{r_1 t} - \frac{r_1}{r_2 - r_1}e^{r_2 t}\right) \\ &\quad + A\underline{p}(0,2)\left(\frac{1}{r_1 - r_2}e^{r_1 t} + \frac{1}{r_2 - r_1}e^{r_2 t}\right) \\ &= \underline{b} + e^{r_1 t}\left(\frac{r_2}{r_2 - r_1}\left(\underline{p}(0,2) - \underline{b}\right) - \frac{A\underline{p}(0,2)}{r_2 - r_1}\right) \\ &\quad + e^{r_2 t}\left(-\frac{r_1}{r_2 - r_1}\left(\underline{p}(0,2) - \underline{b}\right) + \frac{A\underline{p}(0,2)}{r_2 - r_1}\right) \end{aligned} \tag{2.3.3}$$

where $\underline{p}(0,2)$ is the initial probability vector.

Our transient blocking probability is then

$$\begin{aligned} p_2(t,2) &= E_2(\lambda) + e^{r_1 t}\left(\frac{r_2}{r_2 - r_1}\left(p_2(0,2) - E_2(\lambda)\right) - \frac{\lambda p_1(0,2) - 2p_2(0,2)}{r_2 - r_1}\right) \\ &\quad + e^{r_2 t}\left(-\frac{r_1}{r_2 - r_1}\left(p_2(0,2) - E_2(\lambda)\right) + \frac{\lambda p_1(0,2) - 2p_2(0,2)}{r_2 - r_1}\right) \end{aligned} \tag{2.3.4}$$

where $E_2(\lambda)$ is the Erlang Loss Function.

Equation (2.3.4) gives the exact expression for the transient blocking probability of the M/M/2 loss system. Note the symmetry in r_1 and r_2.

2.4 Discussion

The above analytical procedure enables us to evaluate exact transient blocking probability for the M/M/N loss system. However for a system with N trunks, there will be N^2 of the polynomials $L_j(r_i)$, and the eigenvalues of a square matrix of dimension $(N+1)$ would have to be calculated. Therefore for a large system a computer must be utilized to find the various coefficient vectors, $\underline{c}_k$. Once this difficulty is overcome, the procedure can be used in loss systems involving homogeneous Poisson arrivals and a negative exponential service time with constant mean.

CHAPTER III

APPROXIMATION OF THE TRANSIENT BLOCKING PROBABILITY FOR THE M/M/N LOSS SYSTEM

3.1 Introduction

As discussed in the previous chapter, determination of the exact transient blocking probability for the M/M/N loss system, with large N, is still a very difficult undertaking. This measure is important in the technique to be outlined later, which will use it to approximate the time-dependent blocking probability of the M(t)/M(t)/N loss system. Because of this difficulty in obtaining the exact transient blocking probability for large N we look for an approximation method which would yield results good enough for practical purposes. This method must be easy to use and quick for large N.

Utilization of the Erlang Loss Function is one way of achieving this. We can obtain good approximations of the transient blocking probability by using a modified offered traffic in the function. We can then calculate this approximation, for large N, by using the recurrence relation of the Erlang Loss Function. In Section 3.5 other approximations are also discussed.

3.2 The Erlang Loss Function

In teletraffic theory, this function is probably the most well-known. A great amount of literature has been written on it, see for example, Syski [24], Jagerman [7], Farmer and Kaufman [5], and Mejzler [14]. It is fundamental to the study of telephone trunking problems. A. K. Erlang uses $E_N(a)$,

$$E_N(a) = \frac{a^N/N!}{\sum_{k=0}^{N} a^k/k!} \tag{3.2.1}$$

where a is the offered traffic and N is the number of trunks, to express the probability that a call, which is a member of a homogeneous Poisson stream with parameter a, arriving at a group of N telephone trunks will be rejected and lost. The offered traffic, a, is defined as

$$a = \frac{\lambda}{\mu} \tag{3.2.2}$$

and thus in the case when the mean service time, $\frac{1}{\mu}$, is normalized, $a = \lambda$.

For the case when N is continuous, one of the numerous expressions for this function is

$$\frac{1}{E_N(a)} = \int_0^\infty \left(1 + \frac{x}{a}\right)^N e^{-x} dx \tag{3.2.3}$$

Its well-known and very useful recurrence relation is

$$\begin{aligned} E_{N+1}(a) &= \frac{aE_N(a)}{N+1+aE_N(a)} \\ E_0(a) &= 1 \end{aligned} \tag{3.2.4}$$

Another point to note is that the Erlang Loss Function is independent of the service time distribution, that is, it is valid for arbitrary service time distribution.

However this function is derived on the assumption of the whole system being in the steady state. The problem we are interested in is, on the other hand, transient or time-dependent and hence (3.2.1) is not valid in these cases. Therefore it seems that to make use of the Erlang Loss Function in our transient problem, we shall have to introduce a time-dependent modified offered traffic, $\alpha(t)$, say. We hope that $\alpha(t)$ will vary according to time in such a manner that $E_N(\alpha(t))$ will yield results which will approximate the true transient blocking probability.

3.3 Modified Offered Traffic for the Erlang Loss Function

The derivation of $\alpha(t)$ is fairly straightforward. Let us consider, for the moment, the first difference-differential equation in the set of the birth-and-death equations (2.2.1) for the M/M/N loss system, that is

$$\frac{dp_0(t,N)}{dt} = -\lambda p_0(t,N) + \mu p_1(t,N) \tag{3.3.1}$$

The rationale behind this choice will be discussed later. Now, we begin by assuming that there exists a function, $\alpha(t)$, such that

$$\begin{aligned} p_0(t,N) &= \frac{1}{S_N(\alpha(t))} \\ p_1(t,N) &= \frac{\alpha(t)}{S_N(\alpha(t))} \\ p_2(t,N) &= \frac{(\alpha(t))^2}{2!S_N(\alpha(t))} \\ &\vdots \\ p_N(t,N) &= \frac{(\alpha(t))^N}{N!S_N(\alpha(t))} = E_N(\alpha(t)) \end{aligned} \tag{3.3.2}$$

where

$$S_N(\alpha(t)) = \sum_{k=0}^{N} \frac{(\alpha(t))^k}{k!}$$

Note also that

$$\sum_{k=0}^{N} p_k(t,N) = 1$$

The above equations (3.3.2) are analogous to the steady state probabilities, except that the offered traffic is now a time-dependent function, yet to be determined.

Then, using (3.3.1) and (3.3.2), we have

$$\begin{aligned} -\frac{dS_N(\alpha)}{d\alpha}\frac{\dot{\alpha}}{S_N^2(\alpha)} &= -\frac{\lambda}{S_N(\alpha)} + \frac{\mu\alpha}{S_N(\alpha)} \\ -\dot{\alpha} &= (-\lambda + \mu\alpha)\frac{S_N(\alpha)}{S_{N-1}(\alpha)} \end{aligned} \tag{3.3.3}$$

since $\frac{dS_N(\alpha)}{d\alpha} = S_{N-1}(\alpha)$.

Now

$$\begin{aligned} \frac{S_N(\alpha)}{S_{N-1}(\alpha)} &= \frac{\sum_{k=0}^{N}\frac{\alpha^k}{k!}}{\sum_{k=0}^{N-1}\frac{\alpha^k}{k!}} \\ &= \frac{\sum_{k=0}^{N}\frac{\alpha^k}{k!}}{\sum_{k=0}^{N}\frac{\alpha^k}{k!} - \frac{\alpha^N}{N!}} \\ &= \frac{1}{1 - E_N(\alpha(t))} \\ &= \frac{1}{1 - p_N(t, N)} \end{aligned} \tag{3.3.4}$$

from our assumption in (3.3.2).

Thus, using (3.3.4) in (3.3.3), we have, letting t_0 be our initial time,

$$\begin{aligned} -\dot{\alpha} &= \frac{-\lambda + \mu\alpha}{1 - p_N(t, N)} \\ \frac{\dot{\alpha}}{\lambda - \mu\alpha} &= \frac{1}{1 - p_N(t, N)} \\ log(\lambda - \mu\alpha)\Big|_{t_0}^{t} &= -\mu\int_{t_0}^{t}\frac{1}{1 - p_N(x, N)}dx \end{aligned}$$

$$
\begin{aligned}
\lambda-\mu\alpha(t) &= (\lambda-\mu\alpha(t_0))exp\left(-\mu\int_{t_0}^{t}\frac{1}{1-p_N(x,N)}dx\right)\\
\alpha(t) &= \frac{1}{\mu}\left(\lambda-(\lambda-\mu\alpha(t_0))exp\left(-\mu\int_{t_0}^{t}\frac{1}{1-p_N(x,N)}dx\right)\right)\\
&= \frac{\lambda}{\mu}\left(1-exp\left(-\mu\int_{t_0}^{t}\frac{1}{1-p_N(x,N)}dx\right)\right)\\
&\quad + \alpha(t_0)exp\left(-\mu\int_{t_0}^{t}\frac{1}{1-p_N(x,N)}dx\right) \qquad (3.3.5)
\end{aligned}
$$

Equation (3.3.5) gives the expression for $\alpha(t)$ based on the first birth-and-death equation. However, the problem with (3.3.5) is that the very measure we are interested in, $p_N(t,N)$, appears in the integral. Since we require $E_N(\alpha(t)) = p_N(t,N)$, this fact seems to be a setback, but there is a way to get around it, as will be shown in the next section.

Now, we come to the rationale behind the use of the first birth-and-death equation. Why not the second, the third, or the last? It is easy to show that, using the above approach, based on the $k^{th}(k = 2,3,...,N)$ birth-and-death equation, we get

$$\dot{\alpha} = \frac{\lambda(k-1)-(\lambda+(k-1)\mu)\alpha+\mu\alpha^2}{k-1-\alpha(1-p_N(t,N))} \qquad (3.3.6)$$

which is not an easy differential equation to solve. For the last birth-and-death equation, the $(N+1)^{th}$, the corresponding differential equation to solve is

$$\dot{\alpha} = \frac{N(\lambda-\mu\alpha)}{N-(1-p_N(t,N))\alpha} \qquad (3.3.7)$$

The above equation is solvable, but the resulting expression contains $\alpha(t)$ implicitly which is also not easy to solve.

So that leaves us with the first birth-and-death equation, and that is the equation which we have utilized.

3.4 Approximation of the Transient Blocking Probability Using the Modified Offered Traffic

We are considering the system as described in Section 2.2. Let us assume that traffic is held at a constant intensity of λ for $t > t_0$. We also normalize the mean service time, $\frac{1}{\mu}$, to one, and let $\alpha(t_0) = \alpha_0$ for $t = t_0$. Due to this normalization, time is now measured in units of the mean service time.

We are interested in the transient blocking probability of the system before it reaches the steady state. However, to use the above approximation, we have to divide the time $[t_0, t]$ into n intervals, each of length Δt. The smaller Δt is, the better our approximation becomes. To approximate the probability at time t, we first use (3.3.5) on $(t_0, t_0 + \Delta t]$, subject to $\alpha(t_0)$. This is to find $\alpha(t_0 + \Delta t)$. Then we work on $(t_0 + \Delta t, t_0 + 2\Delta t]$ to find $\alpha(t_0 + 2\Delta t)$, subject to $\alpha(t_0 + \Delta t)$. We proceed in the same manner until we reach the final interval, $(t_0 + (n-1)\Delta t, t_0 + n\Delta t]$ or $(t - \Delta t, t]$, which would depend on $\alpha(t_0 + (n-1)\Delta t)$.

On each interval we replace $p_N(t, N)$ in (3.3.5), since it is unknown, with $\bar{p}$, an estimate of $p_N(t, N)$ at t. Hence (3.3.5) becomes

$$\alpha(t_0 + k\Delta t) = \lambda(1 - \exp(-\frac{\Delta t}{1 - \bar{p}})) + \alpha(t_0 + (k-1)\Delta t)\exp(-\frac{\Delta t}{1 - \bar{p}}) \qquad (3.4.1)$$

where $k = 1, 2, ..., n$.

Using this initial estimate, $\bar{p}$, we calculate α and use $E_N(\alpha)$ to obtain a better $\bar{p}$. This new $\bar{p}$ will then be used in (3.4.1) and the whole procedure is repeated until $\bar{p}$ converges. When this occurs we take $\bar{p}$ as an approximation to $p_N(t, N)$. We therefore have an iterative procedure for each interval and this procedure converges very rapidly even when we use $E_N(\lambda)$ as our initial estimate to the very first interval.

Our algorithm is therefore

```
Step 0    Set convergence criterion, $\epsilon$, say.
          Set $\bar{p}_{old} = E_N(\lambda)$.
          Set $\alpha_{old} = \alpha(t_0)$.
          Divide $[t_0, t]$ into $n$ intervals, each of length $\Delta t$.
          Set $k = 1$.
Step 1    $\alpha(t_0 + (k-1)\Delta t) = \alpha_{old}$.
Step 2    On $(t_0 + (k-1)\Delta t, t_0 + k\Delta t]$, calculate
              $\alpha_{new} = \lambda(1 - \exp(-\frac{\Delta t}{1-\bar{p}_{old}})) + \alpha_{old}\exp(-\frac{\Delta t}{1-\bar{p}_{old}})$
          Calculate
              $\bar{p}_{new} = E_N(\alpha_{new})$
          If $\left|\bar{p}_{new} - \bar{p}_{old}\right| \leq \epsilon$ then go to Step 3.
          $\bar{p}_{old} = \bar{p}_{new}$
          Go to Step 2.
Step 3    If $k = n$ then go to Step 4.
          $\alpha_{old} = \alpha_{new}$.
          $k = k + 1$.
          Go to Step 1.
Step 4    $p_N(t, N) \approx \bar{p}_{new}$.
          STOP.
```

3.5 Other Approximations

There are two other approximations worth noting. They are due to Syski [24], Berry [2], and Jagerman [8]. The first is obtained by using the so-called time-dependent offered traffic (Syski [24], Berry [2]), as discussed in Section 1.4. In this case we have

$$p_N(t, N) \approx E_N(a(t)) \tag{3.5.1}$$

where

$$a(t) = \int_0^t \lambda(t - x) H^c(x) dx \tag{3.5.2}$$

Since in the M/M/N loss system λ is time-independent and $H(t) = 1 - e^{-t}$,letting $\mu = 1$, we have

$$\begin{aligned} a(t) &= \lambda \int_0^t e^{-x} dx \\ &= \lambda(1 - e^{-t}) \end{aligned} \tag{3.5.3}$$

We are again considering the system as described in Section 2.2. There is no iterative procedure involved apart from, calculating $E_N(a(t))$, using its recurrence relation. Equation (3.5.3) is a simple approximation and the approximations based on it will be discussed later.

The other approximation is due to Jagerman [8]. It is of the form

$$p_N(t, N) \approx E_N(\lambda) \left(1 - e^{-(\frac{\lambda}{N} + 1)t}\right)^N \tag{3.5.4}$$

where $\lambda = a$, since we normalize the service rate to one.

Another way of deriving approximation (3.5.4), which is much simpler, is to consider the system in Figure 3.5.1 where we have the M/M/N loss system.

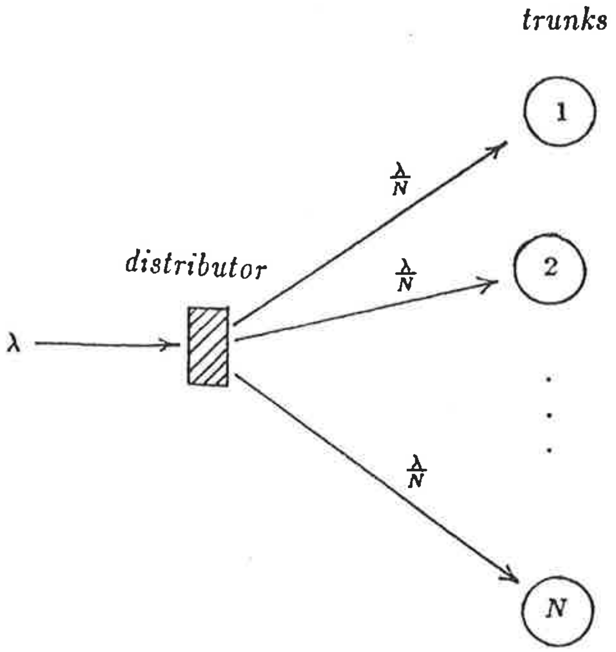

Figure 3.5.1 : A 'distributor' distributes traffic to the trunks.

Let us suppose that there is a kind of distributor that distributes the arriving calls evenly so that the intensity of the traffic to each trunk is $\frac{\lambda}{N}$. Now the exact transient blocking probability for an M/M/1 loss system with arrival intensity of $\frac{\lambda}{N}$ is

$$\tilde{p}_1(t,1) = E_1\left(\frac{\lambda}{N}\right)(1 - e^{-(\frac{\lambda}{N}+1)t}) \tag{3.5.5}$$

Equation (3.5.5) is obtained from solving the two birth-and-death equations for the system exactly. The tilde is used to denote that the arrival intensity is not λ but $\frac{\lambda}{N}$. Our assumption is that the *probability of all N trunks being busy at time t is proportional to the probability that every individual trunk, each with arrival intensity of $\frac{\lambda}{N}$, is busy at the same time.* Thus we have the approximation

$$p_N(t,N) \approx c\,(\tilde{p}_1(t,1))^N \tag{3.5.6}$$

where c is a scaling constant to be determined. We then let $t \to \infty$, that is to let the system reach the steady state, and (3.5.6) becomes

$$E_N(\lambda) \approx c\left(E_1\left(\frac{\lambda}{N}\right)\right)^N \tag{3.5.7}$$

since the Erlang Loss Function is the steady state blocking probability. This implies that

$$c \approx \frac{E_N(\lambda)}{\left(E_1\left(\frac{\lambda}{N}\right)\right)^N} \tag{3.5.8}$$

Inserting (3.5.8) in (3.5.6) gives

$$\begin{aligned} p_N(t,N) &\approx \frac{E_N(\lambda)}{\left(E_1\left(\frac{\lambda}{N}\right)\right)^N}(\tilde{p}_1(t,N))^N \\ &= E_N(\lambda)\left(\frac{\tilde{p}_1(t,1)}{E_1\left(\frac{\lambda}{N}\right)}\right)^N \\ &= E_N(\lambda)(1 - e^{-(\frac{\lambda}{N}+1)t})^N \end{aligned} \tag{3.5.9}$$

which is equation (3.5.4).

3.6 Comparison of Approximations

In this section we present tables of comparison of the above three approximations with simulation and numerical results, where available. They are Tables 3.6.1, 3.6.2 and 3.6.3. The numerical results are obtained using a numerical procedure to be outlined in Section 6.2. However the procedure is applicable for small N, that is for $N \leq 20$. Hence for larger N, no numerical results will be shown.

In the tables, Approximation 1 is obtained by using (3.5.3), Approximation 2 by (3.5.4) and Approximation 3 by (3.4.1). For Approximation 3, $\Delta t_k = 0.01$.

The simulation results are obtained from eight batches of runs, with each batch consisting of 10000 runs. The confidence limit is set at 0.1.

3.7 Discussion

The new approximation procedure, APP3, based on (3.4.1) compares favourably with the other approximations. However it seems that Jagerman's approximation, APP2, gives the best results. Therefore in the case of the M/M/N loss system, there does not appear to be much advantage in using (3.4.1) for approximations. In fact it is a bit slower than the others because of the use of the small time intervals. As expected, the smaller Δt_k is, the slower it becomes. But when we come to the case of the M(t)/M(t)/N loss system, this approximation is much better than the others, as will be shown in Chapter VI. Furthermore, it is easier to use than the approximation based on (3.5.2)

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	SIMULATION
0.5	0.0002	0.0001	-0.0001	0.0001	-0.0001	0.0001	-0.0001	(0.0001,0.0003)
1.0	0.0047	0.0038	-0.0009	0.0045	-0.0002	0.0038	-0.0009	(0.0069,0.0076)
1.5	0.0153	0.0127	-0.0026	0.0167	0.0014	0.0128	-0.0025	(0.0197,0.0210)
2.0	0.0256	0.0221	-0.0035	0.0285	0.0029	0.0224	-0.0032	(0.0275,0.0290)
2.5	0.0329	0.0295	-0.0034	0.0359	0.0030	0.0298	-0.0031	(0.0342,0.0364)
3.0	0.0373	0.0345	-0.0028	0.0397	0.0024	0.0349	-0.0024	(0.0365,0.0381)
3.5	0.0399	0.0378	-0.0021	0.0416	0.0017	0.0381	-0.0018	(0.0408,0.0418)
4.0	0.0413	0.0398	-0.0015	0.0424	0.0011	0.0401	-0.0012	(0.0368,0.0382)
4.5	0.0421	0.0411	-0.0010	0.0428	0.0007	0.0413	-0.0008	(0.0409,0.0432)
5.0	0.0426	0.0419	-0.0007	0.0430	0.0004	0.0421	-0.0005	(0.0431,0.0452)

Table 3.6.1 : $\lambda = 6.0$, $N = 10$, $E_{10}(6.0) = 0.0431$.

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	SIMULATION
0.5	0.0065	0.0048	-0.0017	0.0022	-0.0043	0.0048	-0.0017	(0.0101,0.0113)
1.0	0.0778	0.0535	-0.0243	0.0501	-0.0277	0.0553	-0.0225	(0.0963,0.0996)
1.5	0.1528	0.1113	-0.0415	0.1288	-0.0240	0.1174	-0.0354	(0.1552,0.1587)
2.0	0.1903	0.1514	-0.0389	0.1784	-0.0119	0.1600	-0.0303	(0.1881,0.1930)
2.5	0.2055	0.1764	-0.0291	0.2006	-0.0049	0.1849	-0.0206	(0.2019,0.2069)
3.0	0.2112	0.1915	-0.0107	0.2093	-0.0019	0.1986	-0.0126	(0.2098,0.2136)
3.5	0.2134	0.2006	-0.0128	0.2126	-0.0008	0.2061	-0.0073	(0.2095,0.2128)
4.0	0.2142	0.2061	-0.0081	0.2139	-0.0003	0.2101	-0.0041	(0.2087,0.2117)
4.5	0.2145	0.2095	-0.0050	0.2143	-0.0002	0.2122	-0.0023	(0.2096,0.2130)
5.0	0.2146	0.2115	-0.0031	0.2145	-0.0001	0.2133	-0.0013	(0.2091,0.2124)

Table 3.6.2 : $\lambda = 10.0$, $N = 10$, $E_{10}(10.0) = 0.2146$.

TIME	APP1	APP2	APP3	SIMULATION
0.5	0.0006	0.0000	0.0005	(0.0016,0.0023)
1.0	0.1111	0.0083	0.1187	(0.2799,0.2857)
1.5	0.2387	0.1414	0.2638	(0.3852,0.3945)
2.0	0.3065	0.2981	0.3354	(0.3910,0.3981)
2.5	0.3430	0.3646	0.3677	(0.3813,0.3914)
3.0	0.3635	0.3851	0.3819	(0.3855,0.3940)
3.5	0.3755	0.3909	0.3882	(0.3870,0.3960)
4.0	0.3825	0.3925	0.3909	(0.3869,0.3935)
4.5	0.3867	0.3929	0.3921	(0.3819,0.3959)
5.0	0.3892	0.3930	0.3926	(0.3870,0.3962)

Table 3.6.3 : $\lambda = 80.0$, $N = 50$, $E_{50}(80.0) = 0.3931$.

because (3.5.2) involves an integral. All these approximations will be applied to the time-dependent loss system and compared.

CHAPTER IV

APPROXIMATION OF THE M(t)/M(t)/N LOSS SYSTEM BY THE M/M/N LOSS SYSTEM

4.1 Introduction

In the previous chapters, the M/M/N loss system was considered. Exact and approximation methods to evaluate its blocking probability in the transient state are outlined. However, as mentioned in Chapter I, the M(t)/M(t)/N loss system is more relevant and accurate in dealing with real-life problems of teletraffic. This is because the arrival and service processes are not time-independent, but are time-dependent. Therefore it is worth considering the question of whether the M(t)/M(t)/N loss system can be approximated by the M/M/N loss system, so that methods and ideas developed for the M/M/N loss system can be extended to this time-dependent system. We would like to know if the methods for computing exact or approximate blocking probability in the M/M/N loss system can be used for the same purpose in the corresponding time-dependent system, the M(t)/M(t)/N loss system.

4.2 Why Approximate?

To solve the M(t)/M(t)/N loss system, we have to solve its associated birth-and-death equations, namely (1.5.1), which consist of $(N+1)$ coupled diferential equations with time-dependent coefficients. We can express these equations as a matrix differential equations, that is

$$\begin{aligned} \frac{d\underline{p}(t,N)}{dt} &= A(t)\underline{p}(t,N) \\ \underline{p}(0,N) &= \underline{p}_0 \end{aligned} \tag{4.2.1}$$

where $\underline{p}(t,N)$ and $A(t)$ are as given in equations (1.5.4).

Unfortunately, there is still no general procedure to solve (4.2.1) with this particular coefficient matrix, that is $A(t)$. Matrix differential equations with certain special forms of the coefficient matrix can, however, be solved (Wu [26]).

In the previous chapters, it has been shown that good approximations of the transient blocking probability of the M/M/N loss system can be made using simple procedures. An exact method to compute it is also given, but it is not suitable for large N. Since the M(t)/M(t)/N loss system is very difficult to solve, especially when N is large, it would be very useful if we can approximate the M(t)/M(t)/N loss system by the M/M/N loss system. It is also worth noting that Jagerman [8] considered the M(t)/M/N loss system and obtained an integral equation involving the time-dependent blocking probability, as will be discussed in the next chapter.

4.3 Approximating the M(t)/M(t)/N Loss System

The only difference between the M(t)/M(t)/N and the M/M/N loss systems is the time-dependency of both the arrival and service processes. In fact the M/M/N loss system is a special case of the M(t)/M(t)/N loss system. Therefore it is logical that we work on the arrival and service rate functions in order to derive any approximation method. This can be accomplished by approximating both the arrival and service rate functions by a series of step-functions. Miwa [15] suggested this approach for the time-varying arrival rate function. It is the simplest way of approximating functions, continuous or discrete, but its application in the approximation of the M(t)/M(t)/N loss system by the M/M/N loss system has never been considered before.

In Figure 4.3.1, the continuous function, $f(t)$, is approximated by a series of step-functions, with each step-function corresponding to a time-interval of length Δt. In the case of arrival and service rates with known continuous functions, it is easy to approximate them with the step-functions. The smaller Δt is, the more accurate our approximations should be.

However, for the case when $f(t)$ consists of discrete points, as would be obtained from real-life monitoring of the arrival and service processes, the problem of approximation is more involved (see Figure 4.3.2). In this case, we have two alternatives.

For the first alternative, we start off by assuming that there exists an underlying continuous function that results in these discrete points. We can reproduce this function by drawing a curve of best fit. However, we need to collect enough data to justify the curve. Then we just approximate this curve with step-functions as described before (see Figure 4.3.3). In this case we can take Δt to be small as we please.

For the other alternative, we simply approximate the discrete points with step-functions directly as shown in Figure 4.3.4. This procedure is very straightforward but the choice of Δt has to be made with caution. If it is too small, then we will not have enough data in such a small interval, and our approximation will not be justified (lack of statistics). On the other hand, if Δt is too big, then our approximation is too rough and becomes less accurate because the steady state could be achieved.

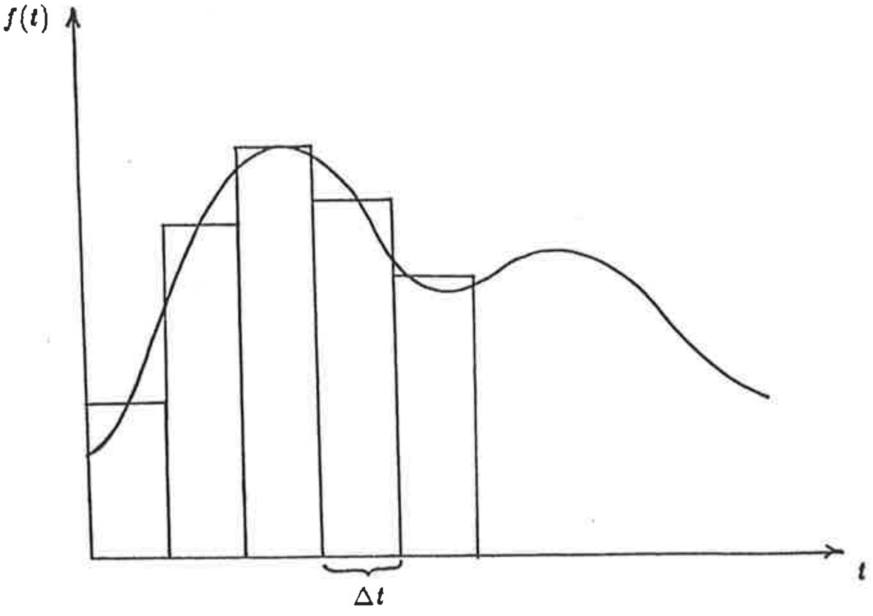

Figure 4.3.1 : Approximating f(t) with step-functions.

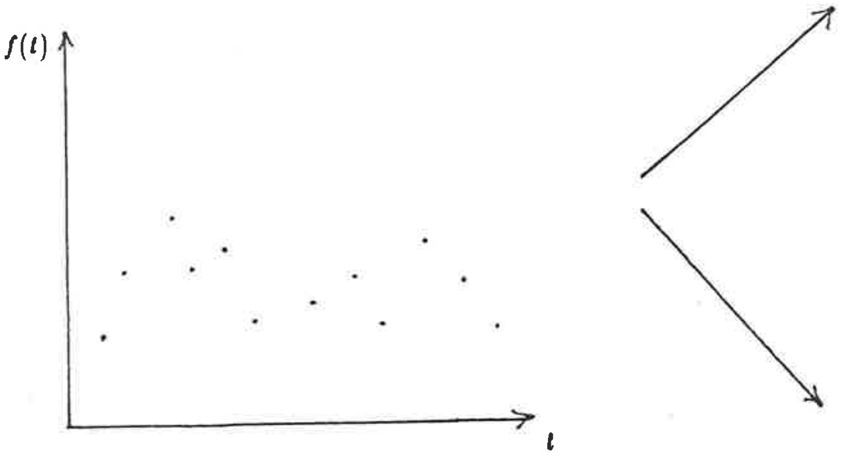

Figure 4.3.2 : A collection of discrete points

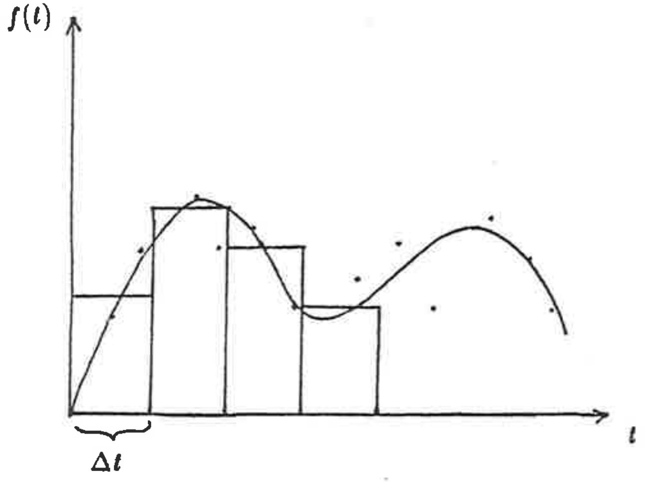

Figure 4.3.3 : Approximating the discrete points with a curve. The curve is then approximated with step-functions.

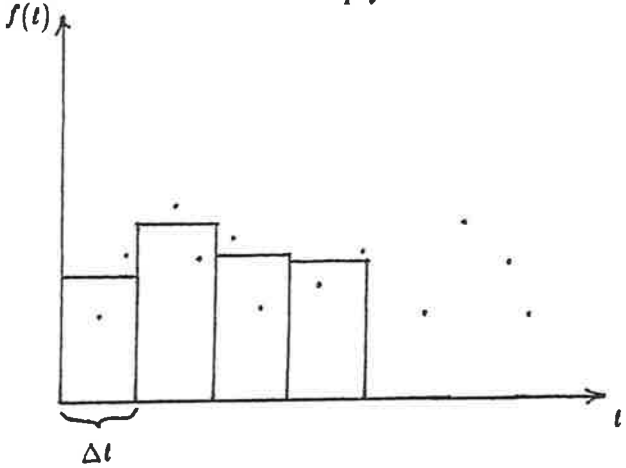

Figure 4.3.4 : Approximating the discrete points directly with step-functions.

This will defeat our purpose of trying to find the time-dependent probability as accurately as possible. Therefore what we require is the optimal Δt. Studies by Martikainen [11] shows that the optimal Δt seems to be in the vicinity of one quarter-hour (15 minutes), that is with $t \approx 5$ when we normalize the mean service rate and let the mean holding (service) time be about three minutes.

Once we have discretized our functions, $\lambda(t)$ and $\mu(t)$, into step-functions, we are dealing with a succession of the M/M/N loss systems in place of the M(t)/M(t)/N loss system. This is to say that the M(t)/M(t)/N loss system is now being approximated by a succession of M/M/N loss systems in the intervals $(0, \Delta t], (\Delta t, 2\Delta t], (2\Delta t, 3\Delta t], \ldots$. In each of these intervals of length Δt, an M/M/N loss system would be operating, and we can use methods developed earlier for this system to approximate the time-dependent blocking probability, subject to certain initial conditions which are due to the system in the preceding interval. This will be illustrated in the succeeding chapters.

4.4 Discussion

The subject of approximating the M(t)/M(t)/N loss system with the M/M/N loss system was discussed in the previous section. This concept is critical in the development of the succeeding chapters. It may sound very simple, but the results obtained using it are very encouraging, as will be shown. It can be applied for any type of arrival and service rate functions, discrete and continuous. In the next two chapters we will treat the simple M(t)/M(t)/1 loss system followed by the more general M(t)/M(t)/N loss system. The above idea will be applied in both cases.

CHAPTER V

TIME-DEPENDENT BLOCKING PROBABILITY FOR THE M(t)/M(t)/1 LOSS SYSTEM

5.1 Introduction

As mentioned before, both the arrival and service processes of teletraffic systems are found to be nonstationary or time-dependent. This is evident from direct monitoring of real-life situations (Martikainen and Lehtinen [12]). Nevertheless, at present, models developed for traffic with constant arrival rates and based on constant holding time are used even when both the traffic and the holding time vary with time. As expected, the results are highly unsatisfactory when both of these quantities are highly time-dependent. Therefore it is desirable that a model be developed that will enable us to predict the time-dependent blocking probability with the assumption that both the arrival and service rates are time-dependent or nonstationary. However, as discussed before, this is not easy due to the difficulty in solving a set of coupled first order ordinary differential equations with time-dependent coefficient functions.

5.2 Jagerman's Integral Equation

This problem of determining the blocking probability at time t, $p_N(t, N)$, has been considered by Jagerman [8]. He derives an integral equation for $p_N(t, N)$ which involves the time-dependent offered traffic function, $a(t)$. It is of the form

$$p_N(t, N) = \beta_N(t, \infty) - \int_0^t K_N(t, x) p_N(x, N) dx \qquad (5.2.1)$$

where

$$\beta_N(t,\infty) = \sum_{j=0}^{N} \beta_{N-j}(0,N)\exp(-(N-j)t)\frac{(\int_0^t exp(s-t)a(s)ds)^j}{j!}$$

$$K_N(t,x) = a(x)\exp(-N(t-x))L_{N-1}^{(1)}\left(-\int_x^t exp(s-x)a(s)ds\right)$$

$L_{N-1}^{(1)}(.)$ is the Laguerre polynomial and is defined by

$$L_m^{(k)}(-x) = \sum_{j=0}^{m} \frac{x^j}{j!}\binom{m+k}{m-j}$$

The above equation (5.2.1) is not suitable for practical applications in real-life situations because of several factors. Firstly, it has a rather complicated form which could make numerical computation difficult. This is especially so when N is big and in real-life cases N could be as high as a few hundred. Secondly, it is restricted to continuous, integrable arrival rate functions, which is not necessarily so in real-life situations. Finally, it assumes the service rate to be constant, which further restricts its application. Despite these shortcomings, the integral equation still represents a major step in the study of time-dependent blocking in teletraffic systems. Several useful results are obtained from it, including the approximation method described in Section 3.5 (Jagerman [8]).

The procedure which follows uses the approach developed in the previous chapter for the case of the M(t)/M(t)/1 loss system. It will involve solving matrix differential equations with time-dependent coefficient matrices. This, in turn, will involve computation of matrix exponentials.

5.3 Method Formulation

For the M(t)/M(t)/1 loss system, the associated birth-and-death equations are

$$\begin{aligned}\frac{dp_0(t,1)}{dt} &= -\lambda(t)p_0(t,1)+\mu(t)p_1(t,1)\\ \frac{dp_1(t,1)}{dt} &= \lambda(t)p_0(t,1)-\mu(t)p_1(t,1)\end{aligned} \tag{5.3.1}$$

with the usual notations. In matrix form the equations (5.3.1) is

$$\frac{d\underline{p}(t,1)}{dt} = A(t)\underline{p}(t,1) \tag{5.3.2}$$

where

$$\underline{p}(t,1) = \begin{pmatrix} p_0(t,1) & p_1(t,1)\end{pmatrix}^T$$

$$A(t) = \begin{pmatrix} -\lambda(t) & \mu(t) \\ \lambda(t) & -\mu(t)\end{pmatrix}$$

Equation (5.3.2) must be solved to obtain the required probability, but as discussed before, this is presently not possible due to the form of $A(t)$.

Let us now assume that calls are time-dependent and arrive to this one-trunk system according to the arrival pattern in Figure 5.3.1, which consists of three step-functions of constant traffic. It is described by

$$\lambda(t) = \begin{cases} \lambda_1, & t \in (0,t_1] \\ \lambda_2, & t \in (t_1,t_2] \\ \lambda_3, & t \in (t_2,t_3] \\ 0, & \text{elsewhere}\end{cases} \tag{5.3.3}$$

Let us assume that the service rate is also time-dependent and obeys the service pattern shown in Figure 5.3.2. It is described by

$$\mu(t) = \begin{cases} \mu_1, & t \in (0,t_1] \\ \mu_2, & t \in (t_1,t_2] \\ \mu_3, & t \in (t_2,t_3] \\ 0, & \text{elsewhere}\end{cases} \tag{5.3.4}$$

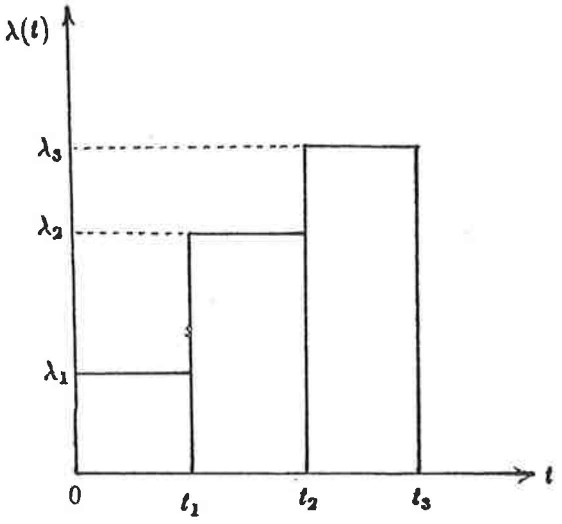

Figure 5.3.1 : A sample arrival rate.

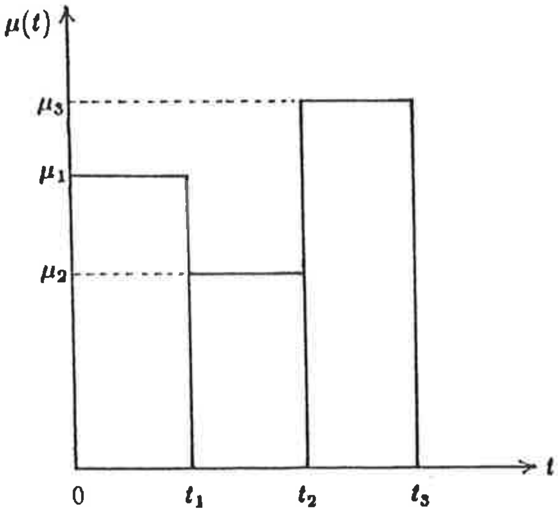

Figure 5.3.2 : A sample service rate.

These patterns are chosen for simplicity. If the λ's and the μ's are different for the same time-interval, we can partition the time-interval and redefine $\lambda(t)$ and $\mu(t)$. This is easily done.

Now consider the first interval $(0, t_1]$. In this interval, the arrival rate is λ_1 and the corresponding service rate is μ_1, that is they are both constants. The blocking probability at $t \in (0, t_1]$ can be found by solving

$$\frac{d\underline{p}(t,1)}{dt} = A_1 \underline{p}(t,1) \tag{5.3.5}$$

where

$$\underline{p}(t,1) = \begin{pmatrix} p_0(t,1) & p_1(t,1) \end{pmatrix}^T$$

$$A_1 = \begin{pmatrix} -\lambda_1 & \mu_1 \\ \lambda_1 & -\mu_1 \end{pmatrix}$$

The coefficient matrix to equation (5.3.2) is now a constant matrix, and thus (5.3.2) can be solved, which gives

$$\underline{p}(t,1) = \exp(A_1 t)\underline{p}(0,1) \tag{5.3.6}$$

The exponential of a matrix A is defined by

$$\exp(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

Its properties are described in numerous texts on matrix analysis, see for example, Bellman [1]. Several methods exist to evaluate $\exp(A)$, both exact and numerical (Moler and Van Loan [16], Bellman [1]).

Therefore at $t = t_1$ the solution is

$$\underline{p}(t_1,1) = \exp(A_1 t_1)\underline{p}(0,1) \tag{5.3.7}$$

where

$$\underline{p}(t_1,1) = \begin{pmatrix} p_0(t_1,1) & p_1(t_1,1) \end{pmatrix}^T$$

and $\underline{p}(0,1)$ is the initial probability vector to the first step-function of traffic. It could be set as required as it represents the probability distribution at $t = 0$.

For the second step-function of traffic in the interval $(t_1, t_2]$, equation (5.3.2) becomes

$$\frac{d\underline{p}(t,1)}{dt} = A_2\underline{p}(t,1) \tag{5.3.8}$$

where

$$A_2 = \begin{pmatrix} -\lambda_2 & \mu_2 \\ \lambda_2 & -\mu_2 \end{pmatrix}$$

We then have at $t = t_2$,

$$\underline{p}(t_2,1) = \exp(A_2(t_2 - t_1))\underline{p}(t_1,1) \tag{5.3.9}$$

where $\underline{p}(t_1,1)$, which was obtained earlier, is now the initial probability vector to this second step-function of traffic.

Likewise, for $t \in (t_2, t_3]$, we have at $t = t_3$,

$$\underline{p}(t_3,1) = \exp(A_3(t_3 - t_2))\underline{p}(t_2,1) \tag{5.3.10}$$

where $\underline{p}(t_2,1)$ is now the corresponding initial probability vector.

From equations (5.3.7), (5.3.9) and (5.3.10), we have

$$\begin{aligned} \underline{p}(t_1,1) &= \exp(A_1\Delta t_1)\underline{p}(0,1) \\ \underline{p}(t_2,1) &= \exp(A_2\Delta t_2)\underline{p}(t_1,1) \\ \underline{p}(t_3,1) &= \exp(A_3\Delta t_3)\underline{p}(t_2,1) \end{aligned} \tag{5.3.11}$$

where $\Delta t_k = t_k - t_{k-1}, k = 1, 2, 3$, and $t_0 = 0$.

Thus we can evaluate the blocking probability at any time-point by choosing the appropriate Δt_k. For example, still referring to Figures 5.3.1 and 5.3.2 with $\Delta t_k = 1$, to find the blocking probability at $t = 2.7$, say, we would have

$$\underline{p}(2.7, 1) = \exp(0.7A_3)\exp(A_2)\exp(A_3)\underline{p}(0, 1) \tag{5.3.12}$$

We can easily generalize the above formulation to the case when the arrival and service rates consist of m step-functions, each of duration one, and $\Delta t_k = 1$. Then we have

$$\begin{aligned} \frac{d\underline{p}(m,1)}{dt} &= A_m \underline{p}(m,1) \\ \Rightarrow \underline{p}(m,1) &= \exp(A_m)\underline{p}(m-1,1) \\ &= \exp(A_m)\exp(A_{m-1})\underline{p}(m-2,1) \\ &= \ldots \\ &= \left(\prod_{k=0}^{m-1} \exp(A_{m-k})\right)\underline{p}(0,1) \end{aligned} \tag{5.3.13}$$

The above expressions are equivalent to starting off from an initial state and keeping λ and μ constant for the appropriate interval. In fact they give the general solution for $\underline{p}(t, 1)$ with constant λ and μ in each interval. In each interval a different M/M/N loss system is operating and the blocking experienced by any one of them influences the succeeding system *through the initial probability vector in the succeeding interval.*

The only problem left is to evaluate the matrix exponentials. Unfortunately, we find that

$$\prod_{k=0}^{m-1} \exp(A_{m-k}) \neq \exp\left(\sum_{k=0}^{m-1} A_{m-k}\right) \tag{5.3.14}$$

since the matrices $\boldsymbol{A}_k$ do not commute (Bellman [1]). However, the above expression, (5.3.13), can be evaluated by a computer quickly and efficiently. The matrix exponentials can be evaluated in closed form by using, for example, the Lagrange Interpolation Method to express $\exp(\boldsymbol{A}_k \Delta t_k)$ as a matrix. Other methods to achieve this are described by Moler and Van Loan [16] and Bellman [1].

Working on $\boldsymbol{A}_k$, where

$$\boldsymbol{A}_k = \begin{pmatrix} -\lambda_k & \mu_k \\ \lambda_k & -\mu_k \end{pmatrix}$$

we have

$$\left| \boldsymbol{A}_k - \boldsymbol{I}s \right| = 0$$

$$\Rightarrow s_1 = 0$$

$$s_2 = -\lambda_k - \mu_k$$

where $\boldsymbol{I}$ is the identity matrix, and s_1 and s_2 are the eigenvalues of $\boldsymbol{A}_k$.

Using the abovementioned method, we have

$$\exp(\boldsymbol{A}_k \Delta t_k) = \frac{1}{\lambda_k + \mu_k} \begin{pmatrix} \mu_k + \lambda_k x_k & \mu_k(1 - x_k) \\ \lambda_k(1 - x_k) & \lambda_k + \mu_k x_k \end{pmatrix} \qquad (5.3.15)$$

where

$$x_k = \exp(-\Delta t_k(\lambda_k + \mu_k))$$

The above method can be extended to higher order matrices which correspond to multiple-trunk systems, although the algebra would become tedious. For these cases other methods could be more feasible. In the next chapter we treat these multiple-trunk systems, that is the M(t)/M(t)/N loss systems. Nevertheless, once we obtain a matrix like in (5.3.15), the computation of the blocking probability is straightforward.

5.4 Application to a Particular Discrete Traffic Pattern

Let us again consider the the birth-and-death equations for the M/M/1 loss system because they are relevant to our work. They are

$$\begin{aligned}\frac{dp_0(t,1)}{dt} &= -\lambda p_0(t,1) + \mu p_1(t,1)\\ \frac{dp_1(t,1)}{dt} &= \lambda p_0(t,1) - \mu p_1(t,1)\end{aligned} \qquad (5.4.1)$$

Now we rescale the time-axis by introducing a new variable

$$\tau = \mu t \qquad (5.4.2)$$

Then (5.4.1) will be transformed into

$$\begin{aligned}\frac{dp_0(\tau,1)}{d\tau} &= -ap_0(\tau,1) + p_1(\tau,1)\\ \frac{dp_1(\tau,1)}{d\tau} &= ap_0(\tau,1) - p_1(\tau,1)\end{aligned} \qquad (5.4.3)$$

by dividing them with μ and letting a, now the offered traffic, be $a = \frac{\lambda}{\mu}$. Therefore, instead of solving (5.4.1), we can solve (5.4.3) which has a simpler form, that is, instead of considering the arrival and service rates defined by (5.3.3) and (5.3.4) respectively, we just consider the offered traffic rates defined by

$$a(\tau) = \begin{cases} a_1, & \tau \in (0, \tau_1] \\ a_2, & \tau \in (\tau_1, \tau_2] \\ a_3, & \tau \in (\tau_2, \tau_3] \\ 0, & \text{elsewhere} \end{cases} \qquad (5.4.4)$$

where $a_k = \frac{\lambda_k}{\mu_k}$, $k = 1, 2, 3$. This simple transformation only changes the time-axis. Supposing that $\frac{1}{\mu} = 3$ minutes. Then $\tau = 2$ is the same as $t = 6$ minutes. Therefore we have to be careful when we refer to the time. Otherwise, the system is not altered. We now work with the offered traffic rate instead of the arrival and service rates.

As an illustration of the technique discussed in the previous section, consider the offered traffic pattern as shown in Figure 5.4.1. Here we have discrete offered traffic

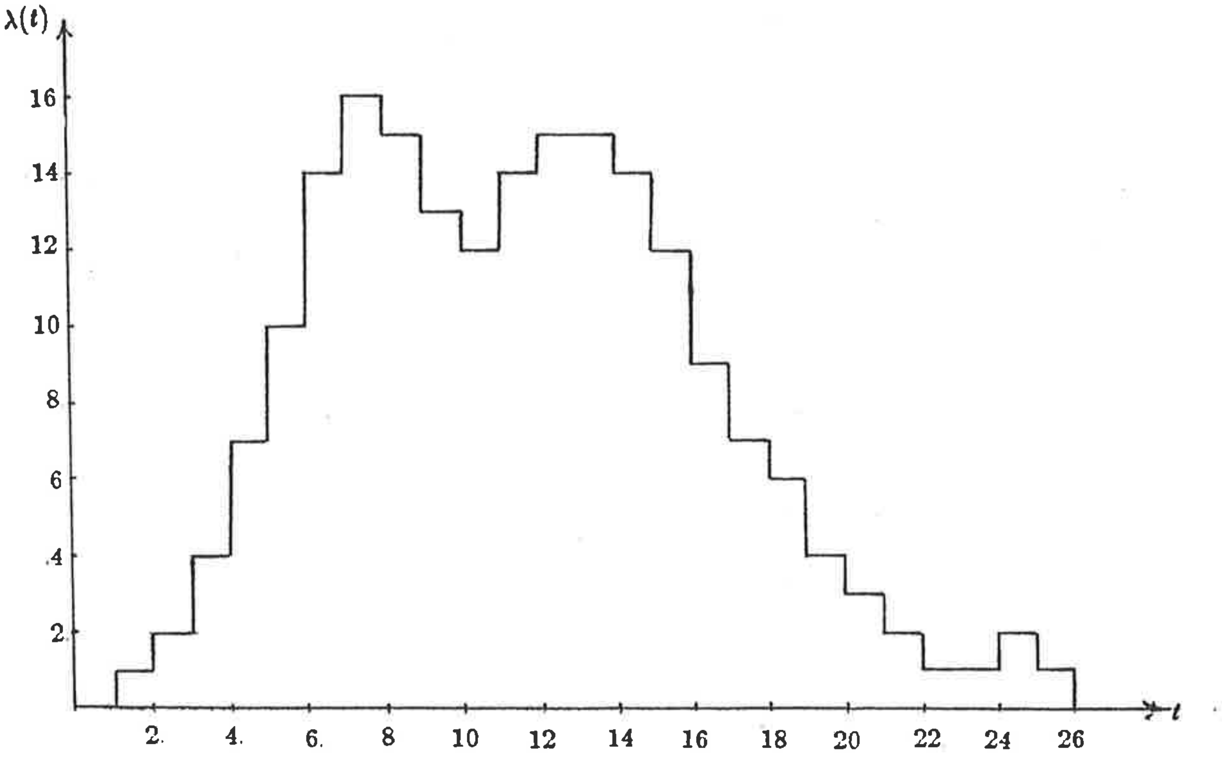

Figure 5.4.1 : A discrete traffic pattern

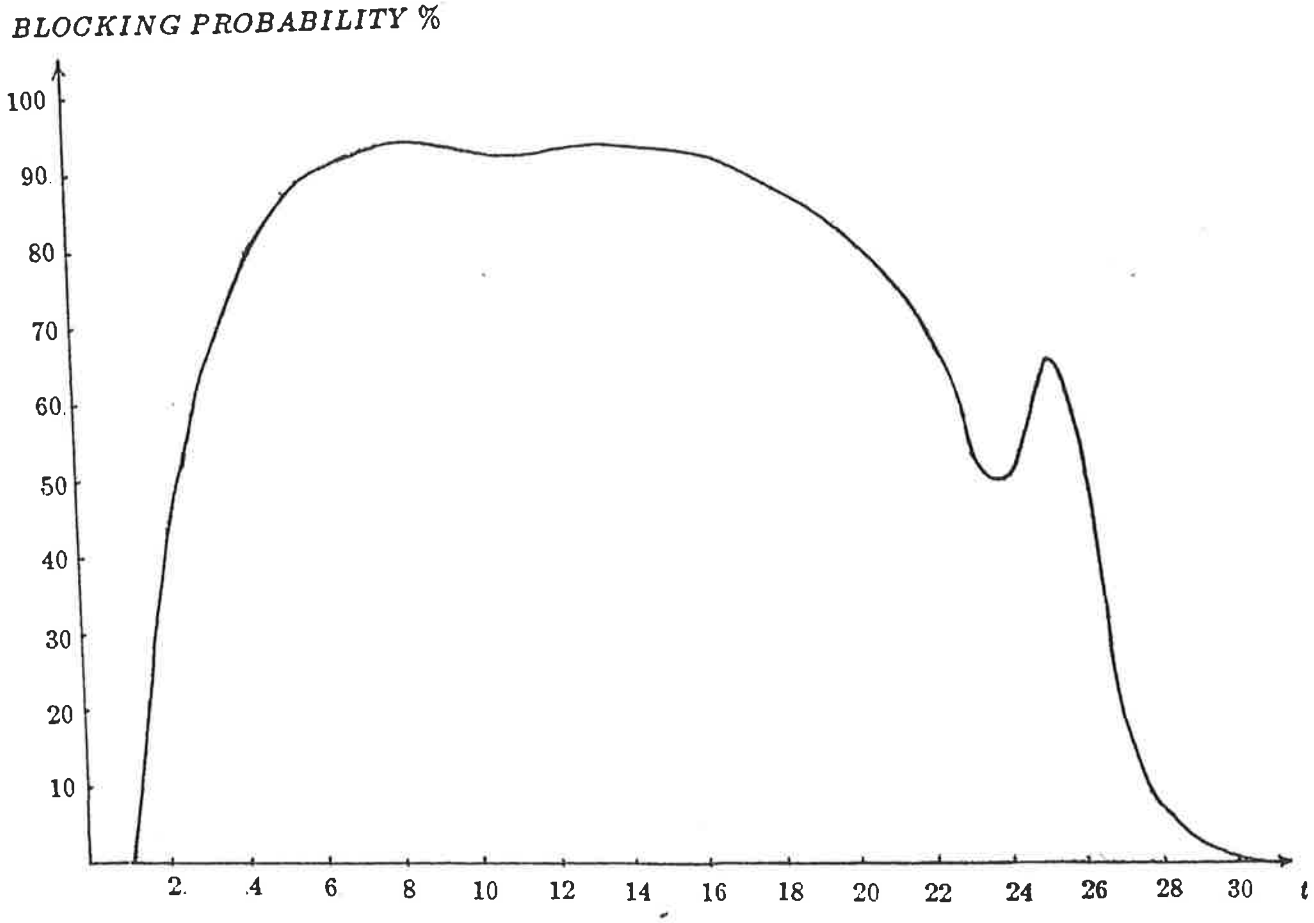

Figure 5.4.2 : Blocking probability for traffic as in Figure 5.4.1.

rates and the time interval is taken to be unity, that is, $\Delta\tau_k = 1$ for $k = 1, 2, \ldots$, for simplicity. The height of the step-function in every interval is actually the ratio of the constant arrival rate to the constant service rate in the same interval. We have to be careful with the time as the time-axis has now been rescaled as described above. Its actual length in minutes would depend on the individual mean service times, if they are measured in minutes, in each interval according to the transformation (5.4.2). The procedure was implemented on a VAX 11/780 computer and we obtained the blocking probability function as illustrated in Figure 5.4.2. At $\tau = 0$ we let

$$\underline{p}(0,1) = (1 \quad 0)^T$$

that is, the single trunk was initially free. Of course, we can set $\underline{p}(0,1)$ as may be required.

As can be seen from the graph, the blocking probability tends to follow the offered traffic as expected. As the offered traffic ends abruptly at $\tau = 26$ the blocking probability 'lingers on' and decays to zero due to the presence of calls still in service after that time.

5.5 Extension to a Continuous Traffic Pattern

The next logical step would be to adapt the procedure to the case when both the arrival and service rates are continuous functions of time. The associated birth-and-death equations are (5.3.1). Using a transformation similar to (5.4.2), that is,

$$\tau = \int_0^t \mu(x)dx \qquad (5.5.1)$$

equations (5.3.1) are transformed to

$$\begin{aligned}\frac{dp_0(\tau,1)}{d\tau} &= -\rho(\tau)p_0(\tau,1) + p_1(\tau,1)\\ \frac{dp_1(\tau,1)}{d\tau} &= \rho(\tau)p_0(\tau,1) - p_1(\tau,1)\end{aligned} \tag{5.5.2}$$

where $\rho(\tau) = \frac{\lambda(\tau)}{\mu(\tau)}$, assuming μ is never zero. Again we will have to be careful with the time as it has been transformed according to (5.5.1), and would depend on μ.

We now discretize the continuous function, $\rho(t)$, by step-functions as described in Chapter IV, and then apply the procedure oulined in Section 5.3. As an example, and for simplicity, let us assume that

$$\rho(\tau) = \tau \tag{5.5.3}$$

as shown in Figure 5.5.1. The results are set out in Table 5.5.1 for $\Delta\tau_k$ taking the values $0.1, 0.01, 0.001, 0.0001$ respectively, with the single trunk being free at $\tau = 0$.

As a comparison we also compute the exact blocking probability for the above function (5.5.3), using

$$\begin{aligned}\frac{dp_1(\tau,1)}{d\tau} &= \rho(\tau)p_0(\tau,1) - p_1(\tau,1)\\ p_0(\tau,1) &= 1 - p_1(\tau,1)\end{aligned} \tag{5.5.4}$$

From (5.5.4) we obtain

$$\frac{dp_1(\tau,1)}{d\tau} = \rho(\tau)(1 - p_1(\tau,1)) - p_1(\tau,1)$$

$$\Rightarrow \frac{dp(\tau,1)}{d\tau} + (\rho(\tau)+1)p_1(\tau,1) = \rho(\tau) \tag{5.5.5}$$

The integrating factor to (5.5.5) is

$$\exp\left(\int_0^\tau (\rho(x)+1)dx\right)$$

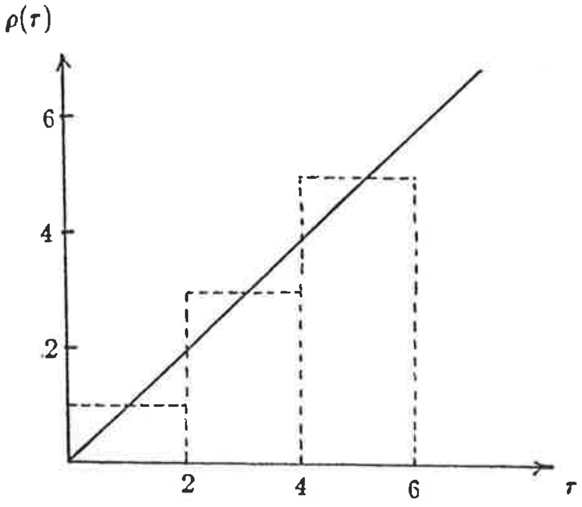

Figure 5.5.1 : $\rho(\tau) = \tau$.

TIME	EXACT	0.1	ERROR	0.01	ERROR	0.001	ERROR	0.0001	ERROR
0.5	0.1005	0.0826	-0.0179	0.1023	0.0018	0.1007	0.0002	0.1006	0.0001
1.0	0.2986	0.2776	-0.0210	0.3006	0.0020	0.2988	0.0002	0.2986	0.0000
1.5	0.4782	0.4622	-0.0160	0.4797	0.0015	0.4783	0.0001	0.4782	0.0000
2.0	0.6018	0.5915	-0.0103	0.6028	0.0010	0.6019	0.0001	0.6018	0.0000
2.5	0.6794	0.6728	-0.0066	0.6800	0.0004	0.6794	0.0000	0.6794	0.0000
3.0	0.7295	0.7250	-0.0045	0.7299	0.0004	0.7295	0.0000	0.7295	0.0000
3.5	0.7645	0.7612	-0.0033	0.7648	0.0003	0.7645	0.0000	0.7645	0.0000
4.0	0.7908	0.7882	-0.0026	0.7910	0.0002	0.7908	0.0000	0.7908	0.0000
4.5	0.8114	0.8094	-0.0020	0.8116	0.0002	0.8115	0.0001	0.8114	0.0000
5.0	0.8282	0.8266	-0.0016	0.8284	0.0002	0.8283	0.0001	0.8283	0.0001

Table 5.5.1 : Blocking probability for the above traffic with $\Delta\tau_k = 0.1, 0.01, 0.001, 0.0001$.

and hence

$$p_1(u,1)\exp\left(\int_0^u(\rho(x)+1)dx\right)\bigg|_0^\tau = \int_0^\tau \rho(u)\exp\left(\int_0^u(\rho(x)+1)dx\right)du$$

$$\begin{aligned}\Rightarrow p_1(\tau,1) = &\exp\left(-\int_0^\tau(\rho(x)+1)dx\right)\times \\ &\left(\int_0^\tau \rho(u)\exp\left(\int_0^u(\rho(x)+1)dx\right)du + p_1(0,1)\right)\end{aligned} \tag{5.5.6}$$

The exact blocking probability for $\rho(\tau)=\tau$ can be obtained by using the function in equation (5.5.6). We compare this exact probability with the probability obtained using the discrete procedure in Table 5.5.1. As expected the discrete method gives a good approximation to the continuous offered traffic function. By choosing a smaller time interval we are able to obtain a better approximation. For the case when $\Delta t_k = 0.0001$, there are 50000×2^2 multiplications, and even in this case the error propagated is minimal. We must also remember that this is for the one trunk system.

5.6 Discussion

The discrete method outlined above has been applied to one discrete and one continuous offered traffic case. Only results obtained for the continuous case for one trunk could be verified as an exact expression for its blocking probability exists.

The main weakness of this method lies in the evaluation of the matrix exponentials. This can be done rather easily and exactly for very small systems. However, for larger systems which are encountered in real-life, numerical methods have to be used to evaluate them. For the above example the error does not seem to be too serious. This is aided by the fact that the expression for the matrix exponentials is exact. A numerical method would have to be developed to compute these matrix exponentials which would exploit the properties of our special, tridiagonal coefficient matrix, $\boldsymbol{A_k}$, in the case of large systems. In the next chapter, when we consider the M(t)/M(t)/N loss system, a numerical method will be given which computes the matrix exponentials numerically using the so-called implicit QL algorithm.

CHAPTER VI

A NUMERICAL METHOD TO EVALUATE TIME-DEPENDENT BLOCKING PROBABILITY FOR THE M(t)/M(t)/N LOSS SYSTEM

6.1 Introduction

In this chapter we present a numerical procedure to evaluate the blocking probability for the M(t)/M(t)/N loss system. This loss system is described by a set of birth-and-death equations, namely (1.5.1). There is no method yet to solve them exactly, and so we have to resort to numerical and approximate methods.

On rescaling the time axis by using the transformation

$$\tau = \int_0^t \mu(x)dx$$

we can replace (1.5.1) by

$$\begin{aligned}
\frac{dp_0(\tau,N)}{d\tau} &= -\rho(\tau)p_0(\tau,N) + p_1(\tau,N) \\
\frac{dp_i(\tau,N)}{d\tau} &= \rho(\tau)p_{i-1}(\tau,N) - (\rho(\tau)+i)\,p_i(\tau,N) + (i+1)\,p_{i+1}(\tau,N) \\
& \qquad 1 \leq i \leq N-1 \\
\frac{dp_N(\tau,N)}{d\tau} &= \rho(\tau)p_{N-1}(\tau,N) - Np_N(\tau,N)
\end{aligned} \tag{6.1.1}$$

where $\rho(\tau) = \frac{\lambda(\tau)}{\mu(\tau)}$. In matrix form, (6.1.1) is

$$\frac{d\underline{p}(\tau,N)}{d\tau} = A(\tau)\underline{p}(\tau,N) \tag{6.1.2}$$

where

$$\underline{p}(\tau, N) = (\, p_0(\tau, N) \quad p_1(\tau, N) \quad \ldots \quad p_N(\tau, N)\,)^T$$

$$A = \begin{pmatrix} -\rho & 1 & 0 & \ldots & 0 & 0 & 0 \\ \rho & -\rho-1 & 2 & \ldots & 0 & 0 & 0 \\ 0 & \rho & -\rho-2 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & -\rho-(N-2) & N-1 & 0 \\ 0 & 0 & 0 & \ldots & \rho & -\rho-(N-1) & N \\ 0 & 0 & 0 & \ldots & 0 & \rho & -N \end{pmatrix} \tag{6.1.3}$$

where $\rho = \rho(\tau)$.

The approximate method described in Chapters IV and V is applied to (6.1.2) and we obtain the blocking probability vector as a function of matrix exponentials as in (5.3.13). Our next task is to evaluate these matrix exponentials to find the blocking probability. The numerical scheme to accomplish this is called the implicit QL algorithm (Martin and Wilkinson [13]).

6.2 The Numerical Procedure

We now discretize $\rho(\tau)$ such that

$$\rho_k = \rho(k\Delta\tau)$$

Then define the matrix

$$A_k = \begin{pmatrix} -\rho_k & 1 & 0 & \ldots & 0 & 0 & 0 \\ \rho_k & -\rho_k-1 & 2 & \ldots & 0 & 0 & 0 \\ 0 & \rho_k & -\rho_k-2 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & -\rho_k-(N-2) & N-1 & 0 \\ 0 & 0 & 0 & \ldots & \rho_k & -\rho_k-(N-1) & N \\ 0 & 0 & 0 & \ldots & 0 & \rho_k & -N \end{pmatrix} \tag{6.2.1}$$

Let us denote the matrix A_k from equation (6.2.1) by just A. For this type of matrix there is a fast, stable method for calculating $\exp(A\Delta\tau)$, which gives accurate results. This method applies the implicit QL algorithm described by Martin and Wilkinson [13]. It is an algorithm for determining the exponential of a symmetric, tridiagonal matrix. Thus, if we can find a diagonal matrix D, and a symmetric, tridiagonal matrix S, such that

$$A = D^{-1}SD \tag{6.2.2}$$

then

$$\exp(A) = D^{-1}\exp(S)D \tag{6.2.3}$$

where $\exp(S)$ can be calculated using the implicit QL algorithm.

Matrices D and S are obtained by solving the system $DA = SD$. Let

$$D = \begin{pmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ 0 & 0 & d_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & d_{N+1} \end{pmatrix}$$

$$S = \begin{pmatrix} s_{11} & s_{12} & 0 & \dots & 0 & 0 \\ s_{12} & s_{22} & s_{23} & \dots & 0 & 0 \\ 0 & s_{23} & s_{33} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & s_{N,N} & s_{N,N+1} \\ 0 & 0 & 0 & \dots & s_{N,N+1} & s_{N+1,N+1} \end{pmatrix}$$

Then

$$DA = \begin{pmatrix} -\rho d_1 & d_1 & 0 & \dots & 0 & 0 \\ \rho d_2 & -(\rho+1)d_2 & 2d_2 & \dots & 0 & 0 \\ 0 & \rho d_3 & -(\rho+2)d_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -(\rho+N-1)d_N & Nd_N \\ 0 & 0 & 0 & \dots & \rho d_{N+1} & -Nd_{N+1} \end{pmatrix}$$

and

$$SD = \begin{pmatrix} d_1 s_{11} & d_2 s_{12} & 0 & \dots & 0 & 0 \\ d_1 s_{12} & d_2 s_{22} & d_3 s_{23} & \dots & 0 & 0 \\ 0 & d_2 s_{23} & d_3 s_{33} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & d_N s_{NN} & d_{N+1} s_{N,N+1} \\ 0 & 0 & 0 & \dots & d_N s_{N,N+1} & d_{N+1} s_{N+1,N+1} \end{pmatrix}$$

If one solution for this system is found, then possible solutions are available by multiplying D by a constant, c, say, and dividing S by c^2. We shall take d_1 to be one. Since

$$(-\rho - i + 1)d_i = d_i s_{ii} \qquad i < N$$

$$-N d_{N+1} = d_{N+1} s_{N+1,N+1}$$

then

$$-\rho - i - 1 = s_{ii} \qquad i < N$$

$$-N = s_{N+1,N+1}$$

and all diagonal elements of S can be found. Also

$$d_1 = d_2 s_{12}$$

$$d_1 s_{12} = \rho d_2$$

which implies that

$$d_2 = \frac{1}{s_{12}} = \frac{s_{12}}{\rho}$$

$$\Rightarrow \frac{s_{12}}{\rho} = \frac{1}{s_{12}}$$

$$\Rightarrow s_{12} = \sqrt{\rho}$$

$$d_2 = \frac{1}{\sqrt{\rho}}$$

We also have

$$2d_2 = d_3 s_{23}$$

$$\rho d_3 = d_2 s_{23}$$

$$\Rightarrow d_3 = \frac{2\left(\frac{1}{\sqrt{\rho}}\right)}{s_{23}} = \frac{s_{23}\left(\frac{1}{\sqrt{\rho}}\right)}{\rho}$$

$$s_{23} = \sqrt{2}\sqrt{\rho}$$

$$d_3 = \frac{\sqrt{2}}{\rho}$$

and so on, until we obtain

$$D = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & \frac{\sqrt{1!}}{\sqrt{\rho}} & 0 & \dots & 0 \\ 0 & 0 & \frac{\sqrt{2!}}{(\sqrt{\rho})^2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{\sqrt{N!}}{(\sqrt{\rho})^N} \end{pmatrix}$$

$$S = \begin{pmatrix} -\rho & \sqrt{1!}\sqrt{\rho} & 0 & \dots & 0 & 0 \\ \sqrt{1!}\sqrt{\rho} & -\rho-1 & \sqrt{2!}\sqrt{\rho} & \dots & 0 & 0 \\ 0 & \sqrt{2!}\sqrt{\rho} & -\rho-2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -\rho-(N+1) & \sqrt{N!}\sqrt{\rho} \\ 0 & 0 & 0 & \dots & \sqrt{N!}\sqrt{\rho} & -N \end{pmatrix}$$

The results obtained using the implicit QL algorithm are to machine accuracy, and so the method for finding $\exp(A\Delta\tau)$ will also be very accurate. Writing $\underline{p}_k = \underline{p}(k\Delta\tau, N)$, we have

$$\begin{aligned} \underline{p}_{k+1} &= \exp(A_{k+1}\Delta\tau)\underline{p}_k \\ &= D_{k+1}^{-1}\exp(S_{k+1}\Delta\tau)D_{k+1}D_k^{-1}\exp(S_k\Delta\tau)D_k \dots \qquad (6.2.4) \\ &\qquad D_1^{-1}\exp(S_1\Delta\tau)D_1\underline{p}_0 \end{aligned}$$

where A_k, D_k and S_k are matrices A, D and S but with ρ being replaced by ρ_k, the height of the step-function approximating $\rho(\tau)$ in the appropriate time-interval.

Due to the high accuracy with which we can compute the exponential of the matrix A, the errors involved in calculating the state probabilities for the nonstationary process are mainly due to the discretization of the function $\rho(\tau)$.

We can get an lower and upper bound for the computed blocking probability by using the minimum and maximum values of $\rho(\tau)$ during every time interval, $(\tau, \tau+\Delta\tau]$, as in Figure 6.2.1. The calculated value of $p_N(\tau, N)$, the blocking probability, will be at a minimum when the minimum value of $\rho(\tau)$ is used in the procedure to calculate the new probability distribution vector during each interval, and will be at a maximum when the maximum value of $\rho(\tau)$ is used.

Let $A_{k,max}$ be the matrix $A(\tau)$ when $\rho(\tau)$ is a maximum on the interval $((k-1)\Delta\tau, k\Delta\tau]$ and $A_{k,min}$ be the matrix $A(\tau)$ when $\rho(\tau)$ is a minimum on the same interval. Let the initial probability vector be $\underline{p}_{0,max} = \underline{p}_{0,min} = \underline{p}(0, N)$. Then writing for $k \geq 1$,

$$\begin{aligned} \underline{p}_{k,max} &= \exp(A_{k,max}\Delta\tau)\underline{p}_{k-1,max} \\ \underline{p}_{k,min} &= \exp(A_{k,min}\Delta\tau)\underline{p}_{k-1,min} \end{aligned} \tag{6.2.5}$$

where $(\underline{p}_{k,max})_{N+1}$ is the $(N+1)^{th}$ element of the vector $\underline{p}_{k,max}$, we have $b_{1,max} = (\underline{p}_{1,max})_{N+1}$ and $b_{1,min} = (\underline{p}_{1,min})_{N+1}$ as the two approximations to the exact blocking probability, b_1, at time $\Delta\tau$. This exact probability at time $\Delta\tau$ would lie between the two approximations, that is

$$b_{1,min} \leq b_1 \leq b_{1,max}$$

If $\rho(\tau)$ is constant then $b_{1,min} = b_{1,max}$ and the approximations so calculated are correct to machine accuracy. In general, if $\tau = k\Delta\tau$, then

$$\begin{aligned} b_{k,max} &= (\underline{p}_{k,max})_{N+1} \\ b_{k,max} &= (\underline{p}_{k,max})_{N+1} \end{aligned} \tag{6.2.6}$$

and

$$b_{k,min} \leq b_k \leq b_{k,max}$$

By taking smaller time intervals, we can force these upper and lower bounds for the blocking probability to be closer to one another. For some cases the upper and

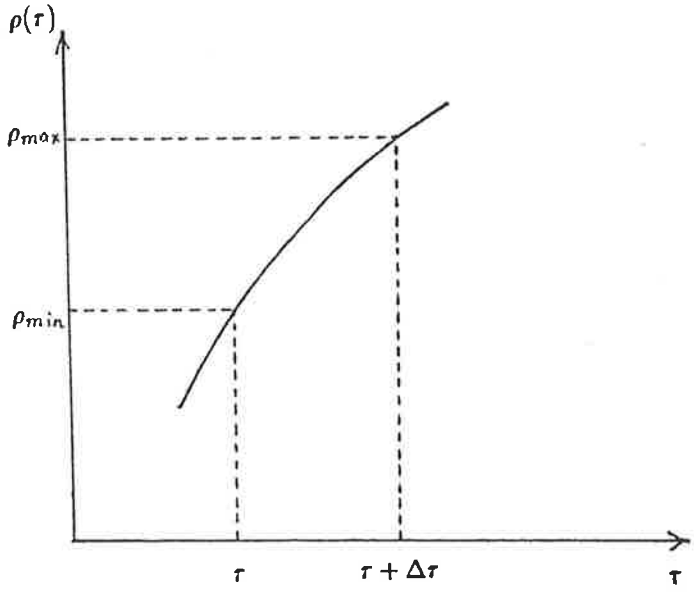

Figure 6.2.1 : Upper and lower bounds of $\rho(\tau)$ *in* $(\tau, \tau + \Delta\tau]$.

lower bounds for the probability are very far apart, indicating the possibility of an inaccurate result. This can happen if $\rho(\tau)$, the number of trunks, N, or the time interval $\Delta\tau$, is too large.

6.3 Numerical Examples

A computer program incorporating the above ideas was developed and several offered traffic patterns were tested with a system consisting of 10 trunks. The time interval used in all the examples was 0.1, that is $\Delta\tau = 0.1$. The numerical results at specific time τ were plotted with graphs of exact blocking probability for the same offered traffic and number of trunks, as given by Jagerman [8]. The graphs and the corresponding offered traffic functions are

Figure 6.3.1	$\rho(\tau) = 7$
Figure 6.3.2	$\rho(\tau) = 5 + 2sin(0.05\pi\tau)$
Figure 6.3.3	$\rho(\tau) = 5 + 2sin(0.1\pi\tau)$
Figure 6.3.4	$\rho(\tau) = 5 + 2sin(0.5\pi\tau)$

The above functions represent functions with rapid and slow oscillations. The continuous lines are the exact blocking probability and the crosses represent the probability obtained using the above procedure. The crosses are actually the mean of the upper and lower bounds of the probability. However the upper and lower bound points for these examples are so close together that on the graphs they cannot be distinguished from one another and the exact results. Table 6.3.1 is a table of these upper and lower bounds for $\rho(\tau) = 10(1 - e^{-(\tau-1)})$ with different $\Delta\tau$'s.

As can be seen from the above graphs, the numerical method gives very good results.

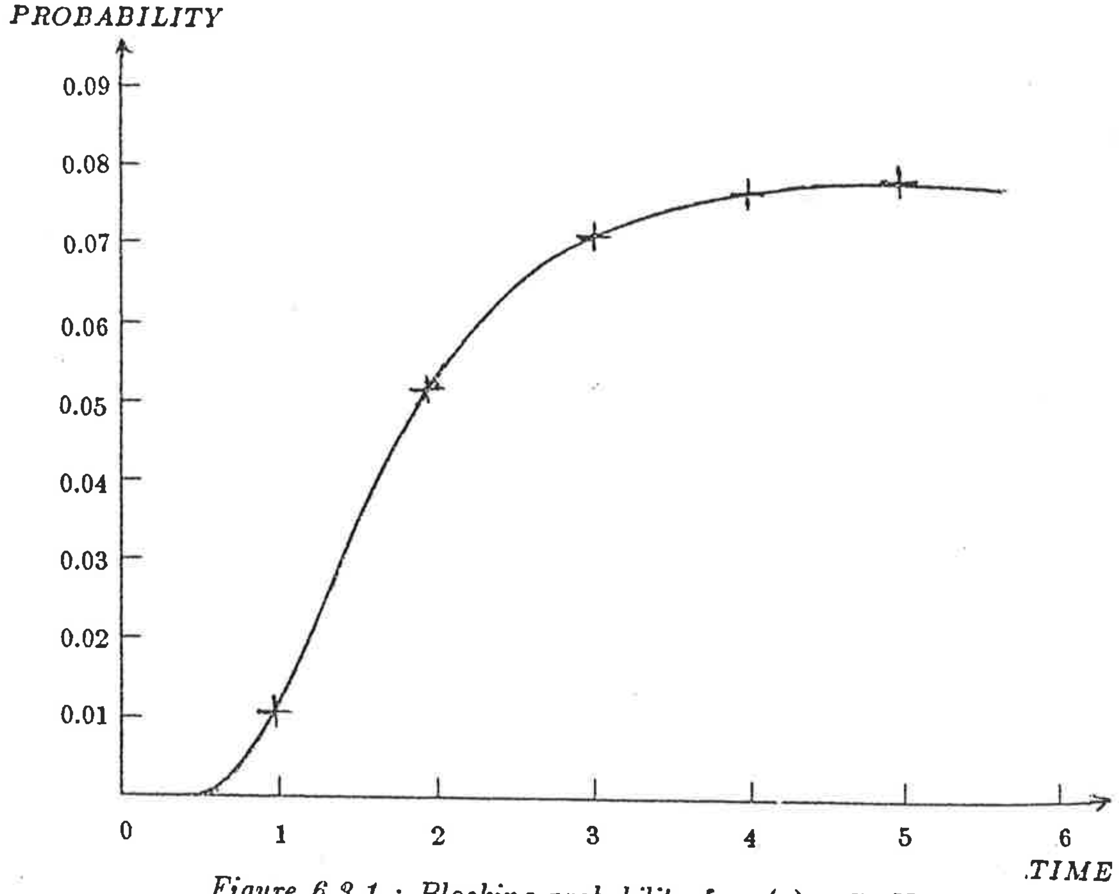

Figure 6.3.1 : Blocking probability for $\rho(\tau) = 7$, $N = 7$.

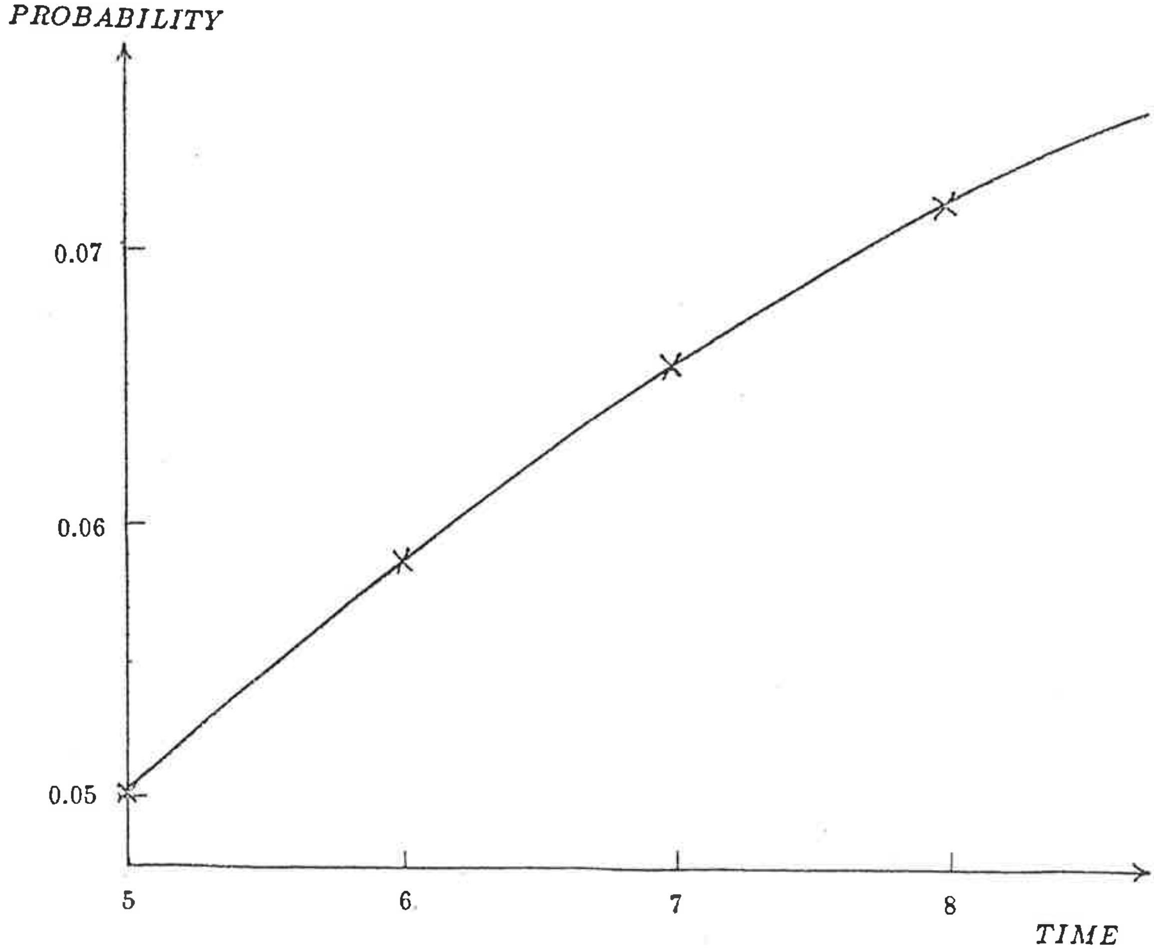

Figure 6.3.2 : Blocking probability for $\rho(\tau) = 5 + 2sin(0.05\pi\tau)$, $N = 7$.

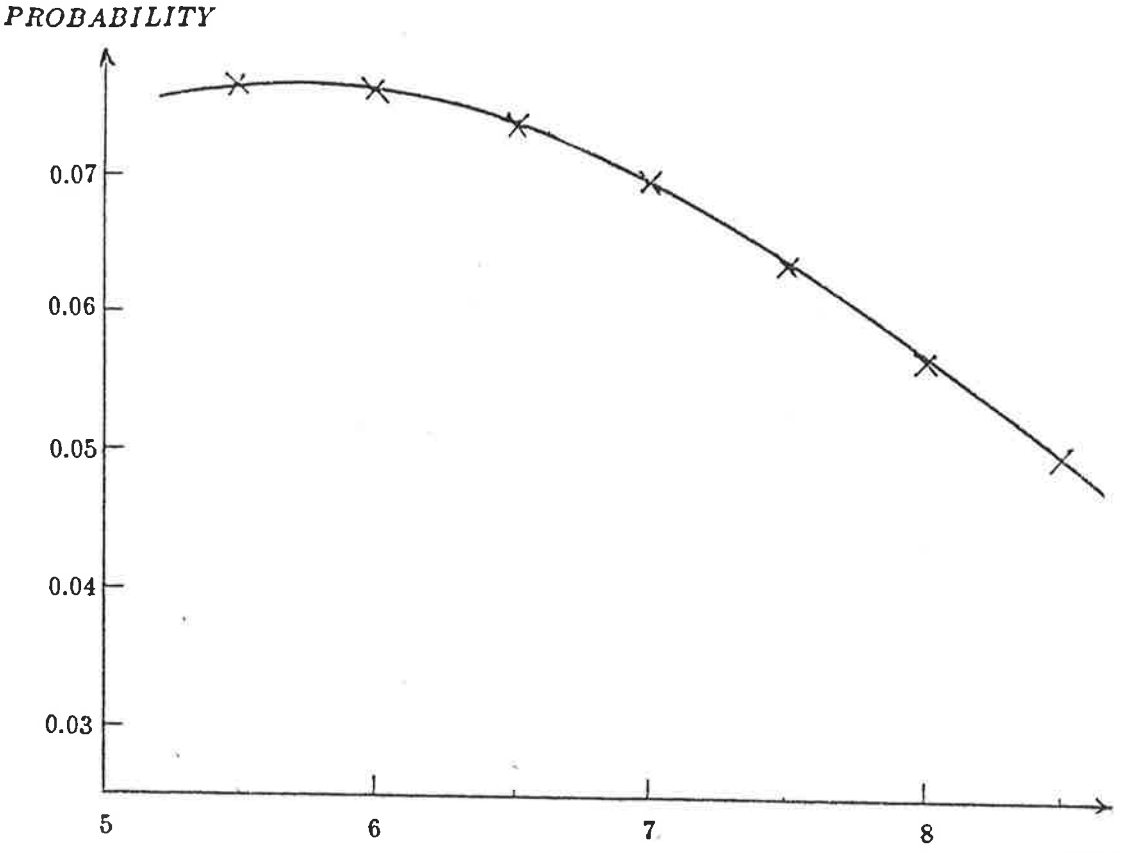

Figure 6.3.3 : Blocking probability for $\rho(\tau) = 5 + 2sin(0.1\pi\tau)$, $N = 7$.

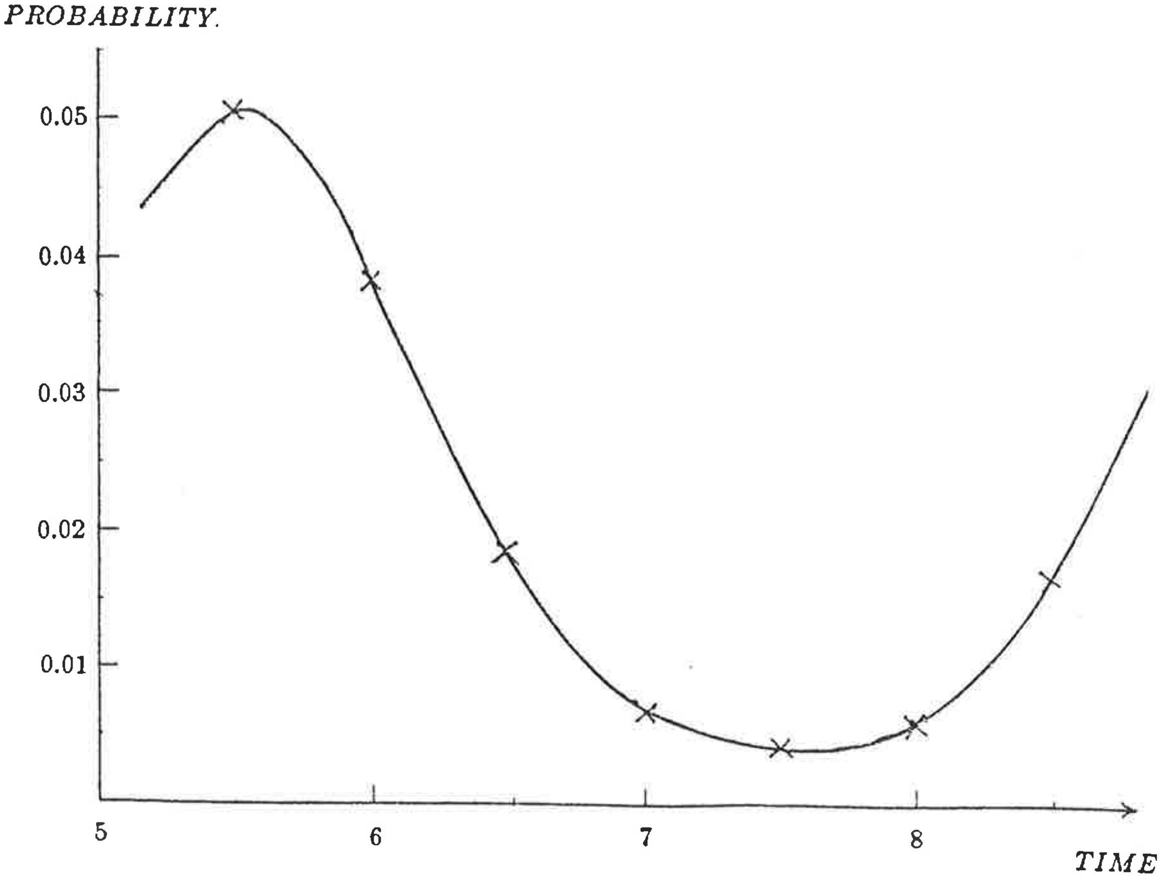

Figure 6.3.4 : Blocking probability for $\rho(\tau) = 5 + 2sin(0.5\pi\tau)$, $N = 7$.

TIME	$\Delta\tau = 0.01$		$\Delta\tau = 0.001$	
	L.B.	U.B.	L.B.	U.B
0.1	6.7276E-7	1.0520E-6	7.2291E-6	5.9719E-6
0.2	6.5468E-6	6.0522E-6	1.1283E-5	1.0253E-5
0.3	9.3232E-5	9.6405E-5	9.0763E-5	8.8732E-5
0.4	6.1272E-4	6.3152E-4	6.1149E-4	6.1175E-4
0.5	2.2991E-3	2.3589E-3	2.3192E-3	2.3236E-3
0.6	6.0826E-3	6.2180E-3	6.1425E-3	6.1519E-3

Table 6.3.1

Blocking probability for $\rho(\tau) = 10(1 - e^{-(\tau-1)}), \Delta\tau = 0.01, 0.001$.

6.4 Error and Time Analyses

We now discuss the error and time analyses of the above procedure. First of all we have to define the matrix and vector norms that we are going to use in the analyses. For a matrix A and a vector $\underline{x}$, their norms are defined as

$$\begin{aligned} \|A\| &= \max_{1\le i\le N+1} \sum_{j=1}^{N+1} |a_{ij}| \\ \|\underline{x}\| &= \max_{1\le i\le N+1} |x_i| \end{aligned} \tag{6.4.1}$$

Then it can be verified that (Bellman [1])

$$\begin{aligned} \|AB\| &\le \|A\|\|B\| \\ \|A\underline{x}\| &\le \|A\|\|\underline{x}\| \\ \left\|\sum_{j=1}^{\infty} \underline{x}_j\right\| &\le \sum_{j=1}^{\infty} \|\underline{x}_j\| \end{aligned} \tag{6.4.2}$$

Let

$$\underline{d}_k = \underline{p}_{k,max} - \underline{p}_{k,min}$$

Then, for any definition of matrix norm, we have

$$\begin{aligned} \|\underline{p}_{k+1}\| &= \|\underline{p}_{k+1,max} - \underline{p}_{k+1,min}\| \\ &= \|\exp(A_{max}\Delta\tau)\underline{p}_{k,max} - \exp(A_{min}\Delta\tau)\underline{p}_{k,min}\| \\ &= \|\exp(A_{min}\Delta\tau + E\Delta\tau)\underline{p}_{k,max} - \exp(A_{min}\Delta\tau)\underline{p}_{k,min}\| \end{aligned} \tag{6.4.3}$$

where $A_{max} = A_{k+1,max}$, $A_{min} = A_{k+1,min}$ and $E = A_{max} - A_{min}$, so that matrix E has the following form

$$E = \begin{pmatrix} -\epsilon & 0 & 0 & \dots & 0 & 0 \\ \epsilon & -\epsilon & 0 & \dots & 0 & 0 \\ 0 & \epsilon & -\epsilon & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -\epsilon & 0 \\ 0 & 0 & 0 & \dots & \epsilon & 0 \end{pmatrix}$$

where $\epsilon = \rho_{max} - \rho_{min}$ (see Figure 6.2.1).

If we assume that $\|A_{max}\Delta\tau\| \leq 1$ for all τ, then

$$\begin{aligned}
\|\underline{d}_{k+1}\| &= \left\| \sum_{i=0}^{\infty} \frac{(A_{min}\Delta\tau + E\Delta\tau)^i}{i!} \underline{p}_{k,max} - \sum_{i=0}^{\infty} \frac{(A_{min}\Delta\tau)^i}{i!} \underline{p}_{k,min} \right\| \\
&= \left\| \left(I + (A_{min}\Delta\tau + E\Delta\tau) + \frac{(A_{min}\Delta\tau + E\Delta\tau)^2}{2!} \right.\right. \\
&\qquad \left. + \frac{(A_{min}\Delta\tau + E\Delta\tau)^3}{3!} + \cdots \right) \underline{p}_{k,max} \\
&\qquad \left. - \left(I + A_{min}\Delta\tau + \frac{A_{min}^2\Delta\tau^2}{2!} + \frac{A_{min}^3\Delta\tau^3}{3!} + \cdots \right) \underline{p}_{k,min} \right\| \\
&\leq \| \exp(A_{min}\Delta\tau)(\underline{p}_{k,max} - \underline{p}_{k,min}) \| \\
&\qquad + \left(\|E\Delta\tau\| + \left\| \frac{(A_{min}\Delta\tau + E\Delta\tau)^2 - A_{min}^2\Delta\tau^2}{2!} \right\| \right. \\
&\qquad \left. + \left\| \frac{(A_{min}\Delta\tau + E\Delta\tau)^3 - A_{min}^3\Delta\tau^3}{3!} \right\| + \cdots \right) \|\underline{p}_{k,max}\| \qquad (6.4.4)
\end{aligned}$$

Now

$$\begin{aligned}
\left\| \frac{(A_{min}\Delta\tau + E\Delta\tau)^{k+1} - A_{min}^{k+1}\Delta\tau^{k+1}}{(k+1)!} \right\| &= \left\| \frac{(A_{min}\Delta\tau + E\Delta\tau)}{k+1} \frac{(A_{min}\Delta\tau + E\Delta\tau)^k}{k!} \right. \\
&\qquad \left. - \frac{A_{min}\Delta\tau}{k+1} \frac{A_{min}^k\Delta\tau^k}{k!} \right\| \\
&= \left\| \frac{A_{min}\Delta\tau}{k+1} \left(\frac{(A_{min}\Delta\tau + E\Delta\tau)^k}{k!} - \frac{A_{min}^k\Delta\tau^k}{k!} \right) \right. \\
&\qquad \left. + \frac{E\Delta\tau}{k+1} \frac{(A_{min}\Delta\tau + E\Delta\tau)^k}{k!} \right\| \\
&\leq \left\| \frac{A_{min}\Delta\tau}{k+1} \right\| \left\| \frac{(A_{min}\Delta\tau + E\Delta\tau)^k}{k!} - \frac{A_{min}^k\Delta\tau^k}{k!} \right\| \\
&\qquad + \left\| \frac{E\Delta\tau}{k+1} \right\| \left\| \frac{(A_{min}\Delta\tau + E\Delta\tau)^k}{k!} \right\| \\
&\leq \frac{1}{k+1} \left\| \frac{(A_{min}\Delta\tau + E\Delta\tau)^k}{k!} - \frac{A_{min}^k\Delta\tau^k}{k!} \right\| \\
&\qquad + \left\| \frac{E\Delta\tau}{k+1} \right\| \frac{1}{k!} \qquad (6.4.5)
\end{aligned}$$

Now, using (6.4.1), we have

$$||A_{min}|| = 2\rho + 2N - 1$$

$$||E|| = 2\epsilon$$

where $\rho = \rho_{k+1,min}$.

Therefore (6.4.11) can be written as

$$||\underline{d}_{k+1}|| \leq e^{(2\rho+2N-1)\Delta\tau}||\underline{d}_k|| + (e-2)(2(2\rho+2N-1)(2\epsilon\Delta\tau^2) + 4\epsilon^2\Delta\tau^2) + (4-e)(2\epsilon\Delta\tau) \tag{6.4.12}$$

As an example, let $N = 10$, $\rho = 7$, $\Delta\tau = 0.001$ and $\epsilon = 0.01$. Inequality (6.4.12) is now

$$||\underline{d}_{k+1}|| \leq 1.0336||\underline{d}_k|| + (2.6583 \times 10^{-5}) \tag{6.4.13}$$

Therefore, as can be seen from (6.4.3), the error propagated from one time interval to another is quite small.

The accuracy of the results obtained is dependent on four things. As the time step, $\Delta\tau$, decreases, the errors also decrease, and in most cases the difference between the maximum offered traffic and the minimum offered traffic over this time interval, ϵ, will also decrease. It is by shortening the time interval used that more accurate results can be obtained, although this leads to a longer running time.

As the value of the offered traffic at time τ, $\rho(\tau)$, or the number of trunks in the system, N, increases, the error will also increase. If $(\rho + N)\Delta\tau$ becomes too large, the error may increase very rapidly until results obtained are useless. The difference between the lower and the upper bounds obtained is a true indication of the errors involved. These actual errors are nearly always much less than the errors obtained

N	$\Delta\tau$	CPU sec	
10	0.1	9.2	constant N, varying $\Delta\tau$
10	0.01	85.0	
10	0.001	820.0	
5	0.1	2.3	varying N, constant $\Delta\tau$
10	0.1	9.0	
20	0.1	42.0	
40	0.1	272.0	

Table 6.4.1 : $\rho(\tau) = 10(1 - e^{-(\tau-1)})$.

using the above analysis which nevertheless indicates in what way error bounds will change as the various parameters involved either increase or decrease.

The number of operations required to calculate the blocking probability depends on two factors. The first of these is the time interval, $\Delta\tau$, or the number of times the process described above must be repeated. If $\Delta\tau$ is halved then the CPU time required to find the probability vector at a given time will double.

The operations required to find $\exp(A\Delta\tau)\underline{p}$ are of the order N^2, that is as the number of trunks, N, involved increases, the CPU time needed will also increase, as a quadratic function of N. If N is doubled, the CPU time will increase by a factor of 4. If N is trebled, the CPU time will increase by a factor of 9, and so on.

The table in Table 6.4.1 shows the CPU time needed to calculate, to a given accuracy, some results using different values of $\Delta\tau$ and N. The test offered traffic function was $\rho(\tau) = 10(1 - e^{-(\tau+1)})$.

6.5 Discussion

This numerical method is efficient as long as ρ, N and $\Delta\tau$ are not too large. All three quatities are interrelated in the error analysis as shown by (6.4.12). The choice of $\Delta\tau$ would depend on the offered traffic function, $\rho(\tau)$ (the steeper its slope, the smaller $\Delta\tau$ should be), and the restrictions on ρ and N will vary accordingly. However, in real-life, traffic may be greater than a few hundred erlangs in parts of the network, and thus, in this case, the method becomes very inefficient. However, in systems with offered traffic below 20 trunks, this method works effectively.

CHAPTER VII

APPROXIMATE METHODS TO EVALUATE TIME-DEPENDENT BLOCKING PROBABILITY FOR THE M(t)/M(t)/N LOSS SYSTEM

7.1 Introduction

In Chapter VI we presented a numerical procedure for evaluating the time-dependent blocking probability for the M(t)/M(t)/N loss system, but this procedure is quite inefficient when N is larger than about 20. This is because of the computation errors involved and the computation time is also very long as discussed in Section 6.4. Therefore for large N we will have to use approximate methods developed for the transient M/M/N loss system, as described in Chapter III.

These approximate methods can be easily extended to the M(t)/M(t)/N loss system by using the discretization technique which is outlined in Chapter IV, and applied in Chapter V. We simply use step-functions to approximate the offered traffic function. We then apply the approximate methods given in Chapter III on each of these step-functions to approximate the transient blocking probability in each subinterval. This transient probability is then used to approximate the time-dependent blocking probability. All these approximations utilize the Erlang Loss Function, as mentioned in Chapter III. We shall look at four approximate methods.

The first method is the most well-known and has been widely used because of its simplicity. In this case the blocking probability is approximated by $E_N(\rho(\tau))$. The second approximate method is an approximation using the time-dependent offered traffic, which was discussed in Section 1.4. The third method is based on Jagerman's approximation for the transient blocking probability for the M/M/N loss system. We modify it for application in the M(t)/M(t)/N loss systems. The last method is based on the modified offered traffic which we derived for the transient M/M/N loss system as given in Section 3.3.

All the above approximate methods for the M(t)/M(t)/N loss system will be outlined in separate sections and their performance will be compared in Section 7.6.

7.2 The $E_N(\rho(\tau))$ Approximation

This is the easiest approximation to use, since we just evaluate the function $\rho(\tau)$ at time τ and use this value in the Erlang Loss Function to approximate the blocking probability at time τ. However, due to the asssumption of the system being in the steady state while deriving the Erlang Loss Function, this function (the Erlang Loss Function) 'sees' the function $\rho(\tau)$ differently.

Suppose that $\rho(\tau)$ is the function shown in Figure 7.2.1, and we want to approximate the blocking probability at time τ_1, say. Note that $\dot{\rho}(\tau_1)$ is positive. When we use $E_N(\rho(\tau_1))$ to make the approximation, the Erlang Loss Function 'sees' $\rho(\tau)$ as shown in Figure 7.2.2, which is a different function to Figure 7.2.1. The steady state assumption in its derivation means that as far as the Erlang Loss Function is concerned, $\rho(\tau)$ starts at $\tau = -\infty$ at a constant intensity of $\rho(\tau_1)$ up to $\tau = \tau_1$. What happens after this instant is irrelevant to the Erlang Loss Function. Hence we expect the approximated probability to be higher than the true probability because the real traffic intensity prior to τ_1 is much lower than what the Erlang Loss Function 'sees'.

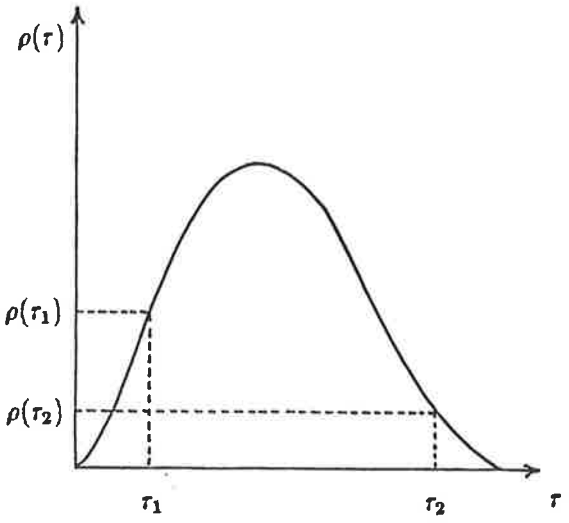

Figure 7.2.1 : Continuous offered traffic, $\rho(\tau)$.

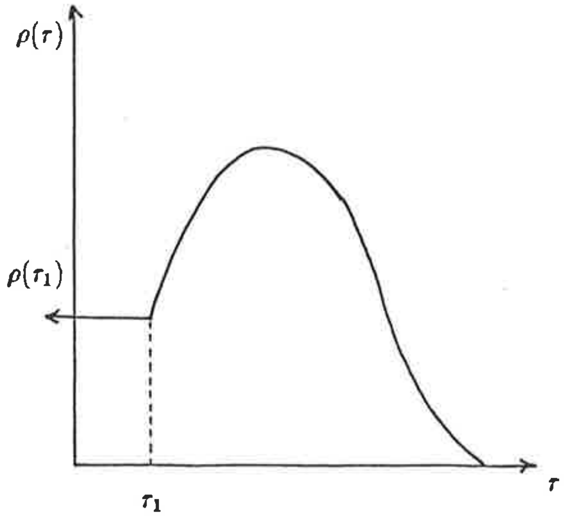

Figure 7.2.2 : The above function as 'seen' by the Erlang Loss Function when approximating at τ_1.

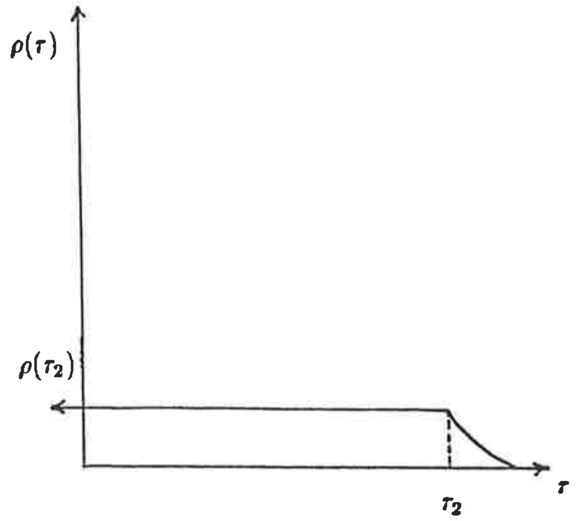

Figure 7.2.3 : The above function as 'seen' by the Erlang Loss Function when approximating at τ_2.

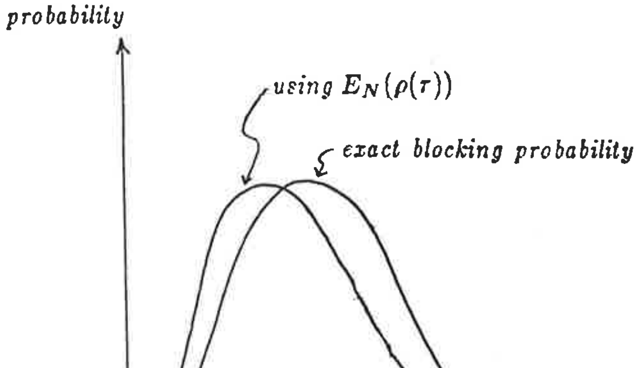

Figure 7.2.4 : The 'delay' experienced when using $E_N(\rho(\tau))$.

On the other hand, if we wish to approximate the blocking probability at $\tau = \tau_2$, where $\dot{\rho}(\tau_2)$ is negative, as shown in Figure 7.2.1, the opposite effect will take place. At τ_2, $E_N(\rho(\tau_2))$ 'sees ' $\rho(\tau)$ as a function shown in Figure 7.2.3. Here the real traffic could cause a higher blocking than the lower steady traffic 'seen' by the Erlang Loss Function prior to τ_2. This could result in an underestimation of the true blocking probability at τ_2.

Combining the effects illustrated in Figures 7.2.2 and 7.2.3, approximations using $E_N(\rho(\tau))$ will be shifted to the left of the true blocking probability as sketched in Figure 7.3.4. The degree of the shift depends on $\dot{\rho}(\tau)$. Therefore the steeper the slope of $\rho(\tau)$ is, the more inaccurate our approximation becomes.

In fact the abovementioned shift has been reported by users of this approximation but no explanation has been given (see for example, Ohsawa, et. al. [20], Miwa [15]). It should only be used when the slope is very small or when congestion is high, because in this case the steady state is almost achieved in very short time-intervals.

In Section 7.6 we compare its performance with the other approximations and the numerical method discussed in the previous chapter.

7.3 Time-Dependent Offered Traffic

For this approximation, we shall use expression (1.4.2) which is

$$a(t) = \int_0^t \lambda(t-u)H^c(u)du + a_0 H^c(t) \tag{7.3.1}$$

where $a(t)$ is the offered traffic at time t, $H^c(t)$ is the complementary service time distribution and a_0 is the 'initial' offered traffic (Jagerman [8]). This approximation is then

$$p_N(t,N) \approx E_N(a(t)) \tag{7.3.1}$$

with $a(t)$ as defined above.

In the case of constant offered traffic and normalized mean service time, we have

$$a(\tau) = \lambda(1-e^{-\tau}) + a_0 e^{-\tau} \tag{7.3.2}$$

Note that we are using the transformed time-axis as described in Chapter V and that a_0 is the initial offered traffic.

We will use (7.3.2) in our example in Section 7.6.

7.4 Jagerman's Approximation

To be able to use Jagerman's approximation (3.5.4) in our case, we shall have to obtain (3.5.4) which will take into account any initial condition. To derive such an equation, we will have to use equation (3.5.5) with any initial condition, namely

$$\tilde{p}_1(\tau,1) = E_1(\frac{a}{N})(1-e^{-(\frac{a}{N}+1)\tau}) + \tilde{p}_1(\tau_0,1)e^{-(\frac{a}{N}+1)\tau} \tag{7.4.1}$$

where $p_1(\tau_0,1)$ is the 'initial' blocking probability due to previous time-interval.

Now we insert (7.4.1) in (3.5.9) to obtain

$$\begin{aligned} p_N(\tau,N) &\approx \frac{E_N(a)}{\left(E_1\left(\frac{a}{N}\right)\right)^N}(\tilde{p}_1(\tau,N))^N \\ &= E_N(a)(1-e^{-(\frac{a}{N}+1)\tau} + \frac{\tilde{p}_1(\tau_0,1)}{E_1(\frac{a}{N})}e^{-(\frac{a}{N}+1)\tau})^N \end{aligned} \tag{7.4.2}$$

Expression (7.4.2) is now used to approximate the blocking probability with the initial probability, $\tilde{p}_1(\tau_0,1)$, obtained by applying (7.4.1) in the previous time-interval.

7.5 Modified Offered Traffic

We now consider the approximation obtained by using the modified offered traffic function which was derived in Section 3.4 for the case of the transient M/M/N loss system. For the M(t)/M(t)/N loss system, the expression (3.4.1) can be used directly, as the initial condition is taken care of in the algorithm.

7.6 Comparison of Approximate Methods

All the above approximate methods are tested with three sinuisodal functions of period 20, 10 and 5. They represent functions with varying degree of time-fluctuations. The functions are:

(a) $a(\tau) = 10 + 5sin(0.1\pi\tau)$

(b) $a(\tau) = 10 + 5sin(0.2\pi\tau)$

(c) $a(\tau) = 10 + 5sin(0.4\pi\tau)$

Function (a) has period 20 which means that its value can increase by 5 in 5 units of time. This is a slowly fluctuating function. Function (b) on the other hand can increase its value by 5 in 2.5 units of time. This can represent a moderately fluctuating function. Function (c) is a highly fluctuating function since its value can increase by 5 in just 1.25 units of time.

For simplicity we let $\frac{1}{\mu} = 1$ and hence we will use the above functions directly in the approximations.

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	APP4	ERROR
1.0	0.5805	0.6137	0.3320	0.4183	-0.1622	0.4666	-0.1139	0.4895	-0.0910
2.0	0.6461	0.6501	0.0040	0.5765	-0.0696	0.6185	-0.0276	0.6303	-0.0158
3.0	0.6724	0.6746	0.0022	0.6732	-0.0352	0.6568	-0.0156	0.6666	-0.0058
4.0	0.6877	0.6887	0.0010	0.6679	-0.0198	0.6794	-0.0083	0.6847	-0.0030
5.0	0.6932	0.6932	0.0000	0.6836	-0.0096	0.6914	-0.0018	0.6923	-0.0009
6.0	0.6896	0.6887	-0.0009	0.6885	-0.0011	0.6941	0.0045	0.6907	0.0011
7.0	0.6766	0.6746	-0.0020	0.6841	0.0075	0.6879	0.0113	0.6789	0.0023
8.0	0.6535	0.6501	-0.0034	0.6702	0.0167	0.6725	0.0190	0.6593	0.0058
9.0	0.6193	0.6137	-0.0056	0.6461	0.0268	0.6472	0.0279	0.6282	0.0089
10.0	0.5726	0.5640	-0.0086	0.6106	0.0380	0.6102	0.0376	0.5856	0.0130

Table 7.6.1 : $a(\tau) = 10 + 5sin(0.1\pi\tau), N = 5.$

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	APP4	ERROR
1.0	0.1142	0.2830	0.1688	0.0745	-0.0397	0.0598	-0.0544	0.0777	-0.0365
2.0	0.3075	0.3389	0.0314	0.2309	-0.0766	0.2406	-0.0669	0.2505	-0.0570
3.0	0.3698	0.3788	0.0090	0.3185	-0.0513	0.3248	-0.0450	0.3401	-0.0297
4.0	0.3987	0.4025	0.0038	0.3678	-0.0309	0.3721	-0.0266	0.3843	-0.0144
5.0	0.4097	0.4103	0.0006	0.4023	-0.0074	0.4015	-0.0082	0.4042	-0.0055
6.0	0.4049	0.4025	-0.0024	0.4023	-0.0026	0.4155	0.0106	0.4062	0.0013
7.0	0.3846	0.3788	-0.0058	0.3947	0.0101	0.4140	0.0294	0.3924	0.0078
8.0	0.3489	0.3389	-0.0100	0.3714	0.0225	0.3961	0.0472	0.3633	0.0144
9.0	0.2987	0.2830	-0.0157	0.3325	0.0338	0.3596	0.0609	0.3194	0.0207
10.0	0.2368	0.2146	-0.0222	0.2784	0.0416	0.3026	0.0658	0.2626	0.0258

Table 7.6.2 : $a(\tau) = 10 + 5sin(0.1\pi\tau), N = 10.$

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	APP4	ERROR
1.0	0.0000	0.0071	0.0071	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2.0	0.0032	0.0175	0.0143	0.0026	-0.0006	0.0035	-0.0004	0.0026	-0.0006
3.0	0.0153	0.0307	0.0154	0.0128	-0.0025	0.0148	-0.0005	0.0129	-0.0024
4.0	0.0304	0.0415	0.0111	0.0264	-0.0040	0.0285	-0.0019	0.0266	-0.0038
5.0	0.0407	0.0456	0.0049	0.0373	-0.0034	0.0401	-0.0006	0.0376	-0.0031
6.0	0.0425	0.0415	-0.0010	0.0414	-0.0011	0.0454	0.0029	0.0416	-0.0009
7.0	0.0362	0.0307	-0.0055	0.0376	0.0014	0.0422	0.0060	0.0376	0.0014
8.0	0.0249	0.0175	-0.0074	0.0278	0.0029	0.0310	0.0061	0.0276	0.0027
9.0	0.0133	0.0071	-0.0062	0.0159	0.0026	0.0169	0.0036	0.0158	0.0025
10.0	0.0052	0.0019	-0.0033	0.0065	0.0013	0.0062	0.0010	0.0065	0.0013

Table 7.6.3 : $a(\tau) = 10 + 5sin(0.1\pi\tau), N = 20.$

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	APP4	ERROR
1.0	0.6220	0.6501	0.0281	0.4493	-0.1727	0.4966	-0.1254	0.5298	-0.0922
2.0	0.6863	0.6887	0.0024	0.6170	-0.0693	0.6634	-0.0229	0.6739	-0.0124
3.0	0.6901	0.6887	-0.0014	0.6677	-0.0224	0.6957	0.0056	0.6906	0.0005
4.0	0.6563	0.6501	-0.0062	0.6692	0.0129	0.6901	0.0338	0.6657	0.0094
5.0	0.5798	0.5640	-0.0158	0.6318	0.0520	0.6497	0.0699	0.6015	0.0217
6.0	0.4605	0.4283	-0.0322	0.5530	0.0925	0.5590	0.0985	0.4969	0.0364
7.0	0.3385	0.3044	-0.0341	0.4412	0.1027	0.4093	0.0708	0.3770	0.0385
8.0	0.2982	0.3044	0.0062	0.3530	0.0548	0.2981	-0.0001	0.3120	0.0138
9.0	0.3878	0.4283	0.0405	0.3649	-0.0229	0.3237	-0.0641	0.3630	-0.0248
10.0	0.5393	0.5640	0.0247	0.4656	-0.0737	0.4548	-0.0845	0.4995	-0.0398

Table 7.6.4 : $a(\tau) = 10 + 5sin(0.2\pi\tau), N = 5$.

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	APP4	ERROR
1.0	0.1546	0.3389	0.1843	0.0966	-0.0580	0.0706	-0.0840	0.1021	-0.0525
2.0	0.3819	0.4025	0.0206	0.2879	-0.0940	0.3002	-0.0817	0.3171	-0.0648
3.0	0.4054	0.4025	-0.0029	0.3674	-0.0380	0.4076	0.0022	0.3904	-0.0150
4.0	0.3561	0.3389	-0.0172	0.3699	0.0138	0.4878	0.1317	0.3729	0.0168
5.0	0.2528	0.2146	-0.0382	0.3103	0.0575	0.3900	0.1372	0.2934	0.0406
6.0	0.1303	0.0812	-0.0491	0.2009	0.0706	0.2320	0.1017	0.1766	0.0463
7.0	0.0505	0.0233	-0.0272	0.0905	0.0400	0.0734	0.0229	0.0763	0.0258
8.0	0.0275	0.0233	-0.0042	0.0400	0.0125	0.0262	-0.0013	0.0353	0.0078
9.0	0.0462	0.0812	0.0350	0.0452	-0.0010	0.0322	-0.0140	0.0434	-0.0028
10.0	0.1416	0.2146	0.0730	0.1099	-0.0317	0.0858	-0.0558	0.1129	-0.0287

Table 7.6.5 : $a(\tau) = 10 + 5sin(0.2\pi\tau), N = 10$.

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	APP4	ERROR
1.0	0.00033	0.01750	0.01717	0.00006	-0.00027	0.00006	-0.00027	0.00006	-0.00027
2.0	0.01220	0.04147	0.02927	0.00773	-0.00490	0.00911	-0.00309	0.00775	-0.00445
3.0	0.00194	0.04147	0.03953	0.02630	0.02436	0.03543	0.03349	0.02658	0.02464
4.0	0.00003	0.01750	0.01747	0.02721	0.02718	0.04046	0.04043	0.02732	0.02729
5.0	0.00008	0.00187	0.00179	0.01123	0.01115	0.01406	0.01398	0.01115	0.01107
6.0	0.00638	0.00003	-0.00635	0.00138	-0.00500	0.00097	-0.00541	0.00137	-0.00501
7.0	0.02099	0.00000	-0.02099	0.00005	-0.02094	0.00002	-0.02097	0.00005	-0.02094
8.0	0.00242	0.00000	-0.00242	0.00000	-0.00242	0.00000	-0.00242	0.00000	-0.00242
9.0	0.00003	0.00003	0.00000	0.00000	-0.00003	0.00000	-0.00003	0.00000	-0.00003
10.0	0.00009	0.00187	0.00178	0.00011	0.00002	0.00010	0.00001	0.00010	0.00001

Table 7.6.6 : $a(\tau) = 10 + 5sin(0.2\pi\tau), N = 20$.

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	APP4	ERROR
1.0	0.6716	0.6887	0.0171	0.4938	-0.1778	0.5517	-0.1199	0.5863	-0.0853
2.0	0.6610	0.6501	-0.0109	0.6287	-0.0323	0.7110	0.0500	0.6690	0.0080
3.0	0.4835	0.4283	-0.0552	0.5800	0.0965	0.6715	0.1880	0.5386	0.0551
4.0	0.3140	0.3044	-0.0096	0.4402	0.1262	0.3609	0.0496	0.3612	0.0472
5.0	0.5070	0.5640	0.0570	0.4534	-0.0536	0.3721	-0.1349	0.4565	-0.0505
6.0	0.6825	0.6887	0.0062	0.5993	-0.0832	0.6226	-0.0599	0.6548	-0.0277
7.0	0.6612	0.6501	-0.0111	0.6540	-0.0073	0.7127	0.0515	0.6723	0.0111
8.0	0.4836	0.4283	-0.0553	0.5925	0.1089	0.6723	0.1887	0.5399	0.0563
9.0	0.3140	0.3044	-0.0096	0.4482	0.1342	0.3622	0.0482	0.3616	0.0476
10.0	0.5077	0.3640	-0.1437	0.4558	-0.0519	0.3709	-0.1368	0.4554	-0.0523

Table 7.6.7 : $a(\tau) = 10 + 5sin(0.4\pi\tau), N = 5.$

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	APP4	ERROR
1.0	0.2266	0.4025	0.1759	0.1356	-0.0910	0.0963	-0.1303	0.1462	-0.0804
2.0	0.3598	0.3389	-0.0209	0.3054	-0.0544	0.4346	0.0748	0.3319	-0.0279
3.0	0.1621	0.0812	-0.0809	0.2354	0.0733	0.4365	0.2744	0.2244	0.0623
4.0	0.0432	0.0233	-0.0199	0.0897	0.0465	0.0604	0.0172	0.0753	0.0321
5.0	0.1089	0.2146	0.1057	0.0998	-0.0091	0.0486	-0.0603	0.0959	-0.0130
6.0	0.3590	0.4025	0.0435	0.2625	-0.0965	0.2186	-0.1404	0.2844	-0.0746
7.0	0.3651	0.3389	-0.0262	0.3451	-0.0200	0.4636	0.0985	0.3642	-0.0009
8.0	0.1625	0.0812	-0.0813	0.2527	0.0902	0.4432	0.2807	0.2333	0.0708
9.0	0.0433	0.0233	-0.0200	0.0958	0.0525	0.0612	0.0179	0.0774	0.0341
10.0	0.1089	0.2146	0.1057	0.1018	-0.0071	0.0483	-0.0606	0.0960	-0.0129

Table 7.6.8 : $a(\tau) = 10 + 5sin(0.4\pi\tau), N = 10.$

TIME	EXACT	APP1	ERROR	APP2	ERROR	APP3	ERROR	APP4	ERROR
1.0	0.00033	0.04147	0.04114	0.00025	-0.00008	0.00017	-0.00016	0.00025	-0.00008
2.0	0.01220	0.01750	0.00530	0.01038	-0.00182	0.02292	0.01072	0.01043	-0.00177
3.0	0.00194	0.00003	-0.00191	0.00290	0.00096	0.00504	0.00310	0.00291	0.00097
4.0	0.00003	0.00000	-0.00003	0.00005	0.00002	0.00002	-0.00001	0.00005	0.00002
5.0	0.00008	0.00187	0.00179	0.00007	-0.00001	0.00003	-0.00005	0.00007	-0.00001
6.0	0.00638	0.04147	0.03509	0.00488	-0.00150	0.00352	-0.00286	0.00485	-0.00153
7.0	0.02099	0.01750	-0.00349	0.01918	-0.00181	0.03655	0.01556	0.01934	-0.00165
8.0	0.00242	0.00003	-0.00239	0.00407	0.00165	0.00585	0.00343	0.00409	0.00167
9.0	0.00003	0.00000	-0.00003	0.00006	0.00003	0.00002	-0.00001	0.00006	0.00003
10.0	0.00009	0.00187	0.00178	0.00008	-0.00001	0.00003	-0.00006	0.00008	-0.00001

Table 7.6.9 : $a(\tau) = 10 + 5sin(0.4\pi\tau), N = 20.$

We also calculate the numerical blocking probability for the above traffic pattern using the method given in the previous chapter. Three cases are considered with N taking the values 5, 10 and 20 respectively. Tables 7.6.1, 7.6.4 and 7.6.7 are for $N = 5$, Tables 7.6.2, 7.6.5 and 7.6.8 are for $N = 10$, and Tables 7.6.3, 7.6.6 and 7.6.9 are for $N = 20$.

In the tables, the method in Section 7.2 is denoted by APP1, Section 7.3 by APP2, Section 7.4 by APP3 and Section 7.5 by APP4. The exact results are obtained numerically using the method given in the previous chapter with $\Delta t = 0.001$ and the mean of the lower and the upper bounds are taken.

From the tables we can make the following deductions:

(a) When traffic fluctuation is expected to be low to high with the congestion expected to be high, APP1 gives the best approximation. This is because when congestion is high the steady state is almost achieved in a very short time-interval, although of course the steeper the slope of the traffic function is, the worse it becomes. APP4 gives a slightly worse approximation.

(b) When traffic fluctuation is expected to be low to high while the congestion is expected to be low to moderate, APP4 gives the best approximation.

Therefore overall performance of APP4 is very encouraging and its use is recommended in approximating time-dependent blocking probability, although when congestion is expected to be high to very high APP1 will suffice.

CORRECTION

The formula (7.7.2) is not for discrete offered traffic, as indicated in the text, but for piecewise linear offered traffic (Ref. 12). As a consequence, the comparison of Table 7.7.1 on page 81 is incorrect. This is because the offered traffic in the example is a discrete step function, for which APP4 is derived, not a continuous, piecewise linear function, for which the formula (7.7.2) was derived.

7.7 Martikainen's Approximation

Another approximate method worth mentioning is due to Martikainen [11] His procedure is quite complicated but it can handle any continuous arrival or service rate functions, without the need to discretize them as in the case of other approximations. The only restriction is that the arrival rate cannot be zero because this will cause a singularity to occur.

The time-dependent blocking probability is given by

$$p_N(t, N) \approx \exp(d(t))(z(t))^N E_N(\rho(t)) \tag{7.7.1}$$

where

$$\begin{aligned}
\rho(t) &= \frac{\lambda(t)}{\mu(t)} \\
z(t) &= \frac{1}{\rho(t)}\Big(\int_0^t \exp\Big(-\int_u^t \mu(s)ds\Big)\rho(u)du + \rho(0)z(0)\exp\Big(-\int_0^t \mu(s)ds\Big)\Big) \\
d(t) &= \int_0^t (\lambda(u)(z(u)-1) - (1 - E_N(\rho(u))\dot{\rho}(u))du + d(0)
\end{aligned}$$

Note that we are using the real time axis, t, since no rescaling has been done. If the system has been in the steady state until $t = 0$, then $z(0) = 1$ and $d(0) = 0$.

For the case when the offered traffic is discrete, the approximation is

$$p_n(t, N) \approx \exp(\frac{a_i - a_{i-1}}{T} - \delta_i)(1 - \frac{a_i - a_{i-1}}{a_i T})^N E_N(a_i) \tag{7.7.2}$$

where a_i is the offered traffic in interval i, T is the length of each interval in units of average mean holding time, and

$$\delta_i \approx \sum_{k=1}^{i} E_N(\frac{|a_k - a_{k-1}|}{2})|a_k - a_{k-1}|$$

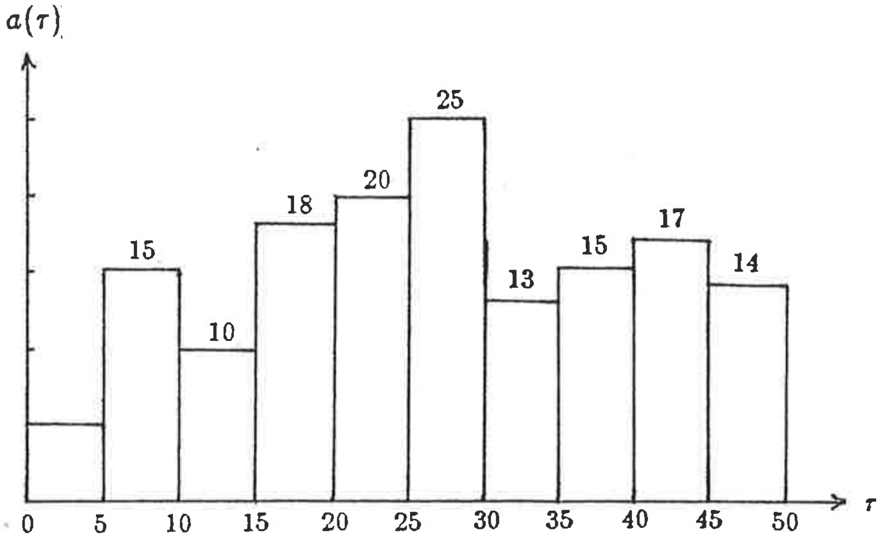

Figure 7.7.1 : Discrete offered traffic with $T = 5$.

TIME	EXACT	MART	ERROR	APP1	ERROR	APP4	ERROR
5.0	1.5198E-4	1.5040E-5	-0.0001	1.5726E-4	0.0000	1.4696E-4	0.0000
10.0	0.1807	0.1555	-0.0252	0.1803	-0.0004	0.1793	-0.0014
15.5	3.6681E-2	5.5995E-2	0.0193	3.6497E-2	-0.0002	3.6976E-2	0.0003
20.0	0.2752	0.3354	0.0602	0.2741	-0.0011	0.2738	-0.0014
25.5	0.3342	0.3630	0.0288	0.3300	-0.0042	0.3300	-0.0042
30.0	0.4311	0.6527	0.2216	0.4437	0.0126	0.4437	0.0126
35.5	0.1179	0.1318	0.0139	0.1159	-0.0020	0.1167	-0.0012
40.0	0.1823	0.1771	-0.0052	0.1803	-0.0020	0.1802	-0.0021
45.5	0.2467	0.2515	0.0048	0.2440	-0.0027	0.2439	-0.0028
50.0	1.7727E-5	1.9714E-3	0.0020	1.5039E-5	0.0000	1.8636E-5	0.0000

Table 7.7.1: $a(\tau)$ as in Figure 7.7.1, $N = 15$

The derivation of (7.7.2) assumes constant mean holding time and the term δ_i is negligible if $E_N(.) < 0.1$. Otherwise it should be taken into account. This expression is derived solely for practical appplications since (7.7.1) is not so useful in these applications. It is recommended that T be at least 5 units of mean service time. It cannot be applied to step-functions like the previous approximations. For the application of (7.7.2) the mean traffic over each subinterval of duration T is taken, and the congestion due to the k^{th} subinterval is approximated using the mean traffic in this subinterval and also in the previous subintervals. For this reason we use a different arrival pattern to test (7.7.2) as in Figure 7.7.1. Here we have discrete traffic consisting of the mean traffic in each subinterval of duration 5 units of mean service time. The results obtained using (7.7.2) are compared with results obtained using $E_N(a(t))$ and the modified offered load, α, which approximate blocking at the end of each time-interval. The results are set out in Table 7.7.1.

As can be seen approximations obtained using (7.7.2) are not as good as the other two approximations.

7.8 Discussion

This chapter discusses several approximate methods to evaluate the time-dependent blocking probability for the M(t)/M(t)/N loss system. Five of them are given and their performances are compared with numerical results. We find that overall the approximation using the modified offered traffic function, APP4, is the best, unless congestion is high, when the direct use of the offered traffic function in the Erlang Loss Function, APP1, gives the best approximation. Of course when the offered traffic function is very highly fluctuative this approximation might be surpassed by

APP4 in performance. Therefore it is recommended that APP4 be used at all times, even when APP1 might give a better result, because the advantage that APP1 has in this is too small.

CHAPTER VIII

TIME-DEPENDENT EXPECTED LOST TRAFFIC AND VARIANCE OF TRAFFIC OVERFLOWING FROM THE M(t)/M(t)/N LOSS SYSTEM

8.1 Introduction

The previous chapters discussed time-dependent blocking in teletraffic systems experiencing nonstationary arrivals and service times. Approximate methods to evaluate the time-dependent blocking probability are given together with a numerical procedure. We now proceed to another important problem in teletraffic: blocking on a secondary group of trunks with traffic overflowing from a primary group of trunks which are offered time-dependent arrival streams.

Consider a group of N trunks which are offered traffic of intensity $a(\tau)$, where τ is the transformed time. Traffic overflowing from this so-called primary group is then offered to a secondary group with an infinite number of trunks (see Figure 8.1.1). We are interested in the time-dependent expected lost traffic and the time-dependent variance of this overflow traffic. The expected lost traffic is the expected overflow traffic. A major assumption is that this is the same as the carried traffic on the infinite secondary group. It is also assumed that the variance of the lost traffic is the same as the variance of the carried traffic on the secondary group.

These moments are vital as we shall see later.

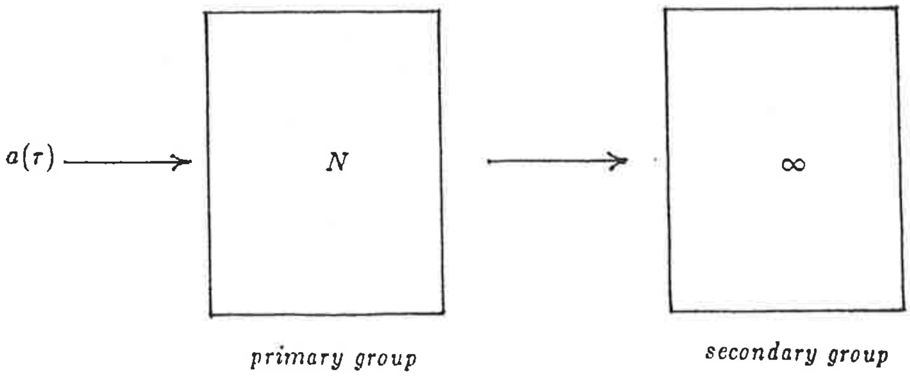

Figure 8.1.1 : A system with N primary trunks and infinite overflow trunks.

The theory for the case when the arrival stream to the primary group is of constant intensity, that is when the system is in steady-state, has been well-developed (see, for example, Wilkinson [25]) which is applied to the Equivalent Random Method (ERM). Since real-life traffic is not assumed to be in steady-state all the time, application of the ERM is not always valid. In the heart of the ERM are the expectation and the variance of the overflow traffic. ERM uses these two moments to find the *equivalent group* and the *equivalent traffic* which in turned are used to find the expectation and the variance of traffic lost from the secondary or overflow group.

Hence to develop a time-dependent scheme analogous to the ERM we need to know the time-dependent expected lost traffic and variance of the overflow traffic.

8.2 Time-Dependent Expected Lost Traffic

In the stationary case the expected lost traffic, or the mean lost traffic, is

$$E[lost\ traffic] = aE_N(a) \tag{8.2.1}$$

where a is the offered traffic. The analogous expression for the time-dependent case is

$$E[lost\ traffic\ at\ time\ \tau] = a(\tau)p_N(\tau) \tag{8.2.2}$$

Validity of the above expression (8.2.2) is supported by simulation, as will be shown. We let the service time be constant and normalize it for simplicity.

We are interested in the practical application of any method developed. We therefore test (8.2.2) with a simulation on two traffic patterns that would resemble real-life traffic. Since real-life traffic is discrete and could be represented as step-functions of

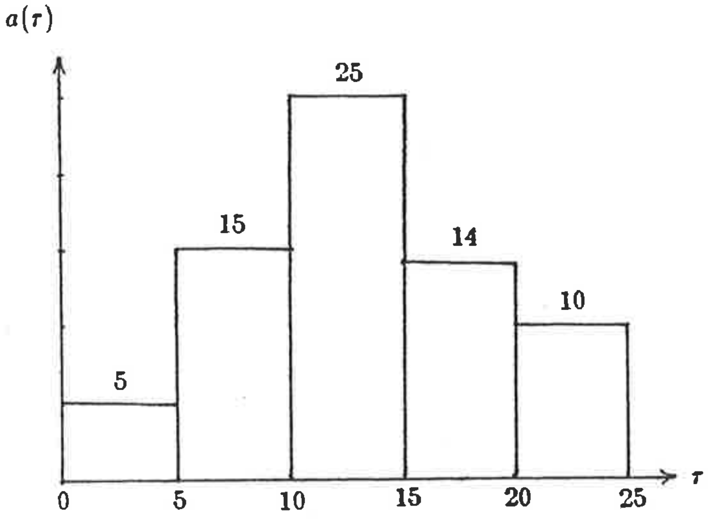

Figure 8.2.1 : Stream 1.

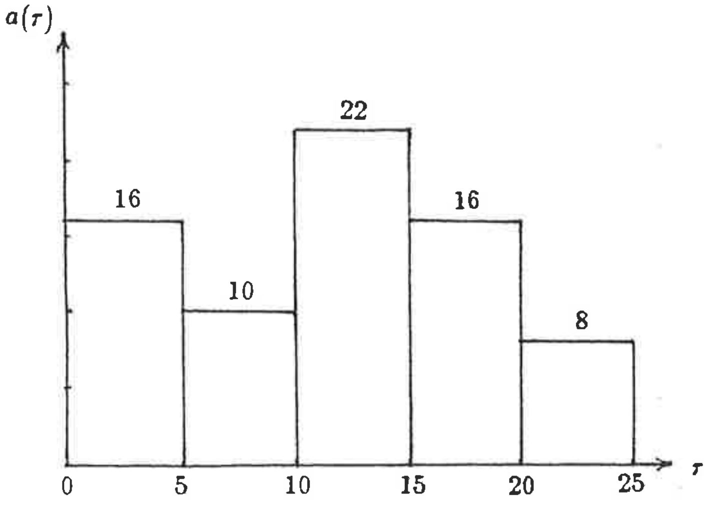

Figure 8.2.2 : Stream 2.

TIME	MEAN LOST CALLS/RUN	ARRIVAL x BLOCKING	APPROX
5.0	(0.0000,0.0000)	(0.0011,0.0020)	0.0007
10.0	(2.7939,2.9681)	(2.6327,2.7692)	2.6894
15.0	(11.2614,11.5036)	(11.0670,11.3281)	11.0911
20.0	(2.1414,2.2438)	(2.0187,2.1491)	2.0786
25.0	(0.3371,0.4466)	(0.3376,0.3766)	0.3690

Table 8.2.1 : Simulation results with traffic as in Figure 8.2.1, $N = 15$. *Confidence limit is 0.1.*

TIME	MEAN LOST CALLS/RUN	ARRIVAL x BLOCKING	APPROX
5.0	(3.2682,3.4531)	(3.3125,3.4366)	3.3778
10.0	(0.2894,0.3693)	(0.3614,0.3910)	0.3705
15.0	(8.3549,8.5938)	(8.3413,8.5299)	8.3509
20.0	(3.2977,3.4323)	(3.2548,3.3306)	3.4055
25.0	(0.0329,0.0521)	(0.0720,0.0876)	0.0757

Table 8.2.2 : Simulation results with traffic as in Figure 8.2.2, $N = 15$. *Confidence limit is 0.1.*

duration $\Delta\tau$, we choose the arrival patterns in Figures 8.2.1 and 8.2.2 for our simulation. We also let $N = 15$. Furthermore in real-life measurement, $\Delta\tau$ is expected to be of the order of the mean holding time, preferably at least 5 (Martikainen [11]). In fact in our simulation we have $\Delta\tau = 5$.

The simulation calculates the blocking probability at specific time τ's. It also counts the number of calls generated that are actually lost due to blocking at those times, which represents the left-hand side of (8.2.2). We also multiply the calculated blocking probability with the arrival rate at that particular time, and this quantity represents the right-hand side of (8.2.2). The two above quantities are set out in Tables 8.2.1 and 8.2.2 together with the approximated lost traffic obtained using the modified offered traffic scheme as outlined in Chapter VII. The simulations were done in eight batches of 10000 runs each with a confidence limit of 0.1.

As can be seen they agree quite well, which gives support to the validity of expression (8.2.2).

8.3 Time-Dependent Variance

In the steady-state case the derivation of the variance is a very lengthy exercise (Wilkinson [25]). In the time-dependent case there is as yet no method to derive the corresponding time-dependent variance. This is because of the existence of the derivatives of the probabilities which makes analysis very difficult. However we can get around this problem and use a very well-known formula, which is used to compute the steady-state variance of the overflow stream, if our arrival pattern consists of step-functions, with reasonably large time-interval, such as in Figures 8.2.1 and 8.2.2. This arrival pattern simulates real-life arrivals and so we not need worry ourselves with deriving variance for a continuous arrival pattern which is not only almost impossible but could also be of little use. Instead we are interested in the approximation of the *transient variance* which will occur on each subinterval. Fortunately an estimate of this variance can be obtained by using the well-known formula

$$v = m\left(1 - m + \frac{a}{m + N + 1 - a}\right) \tag{8.3.1}$$

where v is the variance, m is the expected lost traffic and N is the number of trunks in the primary group.

In the transient case (8.3.1) cannot be used directly to approximate the variance at time τ. This is because division by zero can occur, that is when $m + N + 1 - a = 0$. We have, for $\tau \in [0, \infty)$, $m \in [0, aE_N(a)]$. Since $a - N - 1 \in [0, aE_N(a)]$ is possible for $a > N$, we can have a singularity.

We now come to the problem of approximating the variance of the traffic overflowing from the primary group of trunks with offered traffic such as in Figures 8.2.1 and 8.2.2. These arrival patterns represent real-life traffic patterns and this simplifies our problem greatly. In both figures $\Delta\tau = 5$ which is quite low in real-life application. However as will be shown, even using such a low value our approximation scheme performs rather well. We would expect it to perform much better if $\Delta\tau$ is increased, as a state close to steady-state could be achieved.

For each subinterval, we note the arrival rate, λ. If $\lambda \geq N$ then we simply use (8.3.1) because at the end of this subinterval (of length 5, in our example) the system would be very close to steady-state. This is because traffic is heavy and thus steady-state is quickly achieved. On the other hand if $\lambda < N$ we say traffic is light and we expect steady-state to be achieved at a slower rate. In this case use (8.3.1) but with the modified offered traffic, α, computed at the end of the subinterval, instead of λ. Also in this case the effect from the previous subinterval is expected to be significant because of its slowness in achieving steady-state.

Letting $\alpha_k = \alpha(k\Delta\tau)$ be the modified offered traffic at time $\Delta\tau$ (using (3.4.1)), $m_k = m(k\Delta\tau)$ be the lost traffic at time $\Delta\tau$, and $v_k = v(k\Delta\tau)$ be the variance at time $\Delta\tau$, we have as an algorithm:

```
Step 0   Construct step-functions with $\Delta\tau \geq 5$ units of mean service time
         (assuming that it is stationary and normalized for simplicity).
         Set $k = 1$.
         Set $v_0 = 0$.
Step 1   For step-function $k$, note $\lambda_k$, the corresponding arrival rate.
         If $\lambda_k < N$ then
             $m_k = \lambda_k E_N(\alpha_k)$
             $v_k = m_k\left(1 - m_k + \frac{\alpha_k}{N+m_k+1-\alpha_k}\right) + v_{k-1}e^{-\Delta\tau}$.
         If $\lambda_k \geq N$ then
             $m_k = \lambda_k E_N(\lambda_k)$
             $v_k = m_k\left(1 - m_k + \frac{\lambda_k}{N+m_k+1-\lambda_k}\right)$.
         If all step-functions are completed then go to Step 2.
         $k = k + 1$.
         Go to Step 1.
Step 2   STOP.
```

The term $v_{k-1}e^{-\Delta\tau}$ is heuristically derived and it seems that it does give a good correction due to the variance of the previous subinterval which is quite significant when traffic is not heavy.

As mentioned, the above scheme was used on the test arrival pattern in Figures 8.2.1 and 8.2.2 and the approximations thus obtained are compared with simulation results (Tables 8.3.1 and 8.3.2). The simulations were done in eight batches of 10000 runs each with a confidence limit of 0.1.

We find that this approximate method performs satisfactorily.

8.4 Application to the Equivalent Random Method

In this section we attempt to approximate the expectation and variance of the traffic lost from the overflow link with nonstationary traffic being offered to the primary group of trunks. The nonstationary traffic in question is assumed to be in the form of Figures 8.2.1 and 8.2.2, which resemble real-life traffic patterns and the system we are considering is as in Figure 8.4.1. We have two primary groups, routes 1 and 2, and traffic overflowing from these two routes is offered to the overflow group, route 3. Traffic in Figure 8.2.1 is offered to route 1, and traffic in Figure 8.2.2 is offered to route 2. In the stationary case, the mean and the variance are obtainable via the Erlang Loss Function and the variance formula (8.3.1). However in the nonstationary case there is as yet no method available for the same purpose.

Using data from the previous sections, we construct Table 8.4.1. The expectation and variance are obtained using the approximate methods outlined before. Stream 1 denotes the traffic in Figure 8.2.1 and stream 2 denotes the traffic in Figure 8.2.2.

TIME	SIMULATION	APPROX
5.0	(0.0005,0.0010)	0.0011
10.0	(6.3505,6.3813)	6.3402
15.0	(19.7876,19.9841)	20.6127
20.0	(4.9451,5.2284)	4.9570
25.0	(0.7652,0.7938)	0.8090

Table 8.3.1 : Variance of overflow traffic with traffic as in Figure 8.2.1. Confidence limit is 0.1.

TIME	SIMULATION	APPROX
5.0	(7.9027,8.0476)	7.8332
10.0	(0.8591,0.8921)	0.8284
15.0	(16.5294,16.8100)	16.6309
20.0	(7.8602,8.0315)	7.8332
25.0	(0.1527,0.1621)	0.1854

Table 8.3.2 : Variance of overflow traffic with traffic as in Figure 8.2.2. Confidence limit is 0.1.

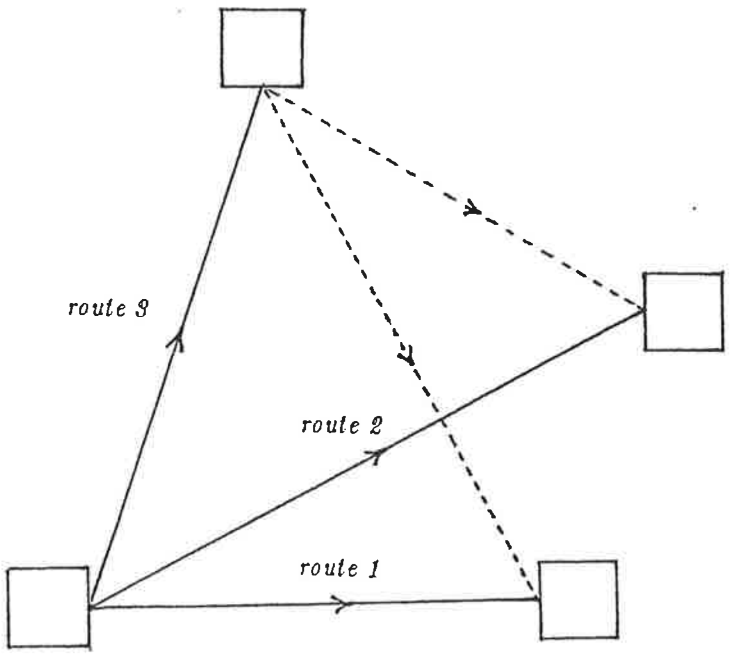

Figure 8.4.1 : A simple alternate routing network.

TIME		STREAM 1	STREAM 2	OVERFLOW
5.0	exp. lost	0.0007	3.3778	3.3785
	variance	0.0011	7.8332	7.8343
10.0	exp. lost	2.6894	0.3705	3.0599
	variance	6.3402	0.8284	7.1686
15.0	exp. lost	11.0911	8.3509	19.4420
	variance	20.6125	16.6309	37.2434
20.0	exp. lost	2.0786	3.4055	5.4841
	variance	4.9570	7.8332	12.7902
25.0	exp. lost	0.3690	0.0757	0.4447
	variance	0.8090	0.1854	0.9944

Table 8.4.1 : The two moments of the overflow stream; 'exp' stands for 'expected'.

TIME	λ_{eq}	N_{eq}
5.0	17.0092	16.2517
10.0	16.6059	16.3177
15.0	42.5053	24.1512
20.0	22.1114	18.8712
25.0	9.2867	13.3671

Table 8.4.2 : The equivalent offered traffic and number of trunks.

The expectation and variance of the overflow traffic are then added up, with the assumption that the two streams are indepedent, to obtain the total expectation and variance of the overflow stream.

To obtain the $N_{eq}(\tau)$, the equivalent number of trunks, and $\lambda_{eq}(\tau)$, the equivalent offered traffic, we use Rapp's formula and the inverted variance formula. We use them for this approximation because the variance formula was initially used in approximating the overflow variance. Rapp's formula and the inverted variance formula are, respectively,

$$\begin{aligned} \lambda_{eq} &= v + 3\frac{v}{m}\left(\frac{v}{m} - 1\right) \\ N_{eq} &= \frac{\lambda_{eq}\left(m + \frac{v}{m}\right)}{m + \frac{v}{m} - 1} - m - 1 \end{aligned} \tag{8.4.1}$$

where m is the expected lost traffic, and v is its variance.

Equations (8.4.1) were used to obtain Table 8.4.2. This means that the system where streams 1 and 2 are offered to the primary group of 15 trunks is equivalent to the system where the traffic offered and the number of trunks are as shown in Table 8.4.2. As expected both the traffic and the number of trunks in the equivalent group are time-dependent. These quantities can then be used to dimension the overflow link at specific times to maintain the specified grade-of-service.

Finally, as an illustration, we simulate the system with 15 trunks in the primary groups, and just 5 trunks in the overflow group. The traffic offered to the first primary group is stream 1 and the traffic offered to the second primary group is stream 2 as in Figure 8.4.2. The simulated expectation and variance of traffic lost from the overflow link are then compared to the results obtained when applying the

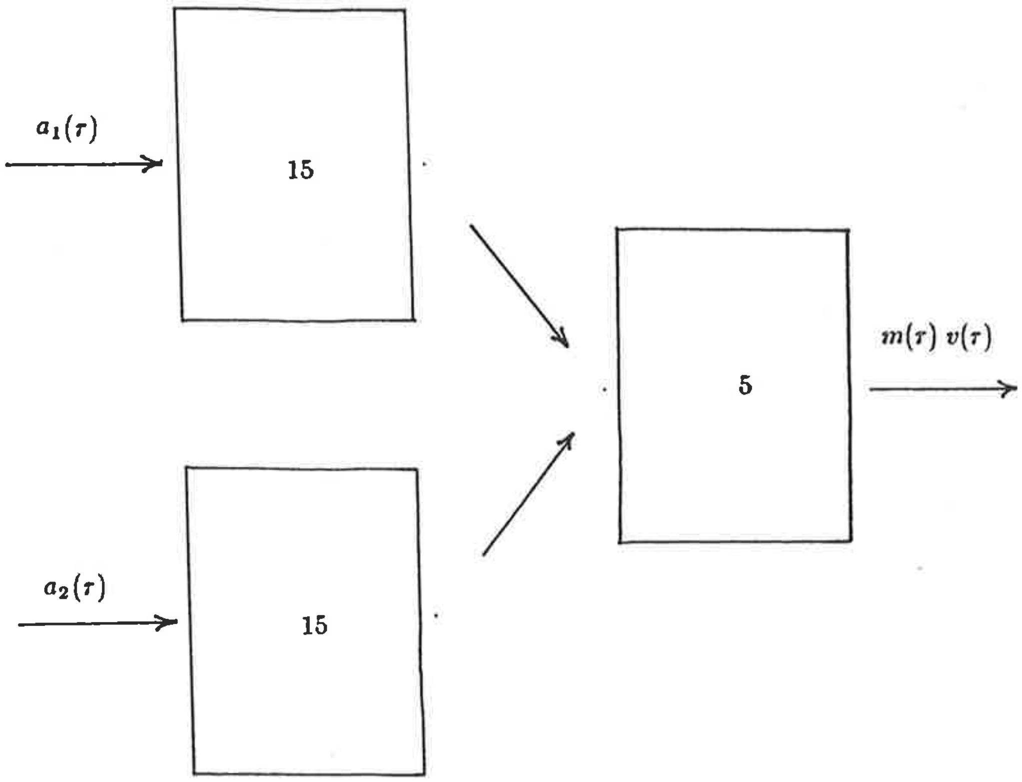

Figure 8.4.2 : The system considered in the simulation.

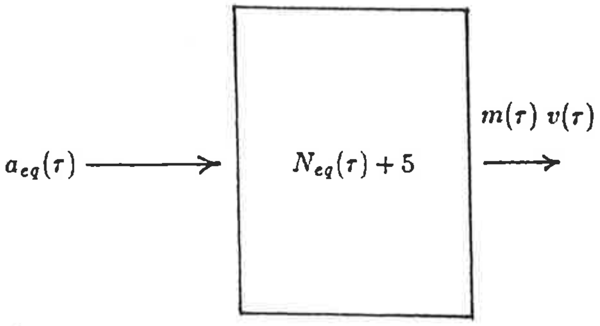

Figure 8.4.3 : The system 'equivalent' to the above system.

TIME	EXPECTATION	VARIANCE
5.0	(0.8857,0.9225)	(2.4383,2.5896)
10.0	(0.7845,0.8040)	(2.1292,2.3187)
15.0	(14.6403,14.7320)	(32.5772,33.5738)
20.0	(2.4014,2.4615)	(6.4271,6.8414)
25.0	(0.0454,0.0518)	(0.0727,0.0898)

Table 8.4.3 : Simulation results of the two moments of overflow traffic, as in Figure 8.4.2. Confidence level is 0.1.

TIME	EXPECTATION	VARIANCE
5.0	1.0742	2.6790
10.0	0.7680	2.2484
15.0	14.6336	34.2636
20.0	2.7614	6.7123
25.0	0.0286	0.0515

Table 8.4.4 : Approximations of the two moments of overflow traffic in the equivalent system, as in Figure 8.4.3.

approximation method to the equivalent system as in Figure 8.4.3. The data are set out in Tables 8.4.3 and 8.4.4. As can be seen from the table the approximations are quite consistent with the simulated values.

8.5 Discussion

This chapter discusses approximate methods for the expectation and variance of traffic overflowing from a primary group of trunks. These two approximations are then utilized in another approximate method for the equivalent offered traffic and number of trunks. These two quantities are important in dimensioning the overflow group to maintain the specified grade-of-service. The above approximations are quite rough but probably the only ones available so far.

Actually, in the problem of satisfying the grade-of-service in an alternate routing network with time-dependent arrival streams, we have at least two choices. Either dimension the primary groups dynamically or just dimension the common overflow group dynamically. The decision on which strategy to adopt will depend on other factors which still need to be researched.

CHAPTER IX

CONCLUSIONS AND FURTHER INVESTIGATIONS

This work outlines research on the problem of time-dependent blocking in teletraffic systems. It is heavily-oriented towards practical applications and hence is not mathematically very rigorous. However, before dealing directly with this problem, the somewhat simpler problem of transient blocking in similar systems is investigated. An exact procedure for evaluating the transient blocking probability in the M/M/N loss system is given. It is envisaged that this procedure would be helpful not just in teletraffic theory, but also in other areas of research where loss queueing systems are modelled. It is the only analytic method, apart from Riordan's method [23], to find the blocking probability in the M/M/N loss system. Furthermore it is easier to use than Riordan's method because it doesn't involve Laplace transforms. It is also better in the sense that it can take into account any initial condition (blocking).

The same technique could also be extended to delay queueing systems, which are gaining more importance nowadays. The use of matrix analysis to solve birth-and-death equations is not new, but with new developments in this area, its role could be more important. The problem of finding $\exp(\boldsymbol{A}(t))$ exactly where $\boldsymbol{A}(t)$ is a time-dependent square matrix is a very hard one. However success in this problem will greatly aid in solving queueing systems which will result in more accurate solutions. Other areas, such as control theory, will also benefit.

Approximate methods are useful when exact methods are not available, or if the ones available are not suitable for practical purposes. This could be due to their excessive computer time and/or space requirements. Approximate methods on the other hand are easy to use and quick. Several approximate methods for evaluating transient blocking probability in loss systems are given and shown to be feasible for practical applications. These methods are then extended to the real-life problem of loss systems with time-dependent arrival and service times. A simple discretization method is applied to the arrival and service rate functions in order to use the methods developed for the stationary systems on the nonstationary systems. Approximations of the blocking probability using these approximate methods are very encouraging. They are quite feasible for practical application.

Using the above approximate methods, approximations for the time-dependent expectation and variance of traffic overflowing from primary groups can be obtained. These approximations are quite rough but they are still adequate for practical applications. Furthermore they are the only ones available so far. It is hoped that these approximations can be improved in future research. Using these moments of the overflow traffic, links can be dynamically dimensioned so as to satisfy the grade-of-service for specific time-dependent arrivals.

REFERENCES

[1] Bellman, R., *Introduction to Matrix Analysis* (McGraw-Hill, New York, 1972).

[2] Berry, L.T.M., "On the Definition of Offered Traffic", *Aust. Telecom. Res.* 14 (1980), 35-38.

[3] Coyle, A.J. and Yunus, M.N., "A Numerical Method to Evaluate Blocking Probability in the M(t)/M/N Loss System", *to appear in the Aust. Telecom. Res.*.

[4] Csaki, F. G., "Some Notes on the Inversion of Confluent Vandermonde Matrices", *IEEE Trans. Automatic Control* AC-20 (1975), 154-157.

[5] Farmer, R.F. and Kaufman, I., "On the Numerical Evaluation of Some Basic Traffic Formulae", *Networks* 8 (1978), 153-186.

[6] Gnedenko, B.V. and Kovalenko, I.N., *Introduction to Queueing Theory* (Israel Program for Scientific Translations Ltd., Jerusalem, 1968).

[7] Jagerman, D.L., "Some Properties of the Erlang Loss Function ", *BSTJ* 53 (1974), 525-551.

[8] Jagerman, D.L., "Nonstationary Blocking in Telephone Traffic ", *BSTJ* 54 (1975), 625-661.

[9] Khintchine, A.Y., *Mathematical Methods in the Theory of Queueing* (Griffin and Co. Ltd., London, 1960).

[10] Lehtinen, P. and Martikainen, O., "Development of Traffic Theory for the SPC Exchanges and New Teleservices Based on Measured Data", *Proc. ITC-10, Montreal* (1983).

[11] Martikainen, O., "Time-Varying Blocking Probability of Nonstationary Traffic", *private communication.*

[12] Martikainen, O. and Lehtinen, P., "Calculation of Time-Varying Blocking Probability on the Basis of Measured Traffic", *private communication.*

[13] Martin, R.S. and Wilkinson, J.H., "The Implicit QL Algorithm", *Numer. Math.* 12 (1968), 377-383.

[14] Mejzler, D., "A Generalization of Erlang's Formulas in Queueing Theory", *J. Appl. Prob.* 5 (1968), 143-157.

[15] Miwa, E., "Time-Varying Traffic and Congestions in Telephone Exchange Systems", *IECE Japan* 49 (1966), 60-68.

[16] Moler, C.B. and van Loan, C.F., "Nineteen Dubious Ways to Compute the Exponential of a Matrix", *SIAM Review* 20 (1978), 801-836.

[17] Myskja, A.H., *A Study of Telephone Traffic Arrival Processes with Focus on Nonstationary Case* (Ph.D. Thesis, University of Trondheim, 1981).

[18] Myskja, A.H., "Modelling of Nonstationary Traffic Processes ", *Proc. ITC-9, Torremolinos* (1979).

[19] Myskja, A.H., "Distribution Models for Quasi-stationary Random Traffic", *Proc. ITC-10, Montreal* (1983).

[20] Ohsawa, J., Fujihashi, C. and Gambe, E., "Traffic Characteristics of a Full Availability Lost-Call System with Offered Load Time Variation", *Trans. IECE Japan* J61-B (1977), 17-23, *reviewed in Trans. IECE Japan* E61 (1978), 32.

[21] Palm, C., "Intensitätsschwankungen im Fernsprechverkehr", *Ericsson Techniques* 44 (1943), 1-189.

[22] Reid, J. G., *Linear System Fundamentals* (McGraw-Hill, New York, 1983).

[23] Riordan, J., *Stochastic Service Systems* (Wiley, New York, 1962).

[24] Syski, R., *Introduction to Congestion Theory in Telephone System* (Oliver and Boyd, Edinburgh, 1960).

[25] Wilkinson, R.I., "Theories for Toll Traffic Engineering in the U.S.A.", *BSTJ* 35 (1956), 421-514.

[26] Wu, M.Y., "Solution of Certain Classes of Linear Time-Varying Systems", *Int. J. Control* 31 (1980), 11-20.

[27] Yunus, M.N., "A Method of Determining Blocking Probability in the M(t)/M(t)/1 Loss System", *Aust. Telecom. Res.* 18 (1984), 13-17.

[28] Yunus, M.N., "Determination of Time-Dependent Blocking Probability for the M/M/N Loss System", *to appear in the Proc. 3rd. Franco-Southeast Asian Math. Conf., Kuala Lumpur, 1985.*

APPENDICES

This is the Fortran code of the program used to compute time-dependent blocking probability numerically as described in Chapter V. The algorithm used is the implicit QL algorithm (Martin and Wilkinson [13]).

```
.       PROGRAM QL
.       PARAMETER NN=101
.       DIMENSION D(NN),E(NN),Z(NN,NN),X(NN),Y(NN),XMIN(100)
.       IZ=NN
1000    PRINT*,'input no. of circuits'
.       READ*,N
.       PRINT*,'no. of time steps per sec.'
.       READ*,NDEL
.       DELT=1.0/FLOAT(NDEL)
.       PRINT*,'How long do you want the program to run for?'
.       READ*,TLEN
.       PRINT*,'How often do you want results printed out?'
.       READ*,TRES
.       NPRINTS=TLEN/TRES
.       NITER=NDEL*TRES
.       PRINT*,'Do you want the results printed onto FOR001.DAT?'
.       PRINT*,'Print 1 if you do'
.       READ*,PR
.       PRINT*
.       PRINT*,'no. of circuits =',NCCT
.       PRINT*,'time steps per. sec. =',NDEL
.       PRINT*,'delta t, size of each time step =', DELT
.       PRINT*,'Results printed out every',TRES,'seconds'
.       PRINT*
.       PRINT*,'  time        min. cong.       max. cong.'
.       IF (PR.EQ.1) THEN
.       WRITE(1,*)
.       WRITE(1,*),'no. of circuits =',NCCT
.       WRITE(1,*),'time steps per. sec. =',NDEL
.       WRITE(1,*),'delta t, size of each time step =', DELT
.       WRITE(1,*),'Results printed out every',TRES,'seconds'
.       WRITE(1,*)
.       WRITE(1,*),'  time        min. cong.       max. cong.'
.       ENDIF
.       N=NCCT+1
.       DO 120 NMAX=1,2
.       ERR=0.0
.       T=-DELT/2.0
.       DO 40 I=2,N
40      X(I)=0.0
.       X(1)=1.0
.       DO 110 NIT=1,NPRINTS
```

```
.      DO 100 NNN=1,NITER
.      T=T+DELT
.      XLAM1=XLAMBDA(T+DELT/2.0)
.      XLAM2=XLAMBDA(T-DELT/2.0)
.      IF (NMAX.EQ.2) THEN
.        XLAM=AMAX1(XLAM1,XLAM2)
.      ELSE
.        XLAM=AMIN1(XLAM1,XLAM2)
.      ENDIF
.      DO 5 I=1,N-1
.      D(I)=-XLAM-(I-1.0)
5     E(I+1)=SQRT(I*XLAM)
.      DO 7 I=1,N
.      DO 6 J=1,N
6     Z(I,J)=0.0
7     Z(I,I)=1.0
.      D(N)=-(N-1.0)
C
C      EQRT2S is an IMSL subroutine used to find the
C      eigenvalues and (optionally) the eigenvectors
C      of a symmetric tridiagonal matrix  using  the
C      the QL algorithm.
C
.      CALL EQRT2S(D,E,N,Z,IZ,IER)
.      IF (IER.NE.0) PRINT*,'ERROR'
.      PROD=1.0
.      DO 21 I=2,N
.      PROD=PROD*SQRT(FLOAT(I-1))
21    X(I)=X(I)*PROD/((SQRT(XLAM))**(I-1))
.      DO 22 I=1,N
.      Y(I)=0.0
.      DO 22 J=1,N
22    Y(I)=Y(I)+Z(j,i)*X(J)
.      DO 23 I=1,N
23    Y(I)=Y(I)*EXP(D(I)*DELT)
.      DO 24 I=1,N
.      X(I)=0.0
.      DO 24 J=1,N
24    X(I)=X(I)+Z(i,j)*Y(J)
.      PROD=1.0
.      DO 25 I=2,N
.      PROD=PROD*SQRT(FLOAT(I-1))
25    X(I)=X(I)*(SQRT(XLAM))**(I-1)/PROD
100   CONTINUE
.      IF (NMAX.EQ.1) XMIN(NIT)=X(N)
.      IF (NMAX.EQ.2) PRINT*,T,XMIN(NIT),X(N)
.      IF (NMAX.EQ.2.AND.PR.EQ.1) WRITE(1,*),T,XMIN(NIT),X(N)
110   CONTINUE
```

```
120     PRINT*
.        PRINT*,'Do you want to continue with this present function?'
.        PRINT*,'Type 1 if you do.'
.        READ*,ANS
.        IF (ANS.EQ.1) GOTO 1000
.        STOP
.        END
C
C       The following function is the arrival rate function. In this
C       case the arrival rate is λ(t) = t. This function can be
C       altered as required.
C
FUNCTION XLAMBDA(T)
XLAMBDA=T
RETURN
END
```

This is the Fortran code of the program used to simulate blocking as used in the text.

```
.       PROGRAM SIMULATION
.       PARAMETER (NLINKS=12,NARSTR=12,MEVS=3000,
.    *          NCHS=3,NLPCH=2)
.       COMMON/SCHED/EVTIME(MEVS),IEVTYPE(MEVS),ITOP
.       COMMON/RAND/ISEED
.       COMMON/ROUTES/IR(NARSTR,NCHS,NLPCH)
.       COMMON/CCTS/N(NLINKS),NU(NLINKS)
.       COMMON/FLOW/F,NN
.       DIMENSION AR(NARSTR),IOFF(NARSTR),LOST(NARSTR),ISD(8),
.    *         TLARR(NARSTR),STM(NARSTR),TOTLOST(NARSTR),
.    *         NARR(NARSTR,NCHS),PARR(NARSTR,NCHS),
.    *         PLOST(NARSTR),TOTVAR(NARSTR),VAR(NLINKS),
.    *         VARSUM(NLINKS),XLMEAN(NLINKS),
.    *         BTIME(10),PBLOCK(10),NAV(10),NLO(10),
.    *         VART(10),VARTSQ(10)
.       DATA ISD/8913,7347,9825,5569,6571,9123,8765,7001/
.       DATA BTIME/2.5,5.0,7.5,10.0,12.5,15.0,17.5,20.0,22.5,25.0/
.       READ*,IARSTR
.       READ*,LINKS
.       PRINT*,'NARSTR',IARSTR,'LINKS',LINKS
.       DO 1000 I=1,LINKS
.       READ*,N(I)
.       VARSUM(I)=0.0
1000    PRINT*,N(I)
.       DO 999 I=1,IARSTR
.       READ*,NI
.       DO 999 J=1,NI
.       NARR(I,J)=0
.       PARR(I,J)=0.0
.       READ*,NIJ
.       DO 999 K=1,NIJ
.       READ*,IR(I,J,K)
.       PRINT*,I,J,K,IR(I,J,K)
999     CONTINUE
.       DO 9999 NTIMES=1,1
.       DO 7778 I=1,IARSTR
.       TOTLOST(I)=0.0
.       TOTVAR(I)=0.0
.       DO 7778 J=1,NCHS
.       NARR(I,J)=0
7778    PARR(I,J)=0.0
.       DO 1010 I=1,IARSTR
.       READ*,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10
```

```
1010  PRINT*,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10
.     READ*,NRUNS
.     PRINT*,'NRUNS=',NRUNS
.     PSUM=0.0
.     CSUM=0.0
.     DO 978 II=1,10
.     VART(II)=0.0
.     VARTSQ(II)=0.0
.     NAV(II)=0
.     NLO(II)=0
978   PBLOCK(II)=0.0
.     NRUN=0
.     NRUNC=0
.     JSEED=ISD(1)
.     DO 2000 NRUN1=1,NRUNS
.     DO 2001 NRUN2=1,50
.     NRUN=NRUN+1
.     PTIME=0.0
.     PTOT=0.0
.     CTIME=0.0
.     PCON=0.0
.     ISEED=RAN(JSEED)*10000
.     ITOP=0
.     DO 2010 I=1,LINKS
.     VAR(I)=0.0
.     XLMEAN(I)=0.0
2010  NU(I)=0
.     TIMENOW=0.0
.     TIMELAST=0.0
.     NO1=1
.     DO 2020 I=1,IARSTR
.     TIME=POISSARR(TIMENOW,A1)
.     CALL ADDEVENT(TIME,I)
.     STM(I)=0.0
.     IOFF(I)=0
2020  LOST(I)=0
113   CONTINUE
.     IF (TIMENOW.GE.0.0.AND.TIMENOW.LT.2.5) A=A1
.     IF (TIMENOW.GE.2.5.AND.TIMENOW.LT.5.0) A=A2
.     IF (TIMENOW.GE.5.0.AND.TIMENOW.LT.7.5) A=A3
.     IF (TIMENOW.GE.7.5.AND.TIMENOW.LT.10.) A=A4
.     IF (TIMENOW.GE.10..AND.TIMENOW.LT.12.5) A=A5
.     IF (TIMENOW.GE.12.5.AND.TIMENOW.LT.15.0) A=A6
.     IF (TIMENOW.GE.15..AND.TIMENOW.LT.17.5) A=A7
.     IF (TIMENOW.GE.17.5.AND.TIMENOW.LT.20.0) A=A8
.     IF (TIMENOW.GE.20..AND.TIMENOW.LT.22.5) A=A9
.     IF (TIMENOW.GE.22.5.AND.TIMENOW.LE.25.0) A=A10
.     CALL NEXTEVENT(TIMENOW,ITYPE)
```

```
.      DO 133 NF=1,10
.      IF(TIMENOW.GT.BTIME(NF).AND.TIMELAST.LT.BTIME(NF)) THEN
.        IF(NU(1).EQ.N(1)) PBLOCK(NF)=PBLOCK(NF)+1.0
.        VART(NF)=VART(NF)+NU(2)
.        VARTSQ(NF)=VARTSQ(NF)+NU(2)**2
.        IF(NF.EQ.1) NRUNC=NRUNC+1
.      ENDIF
133    CONTINUE
.      IF (ITYPE.GT.99) GOTO 50
.      DO 233 NF=1,10
.      IF(TIMENOW.GT.BTIME(NF)-0.01.AND.
.    *   TIMENOW.LT.BTIME(NF)+0.01) THEN
.           NAV(NF)=NAV(NF)+1
.           IF(NU(1).EQ.N(1)) NLO(NF)=NLO(NF)+1
.      ENDIF
233    CONTINUE
.      TIMELAST=TIMENOW
.      TLARR(ITYPE)=TIMENOW
.      IOFF(ITYPE)=IOFF(ITYPE)+1
.      IF (IOFF(ITYPE).EQ.1) STM(ITYPE)=TIMENOW
.      TIME=POISSARR(TIMENOW,A)
.      CALL ADDEVENT(TIME,ITYPE)
.      CALL WHATCHAIN(ITYPE,I)
.      IF (I.NE.0) GOTO 22
.      LOST(ITYPE)=LOST(ITYPE)+1
.      GOTO 100
22     J=0
.      NARR(ITYPE,I)=NARR(ITYPE,I)+1
25     J=J+1
.      LINK=IR(ITYPE,I,J)
.      IF(LINK.EQ.0) GOTO 30
.      NU(LINK)=NU(LINK)+1
.      IF(J.LT.NLPCH)GOTO 25
30     TIME=POISSARR(TIMENOW,1.0)
.      ITYPE=ITYPE+100*I
.      CALL ADDEVENT(TIME,ITYPE)
.      GOTO 100
50     TIMELAST=TIMENOW
.      X=FLOAT(ITYPE)/100.0
.      I=INT(X)
.      ITYPE=ITYPE-I*100
.      J=0
55     J=J+1
.      LINK=IR(ITYPE,I,J)
.      IF (LINK.EQ.0) GOTO 100
.      NU(LINK)=NU(LINK)-1
.      IF (J.LT.NLPCH) GOTO 55
100    CONTINUE
```

```
.       IF (TIMENOW.LT.BTIME(10)+0.1) GOTO 113
2001    CONTINUE
2000    CONTINUE
9999    CONTINUE
.       TEMP=FLOAT(NRUN)
.       PRINT*,'NO. OF RUNS',NRUN,NRUNC
.       DO 9998 II=1,10
.       PB=PBLOCK(II)/TEMP
.       XNA=FLOAT(NAV(II))*50.0/TEMP
.       XNL=FLOAT(NLO(II))*50.0/TEMP
.       CHECK=PB*XNA
.       PRINT*
.       PRINT*,'AT TIME',BTIME(II),'PBLOCK =',PB
.       PRINT*,'AND ARRIV., LOST =',XNA,XNL
.       PRINT*,'CHECK LOST',CHECK
.       VT=VART(II)/TEMP
.       VTSQ=VARTSQ(II)/TEMP
.       PRINT*,'E(X) , E(X**2)',VT,VTSQ
.       VTSQ=(VTSQ-VT**2)
.       NN=N(1)
.       PRINT*,'SIMULATED VARIANCE IS ',VTSQ
9998    CONTINUE
.       STOP
.       END
.
.       SUBROUTINE WHATCHAIN(ITYPE,I)
.       PARAMETER (NARSTR=12,NCHS=3,NLPCH=2,NLINKS=12)
.       COMMON/ROUTES/IR(NARSTR,NCHS,NLPCH)
.       COMMON/CCTS/N(NLINKS),NU(NLINKS)
.       I=0
10      I=I+1
.       IF(IR(ITYPE,I,1).EQ.0) GOTO 30
.       J=0
20      J=J+1
.       IF(J.GT.NLPCH) RETURN
.       LINK=IR(ITYPE,I,J)
.       IF(LINK.EQ.0) RETURN
.       IF(NU(LINK).LT.N(LINK)) GOTO 20
.       IF(I.LT.NCHS.AND.J.NE.2) GOTO 10
30      I=0
.       RETURN
.       END
.
.       FUNCTION POISSARR(TIMENOW,A)
.       COMMON/RAND/ISEED
.       R=RAN(ISEED)
.       POISSARR=TIMENOW-ALOG(1.0-R)/A
.       RETURN
```

```
.      END
.
.      SUBROUTINE ADDEVENT(TIME,ITYPE)
.      PARAMETER (MEVS=3000)
.      COMMON/SCHED/EVTIME(MEVS),IEVTYPE(MEVS),ITOP
.      J=0
10    J=J+1
.      IF((TIME.GE.EVTIME(J)).AND.(J.LE.ITOP))GOTO 10
.      IF(J.EQ.ITOP+1) GOTO 30
.      DO 20 I=J,ITOP
.      K=ITOP+J-I
.      EVTIME(K+1)=EVTIME(K)
20    IEVTYPE(K+1)=IEVTYPE(K)
30    EVTIME(J)=TIME
.      IEVTYPE(J)=ITYPE
.      ITOP=ITOP+1
.      RETURN
.      END
.
.      SUBROUTINE NEXTEVENT(TIME,ITYPE)
.      PARAMETER (MEVS=3000)
.      COMMON/SCHED/EVTIME(MEVS),IEVTYPE(MEVS),ITOP
.      TIME=EVTIME(1)
.      ITYPE=IEVTYPE(1)
.      DO 10 I=2,ITOP
.      EVTIME(I-1)=EVTIME(I)
10    IEVTYPE(I-1)=IEVTYPE(I)
.      ITOP=ITOP-1
.      RETURN
.      END
.
.      FUNCTION EN(NN,A)
.      EN=1.0
.      DO 20 I=1,NN
20    EN=A*EN/(I+A*EN)
.      RETURN
.      END
```

This is the Fortran code of the program used to approximate time-dependent blocking probability numerically as described in Section 7.5, using the modified offered traffic, $\alpha(t)$.

```
.      PROGRAM MODIFIEDERLANG
C
C     The offered traffic function is the following
C     function statement.
C
.      XL(T)=10.+5.0*SIN(0.4*4.0*ATAN(1.0)*T)
C
.      EPS=1.0E-6
.      WRITE(6,*),'N'
.      READ*,N
.      WRITE(6,*),'DELT'
.      READ*,DELT
.      WRITE(6,*) ,'TIME INTERVAL FOR PRINTING'
.      READ*,DT
.      WRITE(6,*) ,'FINAL TIME'
.      READ*,TF
.      ND=INT(DT/DELT)
.      WRITE(6,*),'INPUT INITIAL TRAFFIC'
.      READ*,A0
.      KK=INT(TF/DELT)
.      KL=INT(KK/FLOAT(ND))
.      DO 55 I=1,KL
.      DO 56 K=ND*(I-1)+1,ND*I
.9    T=K*DELT
.      X=XL(T-DELT/2.0)
.      P=EN(T,N)
.      J=1
.10   A=X*(1.0-EXP(DELT/(P-1.0)))+A0*EXP(DELT/(P-1.0))
.      PR=EN(A,N)
.      DIFF=ABS(P-PR)
.      IF (DIFF.LE.EPS) GOTO 51
.      P=PR
.      J=J+1
.      GOTO 10
.51   A0=A
.56   CONTINUE
.      PRINT*,'AT TIME ',T,'BLOCKING PROB. IS',PR,
.    *        'MODIFIED TRAFFIC IS',A
.      PRINT*,'EXPECTED LOST IS',PR*X
.55   CONTINUE
.99   STOP
.      END
```

```
.
.      FUNCTION EN(A,N)
.      X=1
.      DO 60 I=1,N
.        X=A*X/(I+A*X)
.60    CONTINUE
.      EN=X
.      RETURN
.      END
```

Yunus, M.N., (1984) A method of determining blocking probability in the M(t/M(t)1
loss system.
Australian Telecommunication Research, v. 18 (2), pp. 13-17.