ON THE DESIGN AND ANALYSIS OF MULTI-USER
MULTI-CHANNEL SYSTEMS WITH CENTRALIZED
CONTROL

by

I. Chlamtac and O. Ganz

Technical Report #257

January 1983.
ABSTRACT

This study considers multi-user distributed systems with multiple channel interconnection. Communicating users send messages to a finite length central buffer from which they are picked synchronously by a channel controller. Several messages can be allocated channels simultaneously provided no reference conflict occurs.

Exact and approximate Markovian models which account for the various reference conflicts, channel contention and buffer overflow probability are presented. For approximate models assumptions are validated by detailed simulation.

Alternative system configurations are analyzed and compared. The presented results can be used in the system design process with the objective of satisfying system constraints in terms of message delays and overflow probabilities while reducing the cost of channels and buffering.

The application of the model is demonstrated for satellite communication systems with on board processing and for multiprocessor distributed systems with multiple channel controllers.

Keywords: multiple-user systems, multiple-channel interconnections, performance modeling, channel allocation control, channel contention, reference conflicts.
1. INTRODUCTION

In a distributed system with multiple channel based interconnection the information between communicating parties can be exchanged in parallel. The interconnection system can be used to effect communication between the system users, e.g. network nodes [3,4,5], or to provide access to common points of reference, e.g., memories in multiprocessor systems [6,7].

When the number of interconnecting channels, or buses, is smaller than the number of users, i.e., the system is highway inefficient, a bus conflict or contention will occur [5,8]. If, furthermore, more than one user attempts to access a common point of reference, a reference conflict results [6,7]. The effect of the various conflicts must be accounted for in distributed systems design and analysis so that high total throughput and efficient resource utilization can be achieved.

Several types of distributed systems exhibit the model behaviour just described. In multiple bus multiprocessor systems common memory areas are accessed by processors. Simultaneous access to as many memories as there are buses available is permitted but every memory area can be accessed by only a single processor at a time [7]. Satellite communication systems also fall into this category. Here traffic arriving from various source regions is routed at the satellite to the destination regions by a number of synchronized transponders [2,9]. Again, the number of transponders can be smaller than the number of regions and every destination region will be accessed by no more than one transponder at a time. In satellite systems we can further assume that transponders will not be used to route traffic from a region, or station, to itself.

All of the described systems are similar in that they are affected
by bus contention and reference conflicts although the exact nature of the reference conflict may differ from one system to another. Another common feature of the systems under consideration is the synchronous allocation of channels to user requests in constant time intervals. This type of allocation is found in several system bus architectures of tightly coupled multiprocessor systems \([1,4]\) and in multiaccess protocols, e.g., TDMA for satellite communication \([9,10]\).

To achieve controlled allocation of channels in face of bus contention and reference conflicts an arbitration policy is enforced. One common approach to controlling the channels is had by the use of a multiple channel controller which serves as a centralized arbitrator \([6]\). The channel controller can typically store a finite number of user requests and will allocate channels to users while attempting to reduce conflict probability.

In this paper we deal with distributed synchronized systems using multiple channels with centralized channel control. We present exact and approximate Markovian models which allow us to model different types of reference conflicts within a single modeling framework.

We consider the system performance in terms of expected delays and buffer overflow probabilities for various interconnection system configurations. Results from exact and approximate models are presented and compared with those derived from detailed simulations.
2. THE MODEL

This study considers a system which consists of a buffer of a finite size $M$ (the controller buffer) to which there is an arrival of messages from $N_1$ independent users (processors, areas) destined to $N_2$ points of reference (e.g. memories, areas). There are $b$ synchronized and dependent servers (buses) with deterministic service time.

At the beginning of every time slot up to $b$ suitable messages are picked for service (transmission). We assume this arbitration to be executed in zero time.

Which and how many of the messages in the buffer will be chosen in each service interval depends on the number of system buses and on the referencing pattern of the system.

In the preceding section we have introduced two different reference models. In case of a user to user communication e.g., satellite computer network, a user will not refer to itself. In this case $N_1 = N_2 = N$.

In the case of resource access, e.g. processor to memories communication, no such limitation exists. Clearly in this case $N_1$ can be different from $N_2$. The model introduced here is not limited to a single reference model and in following sections we show how steady state probabilities can be obtained for both reference models. For sake of clarity of presentation we shall henceforth, unless stated otherwise, refer to the user to user communication reference model.

We will refer to a system with $N$ users, $M$ buffers and $b$ buses as a $N \times M \times b$ system.

The following assumptions are made regarding the system operation:

1. Message arrivals to the users are assumed to be generated from a
Poisson process with an average aggregate arrival rate $\lambda$. Users have uniform access rate $\lambda_i$,

$$\lambda_i = \frac{\lambda}{N}, \quad i = 1, \ldots, N$$

(2) The buffer is of finite size $M$. When a message arrives and finds the buffer full it is rejected, i.e., an overflow results.

(3) A uniform reference model is assumed. This implies that every access request from every user is directed to any other user with equal probability. Thus the access rate from user $i$ to user $j$ is given by:

$$\lambda_{ij} = \frac{\lambda_i}{N-1}, \quad \forall i, j \quad i \neq j.$$ 

(4) Service time is constant $\frac{1}{\mu}$.

(5) Up to $b$ messages with different destinations can be randomly selected for service. The selection process is done in zero time.

(6) Simultaneous access of two or more customers to a single reference point (memory, area, etc.) is not permitted.

It is possible to construct a queueing network for the analysis of the system as shown, for the general case, in Figure 1.

**Figure 1** - Open queueing network of an $N \times M \times b$ system
In queueing theory terminology the above system implies that there is a gate between the servers and the waiting room which is opened at fixed intervals.

The queueing model has finite buffer size (waiting line) and synchronous multiple transmission permits (b servers) which can be granted as long as reference conflicts do not occur.

3. PERFORMANCE MEASURES

The key performance measure of the system are the queueing delay and the buffer overflow probability. We define the queueing delay as the time that passes between the message arrival into the buffer and the beginning of its transmission. For a given bus speed the number of buses and the number of users will play the dominant role in determining this delay. To derive the user average queueing delay \( W \), we proceed as follows: We define \( p_i \) to be the probability that at the end of a service interval there are \( i \) messages in the system.

Define \( \alpha \) to be the carried load. When a message arrives and finds the buffer full, an overflow will result. Therefore the average message departure rate from the buffer (carried load) \( \alpha \) is less than the offered load from the users, \( \lambda \).

Define \( P_{of} \) to be the buffer overflow probability which is the expected fraction of the total number of messages rejected by the buffer. Consequently,

\[
P_{of} = P_M \tag{1}
\]

and

\[
\alpha = \lambda (1 - P_{of}) \tag{2}
\]

Define \( L \) to be the average number of messages in the system:

\[
L = \sum_{i=1}^{M} i p_i + \frac{\alpha}{2} \tag{3}
\]
The first term in equation (3) is the expected number of messages in the system at the beginning of a service interval. Since the packets cannot leave the system during the service interval we add the time averaged number of packet arrivals during the service interval.

Applying Little's result to the entire system, including queues and servers, we obtain $D$ in (4) the average access time which is the amount of time which passes from the submission of a bus request until the request is satisfied.

$$D = \frac{L}{\alpha} \quad (4)$$

In the given system the message waiting time results from contending for the bus with other messages and from conflicts created by more than one message trying to access a common point of reference (destination conflict).

The average queueing delay $W$ is simply given by:

$$W = \frac{1}{\alpha} - \frac{1}{\mu} \quad (5)$$

and the average number of queued messages $L_q$ will be:

$$L_q = W \cdot \alpha \quad (6)$$
4. EXACT MODELS

To model the system behaviour under the assumptions given in the previous section we build a discrete time Markov chain, obtained by observing the state of the system at the end of a service time interval.

The state of the Markov chain is defined by the N-tuple:

$$(S_1, \ldots, S_N)$$

where $S_i$: the number of messages queued for user i (i.e. reference point) at the end of a service time interval.

In this model the number of states increases very rapidly with the system size. The explosive growth is due to the detailed information that the states record about the messages queued for each destination.

Using the theory of "lumpable" Markov chains, we may lump equivalent states and obtain a Markov chain of substantially smaller size [11].

The state definition of the exact lumped chain is given by:

$$(q_1, \ldots, q_N)$$

where: $q_i$ is the number of messages waiting for a destination at the end of a service interval, arranged in decreasing order.

The number of states of the exact lumped chain is:

$$S = \sum_{j=0}^{M} \sum_{i=0}^{j} p_i(j)$$

where: $p_i(j)$ is the number of unordered partitions of $j$ messages into $i$ destinations (reference points) assuming that at least one message exists for every destination.

The expression for $S$ is derived in Appendix I.

Let $\pi_i$ be the probability of exactly $i$ messages originating from a Poisson process in a service time interval. Without loss of
generality we can let the service interval to equal unity \( \mu = 1 \)

\[
\pi_i = \frac{e^{-\lambda} \lambda^{i}}{i!}
\]  

(9a)

where \( \lambda \) is the average aggregate message arrival rate as defined earlier.

Let further \( \pi_{Gi} \) be the probability of at least \( i \) messages originating from a Poisson process

\[
\pi_{Gi} = 1 - \sum_{k=0}^{i-1} \pi_k
\]  

(9b)

An example of the state transition rate diagram for a \( 3 \times 2 \times 2 \) system with \( \pi_i, \pi_{Gi} \) as given in (9) is shown in Figure 3.

![State transition rate diagram for 3x2x2 system](image)

Figure 3 - State transition rate diagram for \( 3 \times 2 \times 2 \) system.

To obtain the required performance measures we have to find the set of state probabilities \( p_i \)'s - the probability of \( i \) messages in system at the end of a service interval. In the case of Figure 3 the \( p_i \)'s are composed of:

\[
p_0 = p_{00}
\]

\[
p_1 = p_{10}
\]

\[
p_2 = p_{11} + p_{20}
\]
We thus have to find the state probabilities \( p(q_1, q_2) \) from the following set of linear equations with \( q_i \) as given in (8).

\[
\begin{align*}
    p_{00} &= p_{00} \pi_0 + p_{10} \pi_0 + p_{11} \pi_0 \\
    p_{10} &= p_{00} \pi_1 + p_{10} \pi_1 + p_{11} \pi_1 + p_{20} \pi_0 \\
    p_{11} &= p_{00} \frac{2}{3} \pi G_2 + p_{10} \frac{2}{3} \pi G_2 + p_{11} \frac{2}{3} \pi G_2 + p_{20} \frac{2}{3} \pi G_1 \\
    p_{00} + p_{10} + p_{11} + p_{20} &= 1
\end{align*}
\]

(10)

After finding the \( p_i \)'s we calculate \( p_{of} \), \( \alpha \) and \( L \) from equations (1), (2), (3) respectively.

Using (5) we finally obtain \( W \), the average queueing delay in the system.

As clearly demonstrated in (10) even with lumped state representation an increase in the number of users \( N \) and in the buffer size \( M \) complicates the Markov chain.

The model complexity can be significantly reduced by building approximate Markovian models, in which the amount of information recorded in each state is reduced.

5. APPROXIMATE MODELS

Recall that in the exact model the states recorded complete information about queues inside the system, i.e. for each state of the Markov chain, the number of messages queued for each "destination" was recorded. With this complete state description all possible ways of distributing the messages among the system queues could be explicitly accounted for.

The approximate model that we introduce in this section analyses the system behaviour by using Markov models in which transition between the states are based on partial queueing information only.
In this model the state of the system is represented by the total number of messages in system at the end of a service time interval.

We thus have the following state definition:

\[(n)\]  

where \(n\): the number of messages in the system at the end of a service time interval.

To evaluate the transition rates without the knowledge of internal system queues, we make the assumption that at the beginning of the service time interval, for each queued message, the source and the destination are reselected at random with the same probability.

The results obtained in this way are therefore only approximate but on the other hand the system complexity is significantly reduced as shown in (11).

To compute the state probability \(p_i\) we define \(R_{ij}\) - the probability that among \(i\) messages in the queue there are exactly \(j\) messages with different destinations.

Using \(R_{ij}\), we can define \(R_{ij}\) - the probability that among \(i\) messages in the queue there are exactly \(j\) messages whose transmission can be carried out given there are \(b\) buses in the system:

\[
R_{ij} = \begin{cases} 
1 - \sum_{k=1}^{b-1} R'_{ik} & j = b > 1 \\
R'_{ij} & j < b, b > 1 \\
1 & b = 1
\end{cases}
\]  

The computation of \(R'_{ij}\) depends on the referencing pattern of the system in question (see Section 2):
- for the case discussed so far, i.e. the case of N-1 points of reference where user is not referencing itself, we obtain:

\[
R_{ij}^{*} = \sum_{r=N-j}^{N} (-1)^{r-N+j} \binom{r}{N-j} \binom{N-r}{r} \frac{i}{N(N-1)} \sum_{s=0}^{i} \binom{i}{s} (N-r-1)^{i-s} r^{s}
\]

The calculation of \( R_{ij}^{*} \) for this case is given in Appendix II.

- for the second case, i.e. there are N points of reference e.g. memories, we obtain:

\[
R_{ij}^{*} = \sum_{r=N-j}^{N} (-1)^{r-N+j} \binom{r}{N-j} (1 - \frac{r}{N})^i
\]

The calculation of \( R_{ij}^{*} \) for this case is given in Appendix III.

Given \( R_{ij}^{*} \) we can compute the state probabilities \( p_i \), where

\[
p_i = \lim_{t \to \infty} Pr(n=i)
\]

which are given by the solution of equations

\[
\begin{align*}
    p \cdot S &= p \\
p \cdot T &= 1
\end{align*}
\]

(13)

where \( S = (S_{ij}) \) and \( S_{ij} = Pr(n+1=j|n=i) \).

To compute \( S_{ij} \) we shall assume that state \( i \) is reached from state \( j \) in the following two-step transitions: at the beginning of the service interval we are in state \( i \), i.e., we have \( i \) messages queued for service. After choosing, in zero time, the messages for transmission state \( k \) is reached. Following new message arrivals during the service time interval, we reach state \( j \), \( j \geq k \) at the end of the interval:

\[
S_{ij} = \sum_{k=0}^{M} c_{ik} a_{kj}
\]

or

\[
S = C \cdot A
\]

(14)
Matrices $C$ and $A$ have dimensions $(M+1) \times (M+1)$ and are given as follows:

$$C := c_{ik}$$

$c_{ik}$ - the probability of reaching state $k$ from state $i$ by choosing $i-k$ messages for transmission:

$$c_{ik} = \begin{cases} 1 & i = 0,1 \text{, } k = 0 \\ R_{i,i-k} & b+1 \geq i > k \text{ or } k \geq i-b, i > b+1 \\ 0 & \text{otherwise} \end{cases}$$ (15)

$$A := a_{kj}$$

$a_{kj}$ - the probability of reaching state $j$ from state $k$ following arrival of $j-k$ messages, during the unit slot (service time interval):

$$a_{kj} = \begin{cases} j-k-1 & j = M \\ 1 - \sum_{k=0}^{j-k-1} \pi_k & M > j \geq k \\ \pi_{j-k} & \text{otherwise} \\ 0 & \text{otherwise} \end{cases}$$ (16)

From (13) we have:

$$P_n = P_0 \pi_n + P_1 \pi_n + \sum_{i=2}^{b} \sum_{i=k=\max(i-n,1)}^{\min(n+b,M)} R_{ik} \pi_{n-i+k} \sum_{i=b+1}^{\max(i-n,1)} P_i R_{ik} \pi_{n-i+k}$$

for $M-1 \geq n \geq 0$ (17)

and so, we can now finally obtain the steady state probabilities $p_i$'s from the set of linear equations (17). From here we get $\alpha$, $L$ and $W$ as explained in Section 4.
The multiple dependence on the various states prevents us from using iterative techniques for solving the set of equations (17).

The matrix equations were solved by Gauss elimination method using Crout algorithm [9] with equilibration and partial pivoting.

For purposes of accuracy double precision was used in all phases of computation.

6. RESULTS

Figures 1-8 and Tables I-IV summarize results obtained from the models introduced in the previous sections.

Table I compares results obtained from the exact and the approximate models for a small 3x2x2 system. The error introduced by the approximation is shown to be very small. As system size grows, exact models become extremely complex and approximation is used. Tables II-IV compare results obtained from approximation and simulation. From these results we see that the approximate model provides a lower bound on the average message queueing delay. This observation is clearly supported by intuition since the random redrawing of message destinations at the beginning of each service interval tends to relieve queue accumulation which occurs in the actual system. The approximation model is therefore optimistic.

Figure 1 shows that, as expected, for the same total system load factor $\rho = \lambda/\mu$, the expected queueing delay decreases as the number of buses increases. Figure 2 shows the expected queueing delay versus bus traffic intensity, or utilization, $\rho' = \lambda/b\mu$. Figure 2 also gives an interesting upper limit on the number of buses which can be effectively
added to a system. Specifically, Figure 2 demonstrates that contrary to intuition for a given bus utilization and a given system size $N$, when $b$ approaches $N$ the introduction of additional buses to the system increases the average queueing delay. This effect can be explained as follows: to achieve a given bus utilization $\rho'$, the system load $\rho$, $\rho = bp$ grows with $b$ - the number of system buses. When the number of buses $b$ approaches the number of destinations $N$, additional buses cannot be fully utilized due to destination conflicts. As a consequence for a given $\rho'$, the addition of buses beyond a certain number increases the average queueing delay. When the probability of destination conflicts is reduced, i.e., the ratio $b/N$ becomes smaller, the addition of buses will improve the average queueing delay as expected, see Figure 3.

Figures 4 and 5 show the buffer overflow probability as a function of the buffer size for an increasing number of buses. For a given bus traffic intensity $\rho'$ (Figure 4), or for a given system load factor $\rho$ (Figure 5) the overflow probability decreases exponentially with buffer size.

Figures 6 and 7 give the expected queueing delays as a function of the buffer size $M$. Figure 6 plots-the queueing delay as a function of constant bus utilization, $\rho'$ while in Figure 7 the queueing delay is given for a constant value of total system load, $\rho$. Both figures provide delay curves for different numbers of buses. To relate the number of buses $b$, number of buffers $M$ and the average queueing delay $W$ to the buffer overflow probability $P_{of}$ we further plot curves connecting those values of $(b,M)$ for which the buffer overflow probabilities remain equal.
Figure 6 shows that to satisfy an upper bound on the buffer overflow probability for a given bus utilization (and queueing delay) an increase in the number of buses $b$ must be accompanied by an increase in the number of buffers, $M$. This again is due to the fact that an increase in the number of buses for a constant bus utilization $\rho'$, implies a higher total system load $\rho$. Figure 6 thus provides information on the system behaviour as a function of the bus traffic intensity. In other words, in a system where $M$ and $b$ are given it can be used to establish favorable operating conditions by setting $\rho'$ (e.g. by controlling the traffic flow from users) when constraints on the expected queueing delay $W$ and the buffer overflow probability $P_{of}$ are given.

Alternatively, under the same $W$ and $P_{of}$ constraints we may wish to design a system in terms of $b$ and $M$ when the total system traffic load $\rho$ is given. This information is provided in Figure 7. Figure 7 shows the existence of a tradeoff between the number of buses and the number of buffers. We see that to obtain the same $P_{of}$ for a given $\rho$, $W$ and $N$, we can either increase the number of buses and decrease the number of buffers or vice versa.

We can consequently look for the least expensive combination of $b$ and $M$ which satisfies the constraints on $W$ and $P_{of}$ for a given system $(\rho, N)$. As an example, let us consider a system consisting of 12 users with a message arrival rate, or system load, of 2.5 messages/unit of time. The system is to be designed in such a way that the overflow probability is less than $10^{-4}$ and the average queueing delay less than 0.71 units of time. Let us assume for the sake of simplicity that we have the following linear cost function:

$$C = C_b \cdot b + C_M \cdot M$$
with $C_b = 10, C_M = 1$, the bus and buffer cost respectively.

Figure 8 shows the curves representing the average queueing delay and the overflow probability constraints given in the examples together with the cost function plotted against the two slack variables $b$ and $M$. The actual curves are, of course, discrete and consequently the best system design will be $(b,M) = (4,13)$.

CONCLUSIONS

In this paper we have investigated a multi-user system with multiple channels communicating via a central, finite size communication buffer. The effects of channel contention, destination conflicts and buffer overflow probability on the system behaviour were studied. We have built exact and approximate analytic models and have demonstrated their application to various systems and system reference models. It has been shown that these models can be used in distributed system design. Given system performance constraints we have demonstrated how results obtained from analysis can provide information on minimal cost system design in terms of the number of required buses and the communication buffer memory.
APPENDIX I

To derive an expression for the number of states of the exact lumped chain, let \( p_k(n) \) be the number of unordered partitions of \( n \) messages into \( k \) destinations assuming that at least one message exists for every destination. This is equivalent to the number of unordered partitions of \( n \) into \( k \) parts with \( k \) and \( n \) integers. \( p_k(n) \) can, therefore, be obtained from the following recurrent relation:

\[
p_k(n) = p_k(n-k) + p_{k-1}(n-k) + \ldots + p_1(n-k) + p_0(n-k)
\]  
(A1.1)

with

\[
p_k(n) = 0, \quad \text{if } n < k, \ k < 0
\]

\[
p_0(n) = 0, \quad \text{if } n > 0
\]  
(A1.2)

\[
p_k(k) = 1, \quad \text{if } k \geq 0
\]

The level of a state \( \ell \) is defined as the number of packets in the system.

The number of states at level \( \ell \) is:

\[
B(\ell) = \sum_{i=0}^{\ell} p_i(\ell)
\]  
(A1.3)

The number of states in system, thus becomes:

\[
S = \sum_{j=0}^{M} B(j) = \sum_{j=0}^{M} \left( \sum_{i=0}^{j} p_i(j) \right)
\]  
(A1.4)
APPENDIX II

We calculate $R_{ij}$ the probability that among $i$ messages in the queue there are exactly $j$ messages with different destinations i.e., reference points and none of them originated at one of the points.

In other words, we deal with the case where the destinations are the users themselves.

We define $p_k(N,i)$ the probability that there are exactly $k$ users to which there are no queued messages. We let $A_i$ be the event that there is no message to the $i$-th user. The probability that there is no message to $k$ specific users (i.e. the destinations of all $i$ messages are among $N-k$ users) is:

$$
\beta_k(N,i) = p(A_{i_1} \cap \ldots \cap A_{i_k}) = \prod_{s=0}^{i} p \left( \text{the source of (i-s) messages is between (n-k) users and the source of s messages is between k users} \right) \cdot p \left( \text{the destinations of (i-s) messages is between (n-k) users} \right) \cdot p \left( \text{the destinations of s messages is between (N-k) users} \right)
$$

$$
= \prod_{s=0}^{i} \left( \frac{i}{N} \right)^{i-s} \left( \frac{k}{N} \right)^s \left( \frac{N-k}{N-1} \right)^{i-s} \left( \frac{N-k}{N-1} \right)^s
$$

$$
= \left( \frac{N-k}{N(N-1)} \right)^i \sum_{s=0}^{i} \left( \frac{i}{(N-k-1)^{i-s} k^s} \right)
$$

We now compute the probability $p_k(N,i)$ that there are exactly $k$ users to which there are no queued messages.

We apply the formula:

$$
p_k(N,i) = \sum_{r=k}^{N} \left( -1 \right)^{r-k} \binom{r}{k} S_r
$$

where

$$
S_r = \binom{N}{r} \beta_r(N,i)
$$
Thus

\[ P_k(N,i) = \sum_{r=k}^{N} (-1)^{r-k} \binom{r}{k} \binom{N}{r} \frac{(N-r)}{N(N-1)} \sum_{s=0}^{i} \binom{i}{s} (N-r-1)^{i-s} r^s \]

The probability \( R'_{ij} \) that among \( i \) messages in queue there are exactly \( j \) messages with different destinations is equivalent to the probability that there is no message to exactly \( N-j \) users. Thus

\[ R'_{ij} = P_{N-j}(N,i) \]

or

\[ R'_{ij} = \sum_{r=N-j}^{N} (-1)^{r-N+j} \binom{r}{N-j} \binom{N}{r} \frac{(N-r)}{N(N-1)} \sum_{s=0}^{i} \binom{i}{s} (N-r-1)^{i-s} r^s . \]
APPENDIX III

We calculate $R_{ij}^k$ - the probability that among $i$ messages in the queue there are exactly $j$ messages with different destinations, and any user can access any destination, i.e., we deal with the case where there are $N$ points of reference, we may assume memories, for each user.

We define $p_k(N, i)$ to be the probability that there are exactly $k$ memories to which there are no queued messages.

We let $A_i$ be the event that there is no message to the $i$-th memory, i.e., the destinations of all $i$ messages fall among $N-1$ memories.

$$p(A_i) = (1 - \frac{1}{N})^i,$$

and, in general, if $1 \leq i_1 < i_2 < \ldots < i_k < N$ then the event $A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_k}$ occurs if and only if the destinations of the $i$ messages are among $N-k$ memories. Consequently:

$$p(A_{i_1} \cap \ldots \cap A_{i_k}) = (1 - \frac{k}{N})^i.$$

We now compute the probability $p_k(N, i)$ that there are exactly $k$ memories to which there are no queued messages.

We apply the formula:

$$p_k(N, i) = \sum_{r=k}^{N} (-1)^{r-k} \binom{r}{k} S_r,$$

where

$$S_r = \binom{N}{r} \cdot (1 - \frac{r}{N})^i$$

$$p_k(N, i) = \sum_{r=k}^{N} (-1)^{r-k} \binom{r}{k} \binom{N}{r} \cdot (1 - \frac{r}{N})^i.$$
Notice, however, that the probability $R'_{ij}$ is equivalent to the probability that there are exactly $N-j$ memories to which there are no queued messages and so:

$$R'_{ij} = p_{N-j}(N,i).$$

Using the expression for $p_k(N,i)$ we obtain:

$$R'_{ij} = \sum_{r=N-j}^{N} (-1)^{r-N+j} \binom{N}{r} \binom{N-j}{r} (1 - \frac{r}{N})^i.$$
TABLE I - Average queueing delay versus load factor for a $3 \times 2 \times 2$ system - analytic results: Exact and approximate models.

<table>
<thead>
<tr>
<th>$\rho = \frac{\lambda}{\mu}$</th>
<th>exact model</th>
<th>approximate model</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.50169</td>
<td>0.50169</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.51934</td>
<td>0.51935</td>
<td>0.0019</td>
</tr>
<tr>
<td>0.5</td>
<td>0.57253</td>
<td>0.57254</td>
<td>0.0017</td>
</tr>
<tr>
<td>0.75</td>
<td>0.64290</td>
<td>0.64290</td>
<td>0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.75322</td>
<td>0.75322</td>
<td>0</td>
</tr>
<tr>
<td>2.0</td>
<td>0.88836</td>
<td>0.88836</td>
<td>0.00058</td>
</tr>
</tbody>
</table>

TABLE II - Average queueing delay comparison for a $12 \times 6 \times 2$ system. Approximate and simulation models.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>simulation</th>
<th>approximation</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.06</td>
<td>0.4997</td>
<td>0.5031</td>
<td>-0.67</td>
</tr>
<tr>
<td>0.12</td>
<td>0.5123</td>
<td>0.5068</td>
<td>-1.07</td>
</tr>
<tr>
<td>0.9</td>
<td>0.6722</td>
<td>0.6601</td>
<td>-1.80</td>
</tr>
<tr>
<td>1.2</td>
<td>0.8511</td>
<td>0.8134</td>
<td>-4.42</td>
</tr>
<tr>
<td>1.7142</td>
<td>1.4524</td>
<td>1.3726</td>
<td>-5.49</td>
</tr>
</tbody>
</table>
### TABLE III - Average queueing delay comparison for a 12x12x2 system. Approximate and simulation models.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>Simulation</th>
<th>Approximation</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.06</td>
<td>0.4997</td>
<td>0.5030</td>
<td>0.65</td>
</tr>
<tr>
<td>0.12</td>
<td>0.5123</td>
<td>0.5070</td>
<td>-1.03</td>
</tr>
<tr>
<td>0.9</td>
<td>0.6739</td>
<td>0.6606</td>
<td>-1.97</td>
</tr>
<tr>
<td>1.2</td>
<td>0.8656</td>
<td>0.8240</td>
<td>-4.80</td>
</tr>
<tr>
<td>1.7142</td>
<td>1.8753</td>
<td>1.7615</td>
<td>-6.08</td>
</tr>
</tbody>
</table>

### TABLE IV - Average queueing delay comparison for a 12x20x2 system. Approximate and simulation models.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>Simulation</th>
<th>Approximation</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.06</td>
<td>0.4997</td>
<td>0.5044</td>
<td>0.93</td>
</tr>
<tr>
<td>0.12</td>
<td>0.5123</td>
<td>0.5073</td>
<td>-0.97</td>
</tr>
<tr>
<td>0.9</td>
<td>0.6739</td>
<td>0.6606</td>
<td>-1.97</td>
</tr>
<tr>
<td>1.2</td>
<td>0.8656</td>
<td>0.8240</td>
<td>-4.80</td>
</tr>
<tr>
<td>1.7142</td>
<td>1.9899</td>
<td>1.9142</td>
<td>-3.80</td>
</tr>
</tbody>
</table>
Figure 1 - Average queueing delay vs. the load factor for a 12×8×b system. Approximation results.
Figure 2. - Average queueing delay vs. the traffic intensity for a 8×8×b system. Approximation results.
Figure 3 - Average queueing delay vs. the traffic intensity $\rho'$ for a 16x12xb system. Approximation results.
Figure 4 - Buffer overflow probability vs. number of buffers for a 12×M×b system. Approximation results.
Figure 5 - Buffer overflow probability vs. number of buffers for a 12xMxb system and a given system load $\rho$. Approximation results.
Figure 6 - Average queueing delay vs. buffer size for a 12×M×b system and a given bus traffic intensity, $p'$. Approximation results.
Figure 7 - Average queuing delay vs. number of buses for a 12xMxb system and a given system load, $\rho$. Approximation results.
Figure 8 - Design curves for a 12×M×b system for a given (maximum) system load $\rho$, buffer overflow probability, $P_{of}$, and queueing delay $W$ and cost functions $C$. 

$\rho = 2.5$

$P_{of} = 10^{-4}$

Buffer Size, $M$

Number of Buses, $b$
REFERENCES


