



1-1-1997

# Two Parallel Finite Queues with Simultaneous Services and Markovian Arrivals

Srinivas R. Chakravarthy

*Kettering University*, [schakrav@kettering.edu](mailto:schakrav@kettering.edu)

S. Thiagarajan

*Kettering University*

Follow this and additional works at: [https://digitalcommons.kettering.edu/industrialmanuf\\_eng\\_facultypubs](https://digitalcommons.kettering.edu/industrialmanuf_eng_facultypubs)



Part of the [Industrial Engineering Commons](#)

---

## Recommended Citation

Chakravarthy, Srinivas R. and Thiagarajan, S., "Two Parallel Finite Queues with Simultaneous Services and Markovian Arrivals" (1997). *Industrial & Manufacturing Engineering Publications*. 98.

[https://digitalcommons.kettering.edu/industrialmanuf\\_eng\\_facultypubs/98](https://digitalcommons.kettering.edu/industrialmanuf_eng_facultypubs/98)

This Article is brought to you for free and open access by the Industrial & Manufacturing Engineering at Digital Commons @ Kettering University. It has been accepted for inclusion in Industrial & Manufacturing Engineering Publications by an authorized administrator of Digital Commons @ Kettering University. For more information, please contact [digitalcommons@kettering.edu](mailto:digitalcommons@kettering.edu).

# TWO PARALLEL FINITE QUEUES WITH SIMULTANEOUS SERVICES AND MARKOVIAN ARRIVALS<sup>1</sup>

S.R. CHAKRAVARTHY and S. THIAGARAJAN

*GMI Engineering & Management Institute*

*Department of Science and Mathematics, Flint, MI 48504-4898 USA*

(Received May, 1996; Revised January, 1997)

In this paper, we consider a finite capacity single server queueing model with two buffers, A and B, of sizes K and N respectively. Messages arrive one at a time according to a Markovian arrival process. Messages that arrive at buffer A are of a different type from the messages that arrive at buffer B. Messages are processed according to the following rules: 1. When buffer A(B) has a message and buffer B(A) is empty, then one message from A(B) is processed by the server. 2. When both buffers, A and B, have messages, then two messages, one from A and one from B, are processed simultaneously by the server. The service times are assumed to be exponentially distributed with parameters that may depend on the type of service. This queueing model is studied as a Markov process with a large state space and efficient algorithmic procedures for computing various system performance measures are given. Some numerical examples are discussed.

**Key words:** Markov Chains, Markovian Arrival Process (MAP), Renewal Process, Queues, Finite Capacity, Exponential Services and Algorithmic Probability.

**AMS subject classifications:** 60K20, 60K25, 90B22, 60J27, 60K05, 60K15.

## 1. Introduction

We consider a finite capacity single server queueing model with two buffers, say A and B, of sizes K and N, respectively. Messages arrive one at a time according to a Markovian arrival process (MAP). Messages that enter buffer A are possibly of a different type from those entering buffer B and hence are processed differently by the server. We shall refer to messages that arrive at buffer A(B) as type A(B) messages. Messages that enter the two buffers are processed according to the following rules.

---

<sup>1</sup>This research was supported in part by Grant No. DMI-9313283 from the National Science Foundation.

a) When buffer A(B) has a message and buffer B(A) is empty, then one message from A(B) gets processed by the server and the service time is assumed to be exponentially distributed with parameter  $\mu_A(\mu_B)$ .

b) When both buffers, A and B are not empty, then one type-A message and one type B-message are processed simultaneously by the server and the service time is assumed to be exponentially distributed with parameter  $\mu_{AB}$ .

c) When the buffers are empty, the server waits for the first arrival of a message.

We mention some potential applications of our model.

1. In multi-task operating systems, tasks are usually classified according to their characteristics, and separate queues serviced by different schedulers are maintained. This approach is called *multiple-queue scheduling*. A task may be assigned to a specific queue on the basis of its attributes, which may be user or system supplied. In the simplest case, the CPU is capable of supporting two active tasks (either user or system supplied) simultaneously. If the CPU is busy with both user and system tasks, then the service times (which can be assumed to be exponential with parameter  $\mu_{AB}$ ) are different from the service times (which can be assumed to be exponential with parameters  $\mu_A$  or  $\mu_B$ ) when the CPU is running only one task from either of the queues.

2. In communication systems, there are common transmission lines which transmit messages from more than one data source. In the simplest case, a minimum of two data sources, say A and B, may use a transmission line. If both data sources are using the transmission line simultaneously, then the service times can be assumed to be exponential with parameter  $\mu_{AB}$ . Otherwise, if the channel is being used by only one data source at a time, either A or B, then the service time by the channel can be assumed to be exponential with parameter  $\mu_A$  or  $\mu_B$ , respectively.

3. In transportation systems, we may face the following situation. A factory manufactures two different types of items, say  $I_A$  and  $I_B$ .  $I_A$  and  $I_B$  are transported by a truck to warehouses A and B respectively. Assume that the production of items  $I_A$  and  $I_B$  occurs independently of each other. If, at any time, both items are waiting to be transported, then the truck has to go to both warehouses A and B, and the delivery times can be assumed to be exponentially distributed with parameter  $\mu_{AB}$ . If, however, only one type of item, either  $I_A$  or  $I_B$ , awaits transportation, then the truck needs to go to either A or B and the delivery times can be assumed to be exponentially distributed with parameter  $\mu_A$  or  $\mu_B$ , depending on the item to be transported.

This paper is organized as follows. A brief description of the MAP is given in Section 2. The Markov chain description of the model is presented in Section 3 and the steady state analysis of the model is outlined in Section 4. Numerical examples are presented in Section 5 and concluding remarks are given in Section 6.

## 2. Markovian Arrival Process

The MAP, a special class of tractable Markov renewal processes, is a rich class of point processes that includes many well-known processes. One of the most significant features of the MAP is the underlying Markovian structure and its ideal fit into the context of the matrix-analytic solutions to stochastic models. Matrix-analytic

methods were first introduced and studied by Neuts [3]. The MAP significantly generalizes the Poisson processes but still keeps the tractability of the Poisson processes for modeling purposes. Furthermore, in many practical applications, notably in communications, production and manufacturing engineering, the arrivals do not usually form a renewal process. So, MAP is a convenient tool which can be used to model both renewal and nonrenewal arrivals. In addition, MAP is defined for both discrete and continuous times, and also for single and group arrivals. Here we shall need only the case of single arrivals in continuous time. For further details on MAP, the reader is referred to Lucantoni [2] and Neuts [3].

The MAP with single arrivals in continuous time is described as follows. Let the underlying Markov chain be irreducible and let  $Q = (q_{i,j})$  be the generator of this Markov chain. At the end of a sojourn time in state  $i$ , which is exponentially distributed with parameter  $\lambda_i \geq -q_{i,i}$ , one of the following events could occur: (i) an arrival time of type A could occur with probability  $p_{i,j}^A$  and the underlying Markov chain is in state  $j$  with  $1 \leq I$  and  $j \leq m$ ; (ii) an arrival of type B could occur with probability  $p_{i,j}^B$  and the underlying Markov chain is in state  $j$  with  $1 \leq I$  and  $j \leq m$ ; (iii) with probability  $r_{i,j}$ , the transition corresponds to no arrival and the state of the Markov chain is  $j$ , where  $j \neq i$ . Note that the Markov chain can go from state  $i$  to  $i$  only through an arrival. We define three matrices:  $C = (c_{i,j})$ ,  $D_A = (d_{i,j}^A)$  and  $D_B = (d_{i,j}^B)$ , where  $c_{i,i} = -\lambda_i$ ,  $1 \leq i \leq m$  and  $c_{i,j} = \lambda_i r_{i,j}$ ,  $j \neq i$ ,  $1 \leq i, j \leq m$  where  $d_{i,j}^A = \lambda_i p_{i,j}^A$ ,  $1 \leq i, j \leq m$ , and where  $d_{i,j}^B = \lambda_i p_{i,j}^B$ ,  $1 \leq i, j \leq m$ . We assume  $C$  to be a stable matrix so that the inter-arrival times will be finite with probability one and the arrival process will not terminate. The generator  $Q$  is then given by  $C + D_A + D_B$ . Note that

$$\sum_{j=1}^m p_{i,j}^A + \sum_{j=1}^m p_{i,j}^B + \sum_{j=1, j \neq i}^m r_{i,j} = 1. \tag{1}$$

Thus, the MAP for our model is described by three matrices,  $C$ ,  $D_A$  and  $D_B$  with  $C$  governing the transitions corresponding to no arrival,  $D_A$  governing those corresponding to arrivals of type A and  $D_B$  governing those corresponding to arrivals of type B. If  $\pi$  is the unique (positive) stationary probability vector of the Markov process with generator  $Q$ , then

$$\pi Q = \mathbf{0} \text{ and } \pi e = 1. \tag{2}$$

The constants,  $\lambda_A = \pi D_A e$  and  $\lambda_B = \pi D_B e$ , referred to as the fundamental rates, give the expected number of arrivals of type A and type B per unit of time respectively, in the stationary version of the MAP.

### 3. Markov Chain Description of the Model

The queueing model outlined above can be studied as a Markov chain with  $[m + 3m(N + 1)(K + 1)]^2$  states. The description of the states are given in Table 1.

Table 1

State	Description
*	The system is idle.
$(i, j, k, A)$	There are $i$ type $A$ messages and $j$ type $B$ messages in their respective buffers. The phase of the arrival process is $k$ and the server is processing a message of type $A$ .
$(i, j, k, B)$	There are $i$ type $A$ messages and $j$ type $B$ messages in their respective buffers. The phase of the arrival process is $k$ and the server is processing a message of type $B$ .
$(i, j, k, AB)$	There are $i$ type $A$ messages and $j$ type $B$ messages in their respective buffers. The phase of the arrival process is $k$ and the server is simultaneously processing two message, one of type $A$ and one of type $B$ .

The notation  $\mathbf{e}(e')$ , will stand for a column (row) vector of 1's;  $\mathbf{e}_i(e'_i)$  will stand for a unit column (row) vector with 1 in the  $i$ th position and 0 elsewhere,  $\mathbf{I}$  will represent an identity matrix of appropriate dimensions. The symbol  $\otimes$  denotes the Kronecker product of two matrices. Specifically  $L \otimes M$  stands for the matrix made up of the block  $L_{ij}M$ .

The generator  $Q^*$  of the Markov process governing the system is given as follows.

$$Q^* = \begin{bmatrix} C & B_0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ B_2 & B_1 & A_0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & A_2 & A_1 & A_0 & 0 & \dots & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & 0 & A_2 & A_1 & A_0 & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & A_2 & A_1 & A_0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & A_2 & A_3 \end{bmatrix}, \tag{3}$$

where

$$B_0 = e'_1 \otimes D_0,$$

$$B_1 = \begin{bmatrix} C_1 & I \otimes D_B & 0 & \dots & \dots & \dots & 0 \\ \mu_2 & C_1 & I \otimes D_B & 0 & \dots & \dots & 0 \\ 0 & \mu_2 & C_1 & I \otimes D_B & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & \mu_2 & C_1 & I \otimes D_B & 0 \\ 0 & \dots & \dots & 0 & \mu_2 & C_1 & I \otimes D_B \\ 0 & \dots & \dots & \dots & 0 & \mu_2 & C_1 + I \otimes D_B \end{bmatrix},$$

$$B_2 = e_1 \otimes \mu \otimes I,$$

$$A_0 = \begin{bmatrix} I \otimes D_1 & 0 & \dots & \dots & 0 \\ 0 & I \otimes D_A & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & I \otimes D_A & 0 \\ 0 & \dots & \dots & 0 & I \otimes D_A \end{bmatrix},$$

$$A_1 = \begin{bmatrix} C_1 & I \otimes D_B & 0 & \dots & \dots & \dots & 0 \\ 0 & C_1 & I \otimes D_B & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & C_1 & I \otimes D_B & 0 \\ 0 & \dots & \dots & \dots & 0 & C_1 & C_1 + I \otimes D_B \end{bmatrix},$$

$$A_2 = \begin{bmatrix} \mu_1 & 0 & \dots & \dots & \dots & 0 \\ \mu_3 & 0 & 0 & \dots & \dots & 0 \\ 0 & \mu_3 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & \mu_3 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} C_1 + I \otimes D_A & I \otimes D_B & 0 & \dots & \dots & 0 \\ 0 & C_1 + I \otimes D_A & I \otimes D_B & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & C_1 + I \otimes D_A & I \otimes D_B \\ 0 & \dots & \dots & \dots & 0 & C_1 + I \otimes (D_A + D_B) \end{bmatrix}$$

$$D_0 = [D_A D_B 0], \quad \mu = \begin{bmatrix} \mu_A \\ \mu_B \\ \mu_{AB} \end{bmatrix}, \quad C_1 = I \otimes C - \Delta(\mu), \Delta(\mu) = \begin{bmatrix} \mu_A I & 0 & 0 \\ 0 & \mu_B I & 0 \\ 0 & 0 & \mu_C I \end{bmatrix}$$

$$\mu_i = e'_i \otimes \mu \otimes I, \quad \text{where } I \leq i \leq 3.$$

### 4. Steady State Analysis

In this section we will derive expressions for the steady state probability vectors of the number of type *A* and type *B* messages at an arbitrary time as well as at arrival epochs.

#### 4.1 The Steady State Probability Vector at an Arbitrary Time

The stationary probability vector  $\mathbf{x}$  of the generator  $Q^*$  is the unique solution of the equations

$$\mathbf{x}Q^* = \mathbf{0}, \quad \mathbf{x}\mathbf{e} = \mathbf{1}. \tag{4}$$

In view of the high order of the matrix  $Q$ , it is essential that we use its special structure to evaluate the components of  $\mathbf{x}$ . We first partition  $\mathbf{x}$  into vectors of smaller dimensions as  $\mathbf{x} = (\mathbf{x}^*, \mathbf{u}(0), \mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(K))$ , where  $\mathbf{u}(i)$  is partitioned as  $\mathbf{u}(i) = (\mathbf{u}_0(i), \mathbf{u}_1(i), \mathbf{u}_2(i), \mathbf{u}_3(i), \dots, \mathbf{u}_N(i))$ .  $\mathbf{u}_j(i)$  is further partitioned as  $\mathbf{u}_j(i) = (\mathbf{v}(i, j, A), \mathbf{v}(i, j, B), \mathbf{v}(i, j, AB))$ . The  $\mathbf{v}$  vectors are of order  $m$ . The steady-state equations can be rewritten as

$$\begin{aligned} \mathbf{x}^*C + \mathbf{u}(0)B_2 &= 0, \\ \mathbf{x}^*B_0 + \mathbf{u}(0)B_1 + \mathbf{u}(1)A_2 &= 0, \\ \mathbf{u}(i-1)A_0 + \mathbf{u}(i)A_1 + \mathbf{u}(i+1)A_2 &= 0, 1 \leq i \leq K-1 \text{ and} \\ \mathbf{u}(K-1)A_0 + \mathbf{u}(K)A_3 &= 0. \end{aligned} \tag{5}$$

By exploiting the special structure of  $Q^*$ , the steady state equations in (5) can be efficiently solved in terms of smaller matrices of order  $m$  using block Gauss-Siedel iteration. For example, the first equation of (5) can be rewritten as

$$\mathbf{x}^* = [\mu_A \nu(0, 0, A) + \mu_B \nu(0, 0, B) + \mu_{AB} \nu(0, 0, AB)](-C)^{-1}. \tag{6}$$

The rest of the required equations can be derived in a similar way and the details are omitted.

**4.2 Accuracy Check**

The following intuitively obvious equation can be used as an accuracy check in the numerical computation of  $\mathbf{x}$ .

$$\mathbf{x}^* + \sum_{i=0}^K \sum_{j=0}^N [\nu(i, j, A) + \nu(i, j, B) + \nu(i, j, AB)] = \pi, \tag{7}$$

where  $\pi$  is as given in equation (2).

**4.3 The Stationary Queue Length at Arrivals**

The joint stationary density of the number of messages in the queue, the type of service and the phase of the arrival process at arrival epochs is determined in this section. We denote the stationary probability vectors at an arriving point for type  $A$  message as  $\mathbf{z}_A$ , where  $\mathbf{z}_A = (z_A^*, z_A(0), z_A(1), \dots, z_A(K))$  and for message  $B$  as  $\mathbf{z}_B$ , where  $\mathbf{z}_B = (z_B^*, z_B(0), z_B(1), \dots, z_B(N))$ , respectively. These vectors can be expressed as  $V^{ZA}$  and  $V^{ZB}$  vectors. These vectors are given by

$$\begin{aligned} z_A^* &= \frac{1}{\lambda_A} x^* D_A, \quad \nu^{ZA}(i, j, A) = \frac{1}{\lambda_A} \nu(i, j, A) D_A, \\ \nu^{ZA}(i, j, B) &= \frac{1}{\lambda_A} \nu(i, j, B) D_A \text{ and } \nu^{ZA}(i, j, AB) = \frac{1}{\lambda_A} \nu(i, j, AB) D_A \end{aligned} \tag{8}$$

$$\text{where } 1 \leq i \leq K, \quad 1 \leq j \leq N,$$

and

$$\begin{aligned} z_B^* &= \frac{1}{\lambda_B} x^* D_B, \quad \nu^{ZB}(i, j, A) = \frac{1}{\lambda_B} \nu(i, j, A) D_B, \\ \nu^{ZB}(i, j, B) &= \frac{1}{\lambda_B} \nu(i, j, B) D_B, \quad \nu^{ZB}(i, j, AB) = \frac{1}{\lambda_B} \nu(i, j, AB) D_B, \end{aligned} \tag{9}$$

$$\text{where } 1 \leq i \leq K, \quad 1 \leq j \leq N.$$

**4.4 Stationary Waiting Time Distribution of an Admitted Message of Type A or Type B**

Let  $\mathbf{y}_A^r(i, j)$  denote the steady state probability vector that an arriving type  $A$  message finds  $i$  type  $A$  messages and  $j$  type  $B$  messages and the server is busy with a type  $r$  message, for  $r = A, B, AB$  with  $0 \leq i \leq K$  and  $0 \leq j \leq N$ . We define  $\mathbf{y}_A^*$  to be the steady state probability vector that an arriving type  $A$  message finds the system to be idle.

We can show that the waiting time is of phase type with representation  $(\Theta_A, L_A)$  of order  $K$ . Here  $\Theta_A$  and  $L_A$  are given by

$$\Theta_A = c(z_A(0), z_A(1), \dots, z_A(K-1)), \tag{10}$$

where the normalizing constant  $c$  is given by  $c = (1 - z_A(K)e)^{-1}$ , and



$$L_A = \begin{bmatrix} -I \otimes \Delta(\mu) & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ B_0 & A_1 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & B_1 & A_2 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_{i-1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & B_1 & A_i & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & B_1 & A_K \end{bmatrix}, \tag{11}$$

where

$$\mu = \begin{bmatrix} \mu_A \\ \mu_B \\ \mu_{AB} \end{bmatrix}, \quad \Delta(\mu) = \begin{bmatrix} \mu_A & 0 & 0 \\ 0 & \mu_B & 0 \\ 0 & 0 & \mu_{AB} \end{bmatrix},$$

$$B_0 = \begin{bmatrix} \Delta(\mu) \otimes e & 0 & 0 & \cdots & 0 & 0 \\ e'_3 \otimes \mu \otimes e & 0 & 0 & \cdots & 0 & 0 \\ 0 & e'_3 \otimes \mu \otimes e & 0 & \cdots & 0 & 0 \\ 0 & 0 & e'_3 \otimes \mu \otimes e & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & e'_3 \otimes \mu \otimes e & 0 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} \Delta(\mu) \otimes I & 0 & 0 & \cdots & 0 & 0 \\ e'_3 \otimes \mu \otimes I & 0 & 0 & \cdots & 0 & 0 \\ 0 & e'_3 \otimes \mu \otimes I & 0 & \cdots & 0 & 0 \\ 0 & 0 & e'_3 \otimes \mu \otimes I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & e'_3 \otimes \mu \otimes I & 0 \end{bmatrix},$$

and

$$A_i = \begin{bmatrix} E_1 & E_2 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & E_1 & E_2 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & E_1 & E_2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & E_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & E_3 \end{bmatrix}, \quad 1 \leq i \leq K,$$

with  $E_1 = I \otimes (C + D_A) - \Delta(\mu) \otimes I$ ,  $E_2 = I \otimes D_B$  and  $E_3 = I \otimes Q - \Delta(\mu) \otimes I$ .

The waiting time distribution of type  $B$  messages can be derived in a similar manner, the details of which are omitted.

### 4.5 Measures of System Performance

Once the steady state probability vector  $\mathbf{x}$  has been computed, several interesting system performance measures, useful in the interpretation of the physical behavior of the model, may be derived. We list a number of such measures along with the corresponding formulas needed in their derivation.

(a) *The overflow probabilities of type A and type B messages.*

The probability that an arriving message of type  $A$  or type  $B$  will find the buffer full is given by

$$P(\text{Overflow of type } A \text{ messages}) = \frac{1}{\lambda_A} \sum_{j=0}^N (\nu(K, j, A) + \nu(K, j, B) + \nu(K, j, AB)) D_A e \tag{12}$$

$$P(\text{Overflow of type } B \text{ messages}) = \frac{1}{\lambda_B} \sum_{i=0}^K (\nu(i, N, A) + \nu(i, N, B) + \nu(i, N, AB)) D_B e. \tag{13}$$

(b) *The throughput of the system*

The throughput,  $\gamma$ , of the system is defined as the rate at which the messages leave the system. It is given by

$$\begin{aligned} \gamma &= \lambda_A [1 - P(\text{Overflow of type } A \text{ messages})] \\ &\quad + \lambda_B [1 - P(\text{Overflow of type } B \text{ messages})]. \end{aligned} \tag{14}$$

(c) *The fractions of the time the server is busy with type A, type B and type AB messages*

We denote by  $P_A$  or  $P_B$ , respectively the proportion of time the server is serving a message of type  $A$  or the proportion of time the server is serving a message of type  $B$ .  $P_{AB}$  is the proportion of time the server is serving messages of types  $A$  and  $B$  at the same time.

$$P_A = \sum_{i=0}^K \sum_{j=0}^N \nu(i, j, A) e, \quad P_B = \sum_{i=0}^K \sum_{j=0}^N \nu(i, j, B) e$$

and

$$P_{AB} = \sum_{i=0}^K \sum_{j=0}^N \nu(i, j, AB)e. \tag{15}$$

(d) *The fraction of time the server is idle*

The fraction of the time the server is idle is given by  $\mathbf{x}^*e$ .

(e) *The expected queue lengths of buffers A and B*

The expected queue lengths of buffers A and B are given by

$$E(\text{Queue length of } A) = \sum_{i=1}^K i \sum_{j=1}^N (\nu(i, j, A) + \nu(i, j, B) + \nu(i, j, AB))e$$

and

$$E(\text{Queue length of } B) = \sum_{i=1}^N i \sum_{j=1}^K (\nu(j, i, A) + \nu(j, i, B) + \nu(j, i, AB))e. \tag{16}$$

(f) *The mean waiting time of A and B*

The mean waiting times of message A and message B are given by (using Little’s result)

$$\text{Mean Waiting Time of } A = \frac{\text{Expected queue length of } A}{\lambda_A(1 - P\{\text{Overflow of } A\})}$$

and

$$\text{Mean Waiting Time of } B = \frac{\text{Expected queue length of } B}{\lambda_B(1 - P\{\text{Overflow of } B\})}.$$

### 5. Numerical Examples

We consider four different MAPs which have the same fundamental arrival rate at 10, but which are qualitatively different in the sense that they have different variance and correlation structures. The arrival distributions of the four different MAPs are given in Table 2 below.

**Table 2: MAP Representation**

MAP	C		D	
1	-95.63539	0.00000	95.49759	0.13780
	0.00000	-0.10585	0.01592	0.08993
2	- 5.45210	0.00000	0.81800	4.63410
	0.00000	-163.56480	156.47670	7.08810
3	Erlang of order 5 with rate 50 in each state			
4	-25.00000	0.00000	24.75000	0.25000
	0.00000	-0.16556	0.16390	0.00166

We set  $K = N = 15$ ,  $\mu_A = 4$ , and  $\mu_B = 6$ . For the type A and type B message arrivals, we consider the MAPs listed in Table 2. We use the notation MAP<sub>ij</sub> to denote the case where type A arrivals occur according to MAP<sub>i</sub> and type B arrivals occur according to MAP<sub>j</sub>. The performance measures - the throughput, the type A and type B message loss probabilities, the idle probability of the server, the fraction of time the server is busy with type  $r$ ,  $r = A, B$  and  $AB$ , messages, the mean queue lengths of type A and type B messages and the mean waiting time of an admitted

type  $A$  message - are plotted as functions of  $\mu_{AB}$  in Figures 1-22.

The observations from these figures are summarized as follows.

[1] The type  $A$  and type  $B$  message loss probabilities appear to decrease as the service rate  $\mu_{AB}$  increases for all MAP combinations. This characteristic is as is to be expected. However, we find that the amount of decrease is more significant in the cases where the arrival process for neither type- $A$  nor type- $B$  arrivals is a bursty one. In other words, if either type- $A$  or type- $B$  arrivals are bursty, then increasing  $\mu_{AB}$  does not have much effect on the type- $A$  or type- $B$  message loss probabilities. In addition, for any other combination of MAPs, these probabilities become close to zero when the service rate  $\mu_{AB}$  exceeds the larger of  $\mu_A$  and  $\mu_B$ .

[2] It is interesting to see how the message loss probability is affected significantly when going from highly bursty to non-bursty type arrivals. For example, consider the case where type  $B$  arrivals are governed by MAP\_3. This MAP corresponds to Erlang arrivals and hence arrivals are not bursty in nature. Suppose we compare the type  $B$  loss probabilities when type  $A$  arrivals have (a) MAP\_1 and (b) MAP\_2 representations. The graphs of the performance measures for those two cases can be seen in Figure 3. For values of  $\mu_{AB} \leq \mu_A$ , type  $B$  loss probability is significantly high for (b) compared to that for (a). For values of  $\mu_{AB} > \mu_B$ , there is no significant difference. This can be explained as follows. The bursty nature of type  $A$  arrivals in (a) leads to more services of type  $A$  alone as well as of type  $B$  alone, compared to simultaneous services of type  $A$  and type  $B$ . On the other hand, for the case in (b), both type  $A$  and type  $B$  arrive in a non-bursty manner and hence more simultaneous services of type  $A$  and type  $B$  are performed. Since in the initial stages,  $\mu_{AB} > \mu_A$  and  $\mu_{AB} > \mu_B$ , the type  $B$  message loss probability is greater for case (b) than for case (a). The same phenomenon occurs when the MAPs for type  $A$  and type  $B$  are reversed (see Figure 2).

[3] There is a small increase in the idle probability of the server as  $\mu_{AB}$  increases. Also, for any fixed value of  $\mu_{AB}$ , the idle probability is greater if any of the arrival process is highly bursty. The phenomenon of high loss probability and high idle probability for the same set of parameters seems to be a little contradictory initially. However, when we look at the arrival process that leads to this phenomenon, we shall see that it is not contradictory at all. In the case when the arrivals are highly bursty, we shall see intervals of low to moderate number of arrivals separated by intervals of high number of arrivals. During the periods of high arrivals, the buffer will be filled in quickly leading to high loss probability; and during the periods of low to moderate number of arrivals, the server may clear the messages, which leads to high idle probability. Note that this phenomenon occurs only when the arrivals are bursty and not otherwise.

[4] The probability of the server being busy with a type  $A$  message alone (or type  $B$  alone) increases as  $\mu_{AB}$  increases. Correspondingly, the probability of the server being busy with both type  $A$  and type  $B$  messages decreases. This is obvious because  $\mu_A$  and  $\mu_B$  are fixed at 4 and 6 while  $\mu_{AB}$  increases from 0.5 to 10. So, for larger values of  $\mu_{AB}$ , the server will take more time to service just type  $A$  alone or type  $B$  alone, than to serve them together. Another point to be noted is that, for both MAP\_31 and MAP\_41 cases, the probability of the server being busy with type  $A$  messages is significantly higher than the probability of the server being busy with type  $B$  messages. This can be explained as follows. Since type  $B$  messages arrive in a bursty manner, the server spends most of the time serving type  $A$  messages, and not type  $B$  ones. At other times, the server is busy serving both types simultaneous-

ly. It is also interesting to note that the probability that the server is busy with both types of messages in service is higher when both types of arrivals have MAPs with positive correlation.

[5] As is to be expected, the mean queue lengths of type  $A$  and type  $B$  messages decrease as  $\mu_{AB}$  increases. However, the decrease is less significant when any one of the arrival processes is negatively correlated or is highly bursty in nature.

[6] The throughput for MAP combinations increases to a point as  $\mu_{AB}$  is increased. This is as is to be expected. But for very low values of  $\mu_{AB}$ , we find that the MAP combinations with negatively correlated arrival processes have smaller throughput than positively correlated ones. However, as  $\mu_{AB}$  increases, a crossover of these plots takes place; and for a high value of  $\mu_{AB}$ , the throughput is much higher for processes that have zero or negative correlation.

[7] The mean waiting times of admitted type  $A$  and type  $B$  messages appear to decrease when  $\mu_{AB}$  increases, for all MAP combinations. In the case of low values of  $\mu_{AB}$ , we note the following interesting observation. When at least one of the types has highly bursty arrivals, the mean waiting time of that type appears to be smaller compared to other MAPs, for any fixed  $\mu_{AB}$ . This again is due to the fact that, in this case, the server will be less busy serving both types of messages simultaneously compared to serving type  $A$  alone or type  $B$  alone. Since  $\mu_{AB}$  is small (compared to  $\mu_A$  and  $\mu_B$ ), the server will take more time on the average to process both types simultaneously. However, in this case the server is busy most of the time with either type  $A$  or type  $B$  messages alone, which leads to a lower mean waiting time. When  $\mu_{AB}$  is large, the mean waiting time appears to be large for highly bursty arrival type messages, which is again as is to be expected.

## 6. Variant of the Model-Group Services

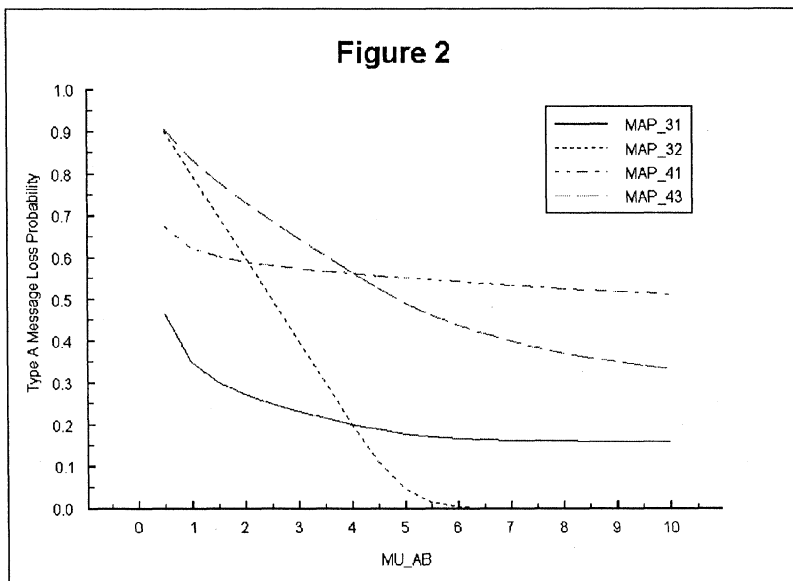
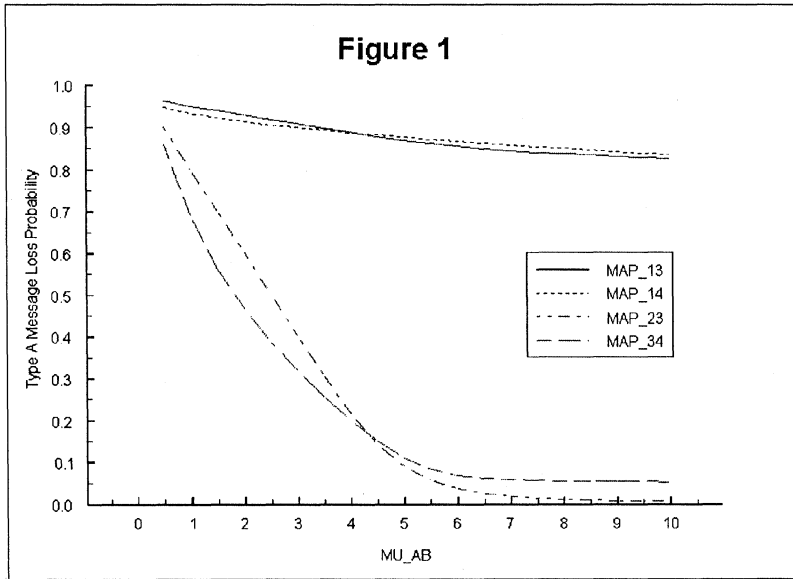
The model can also be extend to include group services. Instead of serving messages one at a time, the server can be modeled such that it is capable of serving  $n_1$  messages of type  $A$ ,  $n_2$  messages of type  $B$  and  $n_3$  combinations of type  $A$  and type  $B$  messages at the same time.

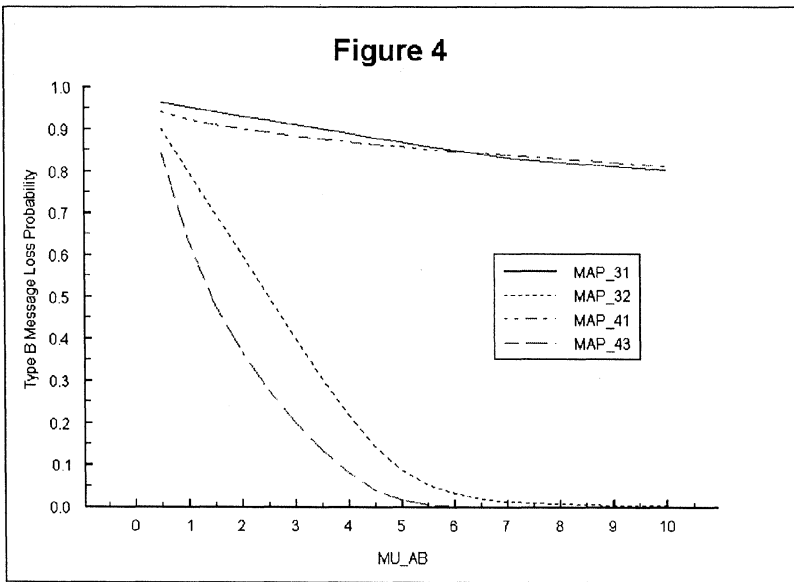
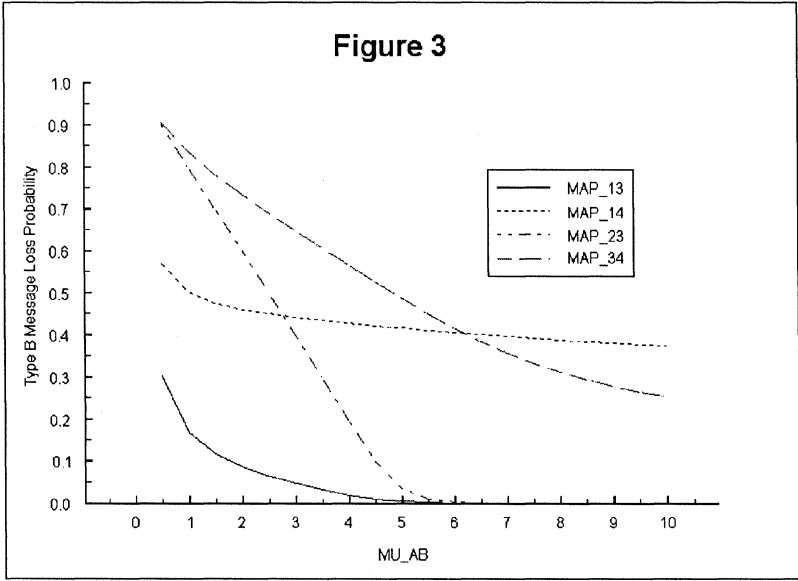
## Acknowledgements

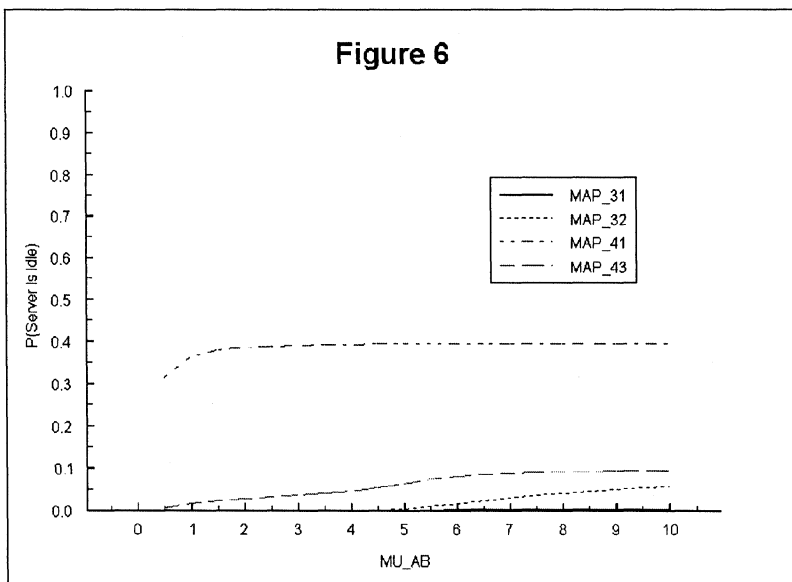
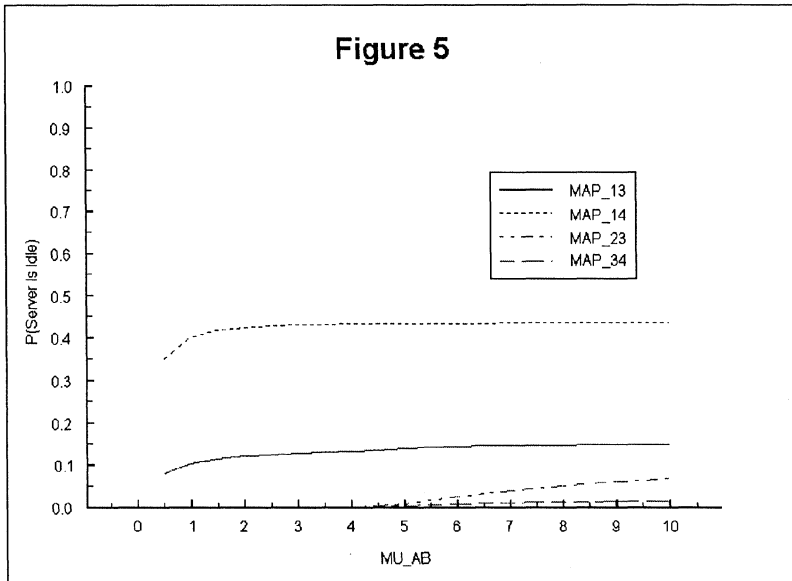
The authors wish to thank the editors and the referees for a careful reading of the manuscript and offering valuable suggestions.

## References

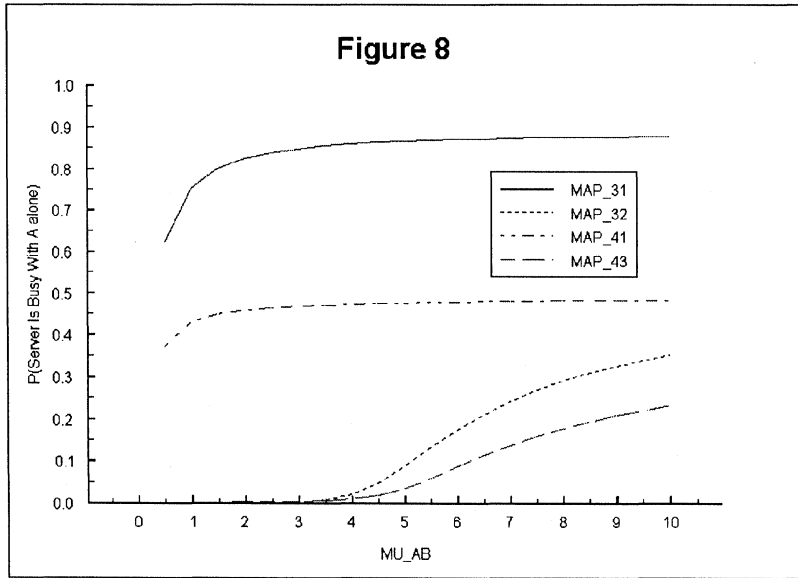
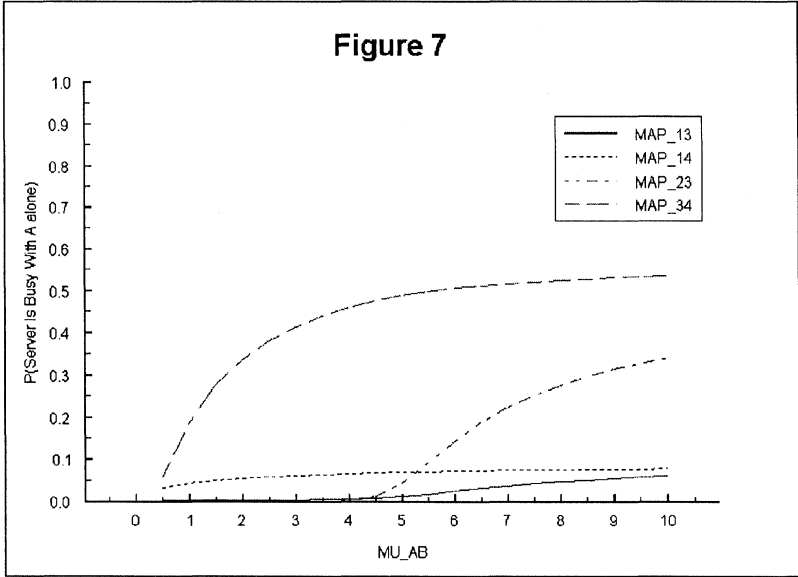
- [1] Bellman, R.E., *Introduction to Matrix Analysis*, McGraw-Hill, New York 1960.
- [2] Lucantoni, D.M., New results on the single server queue with a batch Markovian arrival process, *Stochastic Models* 7 (1991), 1-46.
- [3] Neuts, M.F., A versatile Markovian point process, *J. Appl. Prob.* 16 (1979), 764-799.
- [4] Neuts, M.F., *Matrix-Geometric Solutions in Stochastic Models - An Algorithmic Approach*, The John Hopkins University Press, Baltimore 1981.

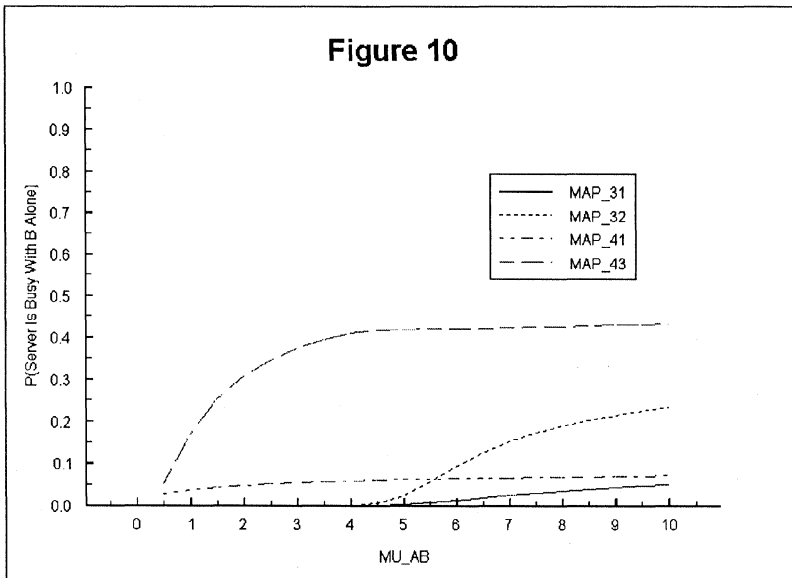
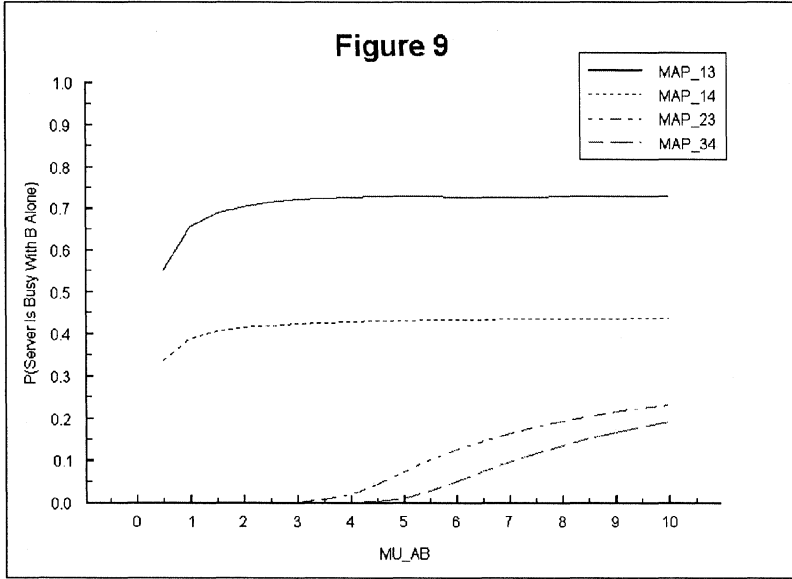


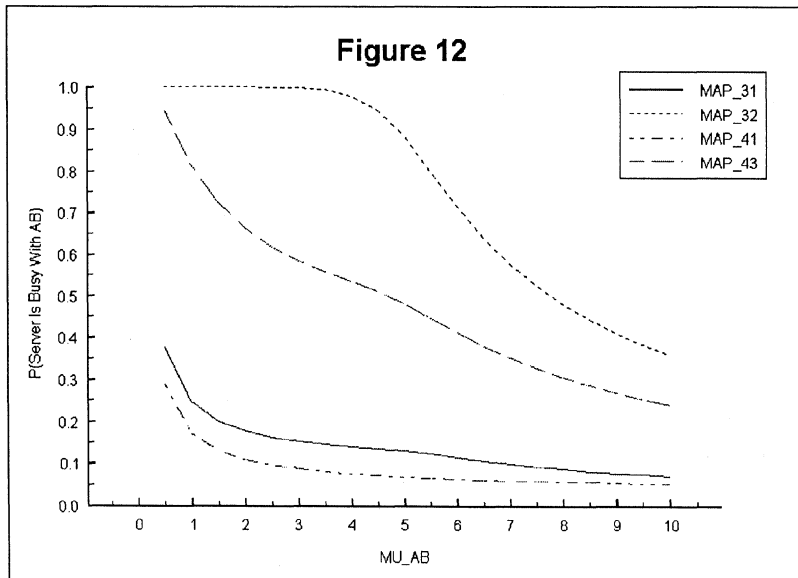
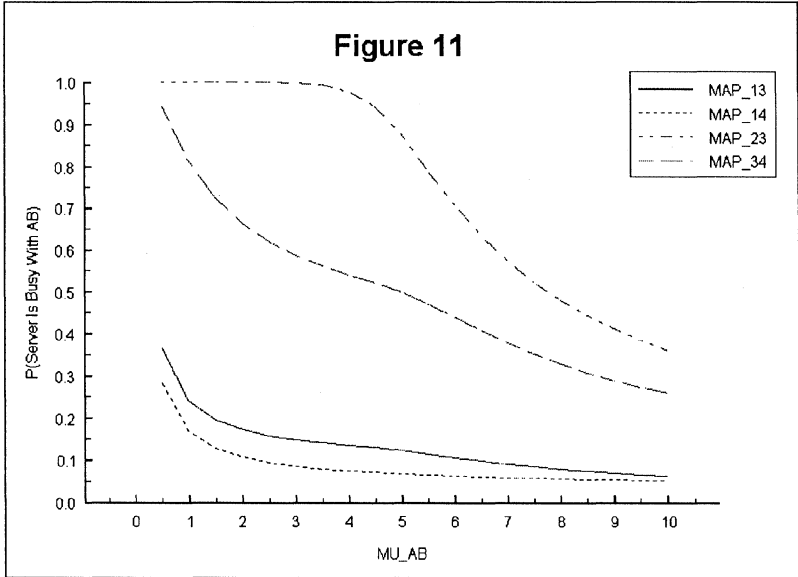


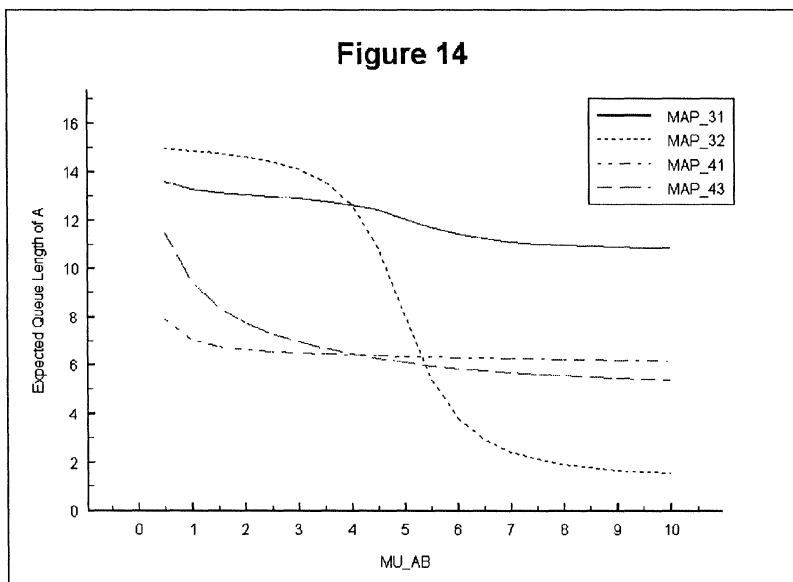
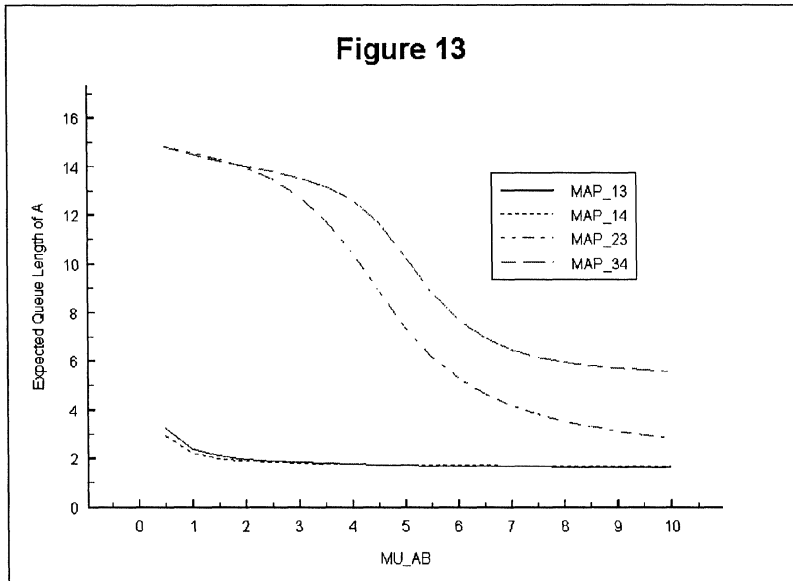


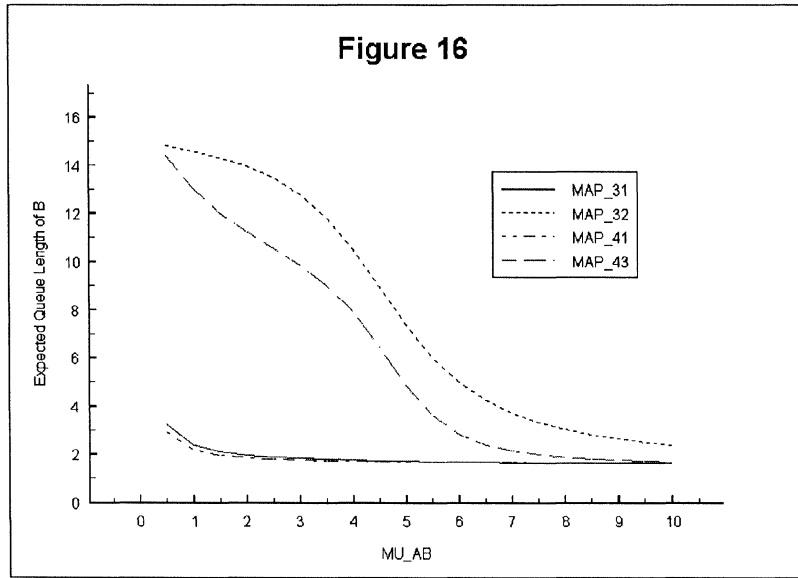
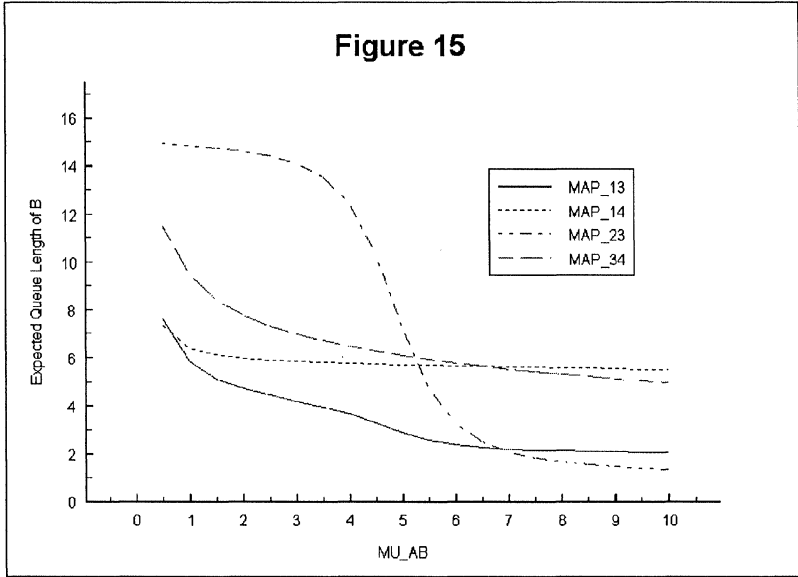


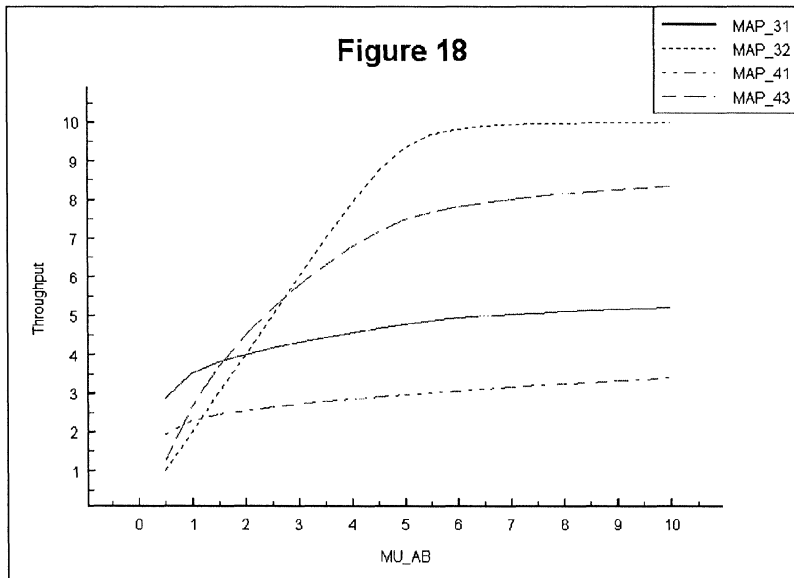
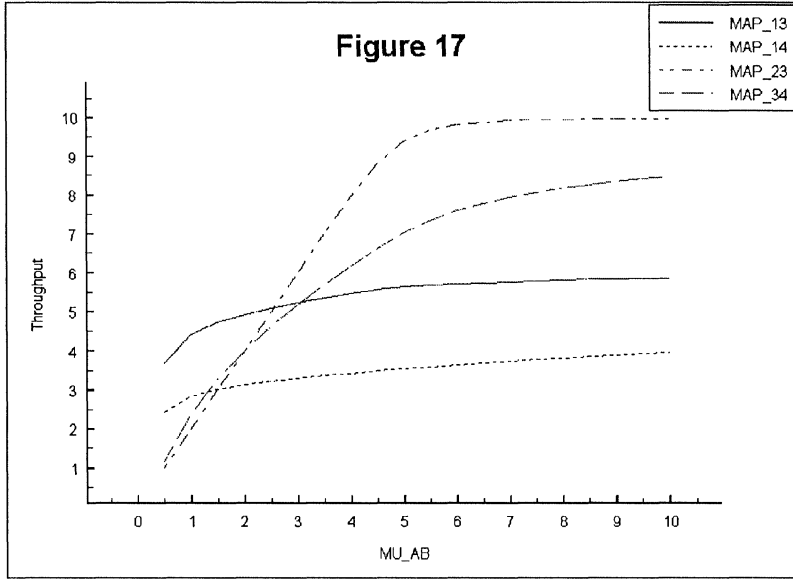


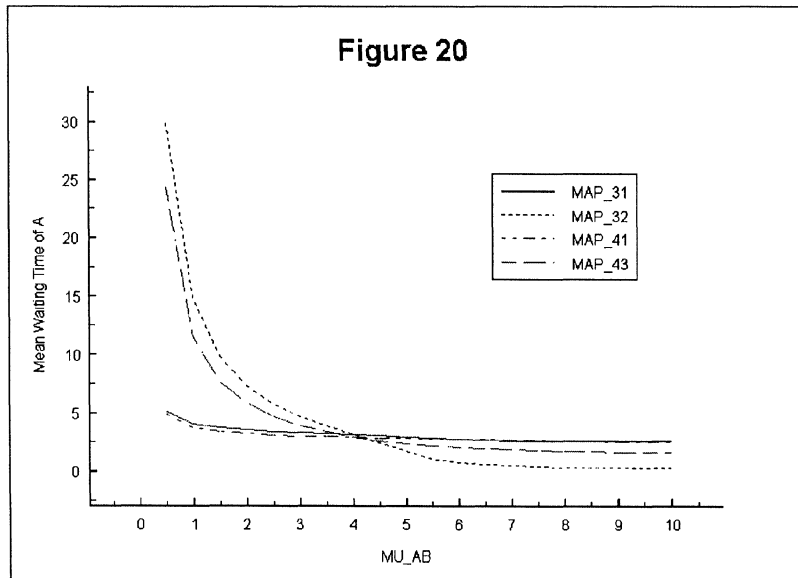
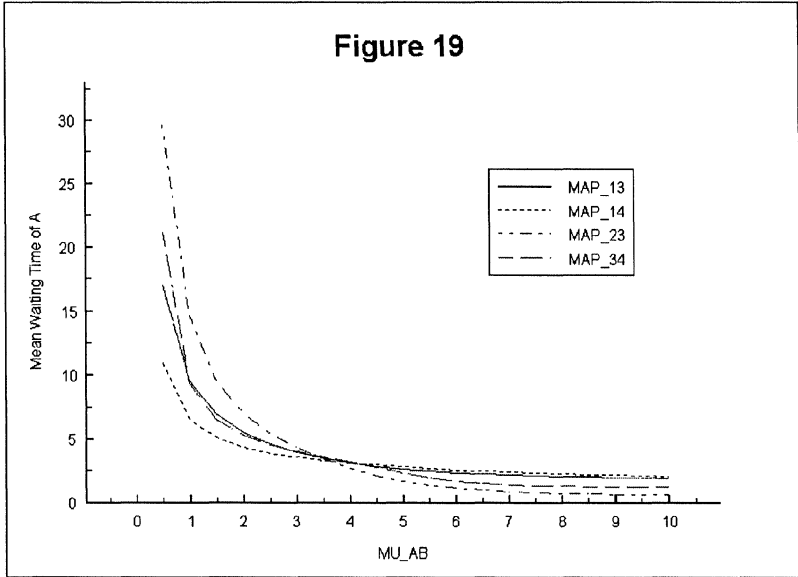


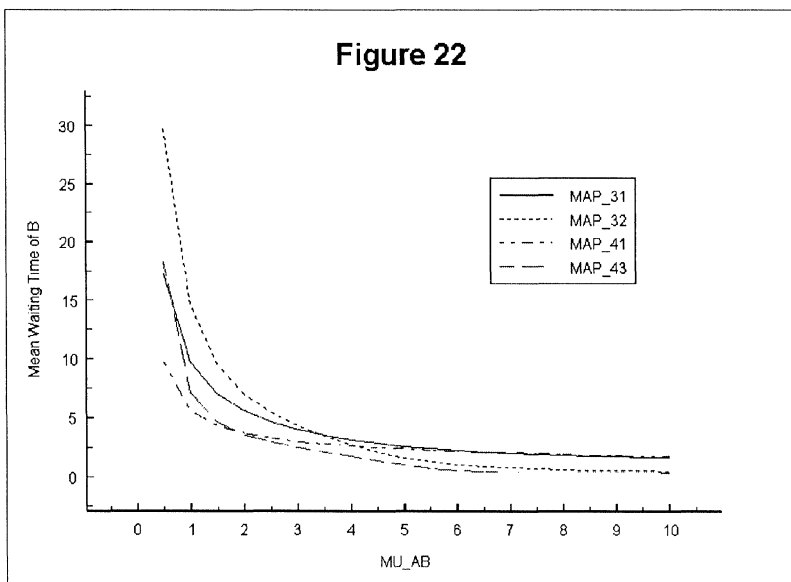
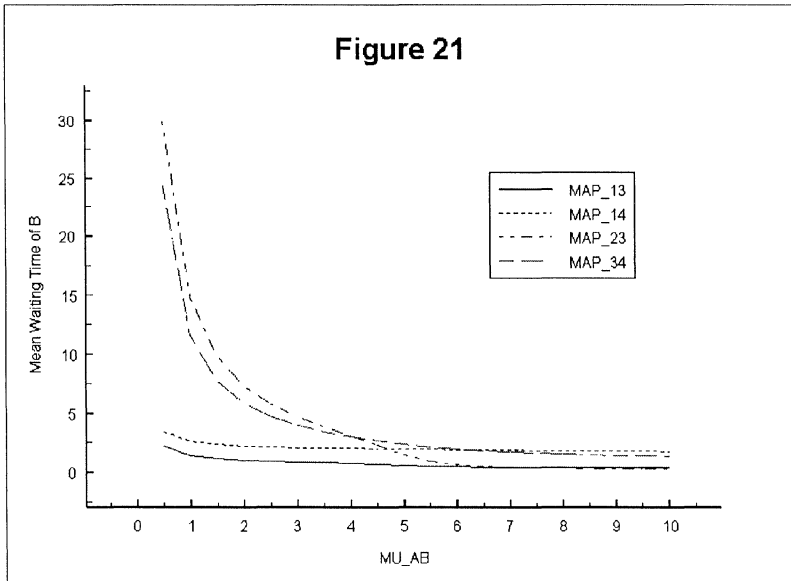
















# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

