INTERCONNECTION SYSTEM DESIGN CONSIDERATIONS
FOR HIGH-AVAILABILITY NETWORKS

by

O. Ganz and I. Chlantac

Technical Report #277
June 1983
ABSTRACT

This study considers multi-user distributed systems connected over a limited area through multiple-buses using partially duplicated bus interfaces.

Exact and approximate Markovian models are introduced which show the effect of bus multiplication, destination conflicts and of bus-synchronization on the interconnection system performance.

Alternative system configurations are analyzed and compared. The presented results can be used for design and optimization of high-availability distributed systems, such as loosely coupled cluster systems or backend storage networks to determine, for instance, the number of required buses, for given messages delay constraints.

Key Words: High availability distributed systems; multiple-buses interconnection; bus contention; performance modeling.
Distributed systems have experienced rapid growth in recent years. One of the emerging trends in these systems is the introduction of highly available multibus oriented interconnection systems, not only in tightly coupled memory/process systems but also in loosely coupled systems [10, 2] and local area computer networks, [3, 4, 6]. The multiple bus configurations offer easy system extendability, high availability, and incremental system growth. Furthermore, using several low-speed buses the cost of the node interfaces is lower than in a single high-speed channel interconnection system.

In a single bus configuration the bus insertions are passive but they cannot be used with state of the art technology if transmission speeds higher than few tens of Mbps are desired. The presence of a large number of taps in the bus limits the product of the end-to-end distance by the data rate to approximately 5 Mbps km [8], so that increasing the data rate the network span is reduced and vice versa. A future solution for increasing the transmission speed are the fiber optics but there are still technological problems. Another disadvantage of a single global bus configuration is the weak reliability. To circumvent this difficulty many systems use redundant communication buses in parallel, simultaneously conveying the same message [5].

The use of multiple buses can circumvent both disadvantages, i.e., relatively low speed and weak reliability.

Local network systems using multiple buses have been recently analyzed [6, 7, 8].
which an infinite population of users can access each bus through a dedicated interface. In [7], a hyperchannel multibus network is analyzed using an interface by which each user can be either receiving or transmitting on a single bus only. In both cases CSMA protocol variants are used to control the access to any one of the channels.

In loosely coupled systems the design and analysis considerations are different. In these systems the distances and the number of users are usually small. Consequently, in such systems the overhead necessary for scheduling collision-free channel access can be tolerated [10,14]. Since, however, practical considerations lead to interfaces which are only partially duplicated [6,10], contention situations which can significantly affect performance can still arise. If, for instance, a user can be receiving on a single bus at a time a destination conflict will occur if more than one node are attempting to pass a message to the same node.

In this paper, we consider a distributed system with a finite number of users connected over a limited area, e.g. a cluster system [10] or backend storage network [3,4,6]. We build our model around assumptions which reflect prevailing design considerations of existing systems and analyze the system behaviour.

We consider the effect of conflicts and of bus synchronization on system performance and show how for a given system model the number of system buses can be chosen so that a guaranteed delay can be provided.
homogeneous devices interconnected through multiple buses as shown in Fig. 1. The devices are connected to the buses through bus interfaces also called adapters. Each adapter is connected to every system bus but a node can transmit on only one of the buses and receive on another simultaneously. This assumption is made to obtain models of prevailing multiple bus systems [6,10]. Although the system can be analyzed in a similar way under other interconnection disciplines.

We further assume a close system model whereby generation of messages is done at a node only when its single transmission buffer is empty. This model has been shown to closely represent the communication activity in similar systems [1,6,11].

With these assumptions, at a given time a node can be in one of three different states:

1) the node can generate messages, i.e. is active;
2) the node is transmitting the message;
3) the node has a message queued for transmission.

Regarding the system operation we make the following assumptions:

1) The message arrivals to the active nodes are assumed to be generated from a Poisson process with average arrival rate \( \lambda_i \) for the i-th station
   
   \[ \lambda_i = \lambda \quad \forall i \]

2) A uniform reference model is assumed; this implies that every bus access request from every node is directed to any other node with equal probability. Thus, the access rate from node \( i \) to node \( j \) is derived as:

   \[ \lambda_{ij} = \frac{\lambda}{N-1} \quad \forall i,j \quad i \neq j \quad (1) \]

   \( N \) - the number of nodes
1) Synchronous

2) Asynchronous

In the synchronous system messages are of constant size with constant service time $\frac{1}{\mu}$. Up to $b$ messages with different destinations can be randomly selected for service at the beginning of a time slot.

In the asynchronous system messages length are an independent exponentially distributed random variable with mean $\frac{1}{\mu}$. When a bus goes idle the next node to use the bus is selected at random among nodes wishing to transmit to a node which has become free.

Several messages can be arriving at an adapter simultaneously. When this occurs the receiver accepts only one of the messages and rejects the other [6,10].

We assume that the message selection for transmission process is done in zero time. This assumption is viable since a number of protocols can provide this service in total delay equal approximately to the cable propagation delay [6,12,13,14]. For loosely coupled multiprocessor systems or backend storage networks this delay is on the order of several microseconds.

For analysis of both systems it is possible to construct a closed queueing network as shown in Fig. 2. Messages generated at the nodes join queues and must be granted access right to the bus for being transmitted. In the synchronous system up to $b$ access rights are given at the beginning of a time slot while in the asynchronous system a permission is given whenever there is an idle bus and a message whose destination is free.
The permit is returned upon the completion of a transmission.

To derive the performance measures, 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 \( \lambda^* \) to be the rate at which messages cycle through the queueing network.

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

Applying Little's result, we obtain the average access time \( D \):

\[
D = \frac{L}{\lambda^*}
\]

and the average queueing delay \( W \) by subtracting from \( D \) the service time \( \frac{1}{\mu} \):

\[
W = \frac{L}{\lambda^*} - \frac{1}{\mu}.
\]

The average queueing delay results from contending for the bus with other nodes or from contention created by destination conflict, i.e., messages who have the same destination and only one of them can be chosen.

The average number of queued messages will be:

\[
L_q = W \cdot \lambda^*
\]

We further derive \( \lambda^* \) and \( L \) for both systems as follows:

a) Synchronous system:

The rate \( \lambda^* \) is given by:

\[
\lambda^* = (N - \sum_{i=0}^{N} i p_i)(1-e^{-\lambda})
\]

obtained by multiplying the expected number of active nodes at the beginning of a service interval by the probability of message generation during the time interval.
b) For the asynchronous system we have similarly

\[ \lambda^* = (N-L) \lambda \]

and

\[ L_i = \sum_{i=0}^{N} i p_i \]
Figure 1 - Block diagram of a system with N nodes and b buses.

Figure 2 - Closed queueing network model of the system

Number of issued permits depends on the number of buses, number of queued messages and their destinations distributions.
3. **EXACT MODELS**

a) **Synchronous System**

Under previous assumptions 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 exact chain is given by:

\[ \{q_1, q_2, \ldots, q_N\} \]

where

\[ q_i \] is the number of messages waiting for a station at the end of a service interval, arranged in decreasing order.

The number of states of the exact lumped chain is

\[ \mathcal{S} = \left( \sum_{j=0}^{N} \sum_{i=0}^{j} p_1(j) \right) - 1 \]

where \( p_1(j) \) is the number of unordered partitions of \( j \) messages into \( \delta \) destinations (nodes) assuming that at least one message exists for every destination.

The expression for \( \mathcal{S} \) is derived in Appendix I. In Appendix II we give an example of the exact model.

b) **Asynchronous Model**

Using the assumptions of the previous section we construct a Markov chain to model the behaviour of the system.

The state definition of the exact Markov chain is

\[ (n_N, q_1, q_2, \ldots, q_N) \]

where

\[ n_N \] is the number of nodes currently transmitting
An example of the state transition diagram of the Markov chain is given in Fig. 3 for a 3×2 system.

Figure 3 - State transition diagram for a 3×2 asynchronous system

An increase in the number of nodes complicates the Markov chain, therefore the general N×b. case is not easy to handle.
4. APPROXIMATE MODELS

To significantly reduce the model complexity we build approximate models by reducing the amount of information recorded in each state, while preserving the Markovian property.

In the exact model the states recorded complete information about queues inside the system, i.e., the number of messages queued for each node.

a) Synchronous System.

In the approximate 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:

\[(n) \quad \text{ (11)}\]

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

To evaluate the transition rates without the knowledge of the interval system queues we assume that at the beginning of the service time interval each queued node reselects a new destination station at random and 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).

Let \(\pi_{ij}\) as defined in Appendix I, be the probability of exactly \(i\) messages originating from a Poisson process in a service time interval \((u - 1)\) when it is given that at the beginning of the interval there are \(j\) messages in system.
Let $p_i$ be the probability that there are exactly $i$ messages in the system at the end of a service time interval.

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 $R_{i,j}$ 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.

$$R_{i,j} = \begin{cases} 
1 - \sum_{k=1}^{b-1} R_{ik} & j = b \\
R_{ij} & j < b 
\end{cases} \quad (13)$$

For $R_{i,j}$ we obtain

$$R_{i,j} = \sum_{r=N-j}^{N} \frac{(-1)^{r-N+j}}{(N-j)!} \frac{r^i}{r!} \frac{\min(i,r)}{i+r} \left( \frac{N-r-1}{N-1} \right)^{i-s} \left( \frac{N-r}{N-I} \right)^{j-r} \quad (14)$$

The calculation of $R_{i,j}$ is given in Appendix III.

To establish the steady-state set of equations we assume that the system has reached its equilibrium.

Without loss of generality we can again let the service interval be equal to unity. We shall express the probability of the number of messages present in system at the end of a unit service time interval (left side of equation (15)) in terms of the probability of the number present in system at the beginning of the interval multiplied by the probability of reaching the number in system at the end of a service interval for all possible transition combinations (right side of equation (15)).
\[ P_0 = P_{00}^{n0} + \sum_{i=2}^{b} \frac{b}{p_i} \right] \\
\[ P_1 = P_{01}^{n1} + \sum_{i=2}^{5} \frac{b}{p_i} \right] \\
\[ P_n = P_{0n}^{nk} + \sum_{i=2}^{b} \frac{b}{p_i} \right] \\
\] for \( n \leq N-1 \)

(15)

The multiple dependence on the various states prevents us from using iterative techniques for solving the set of equations (15).

Thus the set of probabilities \( p_i \)'s must be solved from the set of linear equations (15). We get the performance measures as explained in Section 3 by substituting the \( p_i \)'s.

The size of the matrix corresponds to the number of processors. The matrix equations were solved by Gauss elimination method (Crout algorithm [9]) with equilibration and partial pivoting. For purposes of accuracy double precision was used in all phases of computation.

b) Asynchronous System

We define the system state as the number of messages in system. Again no account is kept on the state of internal queues. The transition rates are evaluated using the averaging technique.

The transition diagram in Fig. 4 is for a \( N \times b \) system.
We define \( a_i \) as the ratio between the sum of transition rates from all states with \( i \) messages to states with \( i-1 \) messages, and the number of states with \( i \) messages in the exact chain, which is derived in Appendix IV.

\[
\begin{align*}
\alpha_i &= \begin{cases} 
\frac{\sum_{j=1}^{b-1} j p_j(i) + b \sum_{j=0}^{i-b} [p_b(j+b)p_{N-b}(i-2b+j+N)]}{\sum_{j=0}^{i-b} [p_b(j+b)p_{N-b}(i-2b+j+N)]} & \text{for } 0 < i < N \\
\frac{\sum_{j=1}^{b+1} j p_j(i) - 1 + b \sum_{j=0}^{i-b} [p_b(j+b)p_{N-b}(i-2b+j+N)]}{\sum_{j=0}^{i-b} [p_b(j+b)p_{N-b}(i-2b+j+N)]} & \text{for } 0 = i < N \\
\frac{\sum_{j=1}^{b-1} j p_j(i) - 1 + b \sum_{j=0}^{i-b} [p_b(j+b)p_{N-b}(i-2b+j+N)]}{\sum_{j=0}^{i-b} [p_b(j+b)p_{N-b}(i-2b+j+N)]} & \text{for } b > i > 1 \\
\end{cases}
\end{align*}
\]
We solve the system like a birth-death process:

\[
\lambda_i = \begin{cases} 
\lambda(N-i) & 0 \leq i < N \\
0 & \text{otherwise}
\end{cases}
\]  

(17)

\[
\mu_i = \mu a_i.
\]

(18)

We use the formulas:

\[
p_k = p_0 \prod_{i=0}^k \frac{\lambda_i}{\mu_i + 1},
\]

(19)

\[
p_0 = \frac{1}{1 + \sum_{k=1}^N \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_i + 1}}.
\]

(20)

We substitute \( \lambda_i \) and \( \mu_{i+1} \) from formulas (17) and (18) in (19) and (20), and we obtain:

\[
p_i = p_0 \prod_{k=1}^i \frac{\lambda(N-i)}{\mu a_k},
\]

(21)

\[
p_0 = \left(1 + \sum_{j=1}^N \frac{(\rho)^j}{(N-j)!} \frac{N!}{\prod_{k=1}^j \mu a_k} \right)^{-1},
\]

(22)

where \( \rho = \frac{\lambda}{\mu} \) is the loading factor.
Table I compares results obtained from the exact and the approximate models for a small $3 \times 2$ system. The error introduced by the approximation is shown to be very small.

Tables II-III compare results obtained from approximation and simulation for the synchronized system. 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.

Tables IV-V compare results obtained from approximation and simulation for the unsynchronized system. We see that the approximation obtained by averaging technique gives an upper bound on the average queueing delay. The bound seem to be rather tight since percentage errors well below 10 percent were typically observed.

Secondly, we observe that the errors tend to be smaller, the larger the system is. This may be due to the fact that the averaging technique gives better results for higher number of states.

Figure 4-6 demonstrate the effect of multiple buses on performance. From these figures we observe that the networks achieve a significant gain in performance with the addition of a second bus while the benefit realized is less for each added bus. This can be explained by the fact that when the number of buses $b$ approaches
the number of destinations $N$, additional buses cannot be fully utilized due to destination conflicts, unless the number of users is sufficiently large.

Figure 6 plots the average queueing delay versus the load factor $\rho = \frac{\lambda}{\mu}$ with different number of buses ($b = 2, 3, 4$) for the two systems. We observe that at a certain load factor there is a crossing point between the curves of the two systems. For loads below this point the average queueing delay for the synchronous system is higher than for the asynchronous system. This is due to the fact that for low load factor in the synchronous system we have to wait in average half a slot. For higher load factor the average queueing delay for the synchronous system becomes lower because of the variance of the message length.
SUMMARY

In this paper we investigated the behaviour of local networks or other loosely coupled systems with multiple bus interconnection systems. We have built exact and approximate analytic models and have demonstrated their application to various systems design.

We have concentrated our attention on system models which closely resemble the design considerations of emerging systems so that the results can benefit current and future designs. Given system performance constraints we have obtained results which provide information on minimal cost system design in terms of the number of required buses.

It has been further shown how the system size in terms of number of users affects performance - a consideration especially important for small systems [10,11]. The analysis included synchronous and asynchronous systems.

Since both system types are found in practice this investigation is useful to evaluate their relative behaviour in face of the unique tradeoffs existing in multiple bus interconnection mechanism, resulting from the different hardware requirements placed by synchronous versus asynchronous transmission selection protocols.

Combined with hardware considerations, performance and reliability constraints, this study can be thus helpful in the interconnection mechanism design process.
APPENDIX I

To derive an expression for the number of states of the exact chain, we define \( p_k(n) \) the number of unordered partitions of \( n \) into \( k \) parts with \( k \) and \( n \) integers. \( p_k(n) \) is obtained from the following recurrent relation:

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

with

\[
p_k(n) = n < k \quad \text{or} \quad k > 0
\]

\[
p_0(n) = n > 0
\]

\[
p_k(k) = 1 \quad \text{for} \quad k \geq 0.
\]

The level of a state \( l \) is defined as the number of messages in the system. The number of states at level \( l \) is:

\[
B(l) = \sum_{i=0}^{l} p_i(l)
\]

The number of states in the system, thus becomes:

\[
S = \left( \sum_{j=0}^{N} B(j) \right) - 1 = \sum_{j=0}^{N} \sum_{i=0}^{j} p_i(j) - 1.
\]
In this appendix we give an example of two buses.

Let \( \pi_{i,j} \) be the probability of exactly \( i \) messages originating from a Poisson process in a service time interval \( (T=1) \) when it is given that at the beginning of the service time interval there are \( j \) messages in system.

\[
\pi_{i,j} = \binom{N-j}{i} (1-e^{-\lambda})^i (e^{-\lambda})^{N-j-i}
\]

This is a binomial trial where \( i \) messages are originating from \( N-j \) active nodes and the probability of a message to be born is \( (1-e^{-\lambda})^i \). Recall that the interval between subsequent access requests is exponentially distributed with mean \( 1/\lambda \).
Thus we have to find the state probabilities $p(q_1, q_2, q_3)$ from the following set of linear equations with $q_i^k$ as given in (8).

\[
\begin{align*}
p_{000} &= P_{000} \pi_{0,0} + P_{100} \pi_{0,1} + P_{110} \pi_{0,2} \\
p_{100} &= P_{000} \pi_{1,0} + P_{100} \pi_{1,1} + P_{110} \pi_{1,2} + P_{200} \pi_{0,2} + P_{111} \pi_{0,3} + P_{210} \pi_{0,3} \\
p_{110} &= P_{000} \pi_{2,0} + P_{100} \pi_{2,1} + P_{200} \pi_{1,2} \\
p_{200} &= P_{000} \pi_{2,0} + P_{100} \pi_{2,1} \\
p_{111} &= P_{000} \pi_{3,0} \\
p_{000} + p_{100} + p_{110} + p_{200} + p_{111} + p_{210} &= 1
\end{align*}
\]
We define \( p_k(N,i) \) the probability that there are exactly \( k \) nodes to which there are no queued messages. We let \( A_i \) be the event that there is no message to the \( i \)th node. The probability that there is no message to \( k \) specific nodes (i.e., the destinations of all \( i \) messages are among \( N-k \) nodes) is:

\[
\beta_k(N,i) = p(A_{i_1} \cap \ldots \cap A_{i_k}),
\]

\[
= \min(i,k) \begin{cases} \text{the source of (i-s) messages is between} \end{cases} \begin{cases} \text{the destinations of} \end{cases} \begin{cases} \text{p between (N-k) nodes} \end{cases} \begin{cases} \text{p for s messages is between} \end{cases} \begin{cases} \text{p for (N-k) nodes} \end{cases}
\end{cases}
\]

\[
\begin{align*}
&= \min(i,k) \left( \frac{N-k}{i-s} \right) \left( \frac{k}{s} \right) \left( \frac{i-s}{N-1} \right) \left( \frac{N-k}{N-1} \right) \\
&= \min(i,k) \frac{(N-k)(k)}{i-s} \frac{(i-s)(s)}{N} \frac{(N-k)}{N-1} \frac{i-s}{N-1}
\end{align*}
\]

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

We apply the formula:

\[
p_k(N,i) = \sum_{r=k}^{N} (-1)^{r-k} \binom{N}{r} s_r
\]

where

\[
s_r = \binom{N}{r} \beta_r(N,i)
\]

\[
p_k(N,i) = \sum_{r=k}^{N} (-1)^{r-k} \binom{N}{r} \left( \frac{\min(i,r)}{r} \right) \frac{N-r}{N-1} \frac{i-s}{N-1} \frac{N-r}{N-1} \frac{i-s}{N-1}
\]
exactly \( j \) messages with different destinations is equivalent to the probability that there is no message to exactly \( N-j \) nodes.

Thus,

\[
R_{ij} = p_{N-j}(N,i)
\]

or

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

\[
= \max(0, i-N+r) \binom{N-i}{r-N+i} \binom{N}{r}
\]
In this appendix we give expressions for

at level 2 of the exact chain in the case of $p \times b$ system.

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 as defined in Appendix I. The number of unordered partitions without assuming that at least one message exists for every destination is $p_k(n+k)$.

The level of a state $i$ is defined as the number of messages in the system.

There are two cases for $n_b$ the number of nodes currently transmitting:

a) $n_b < b$. In this case the $i-n_b$ messages are queued for nodes currently accessed.

$$B_s(i) = \sum_{j=1}^{b-1} p_j(i-j+j) = \sum_{j=1}^{b-1} p_j(i), \quad i \neq N$$

If $i = N$ we do not have the state $(1,N-1,0,...)$ therefore the number of states is $B_s(i)-1$.

This is the number of possibilities of unordered partitions of $i-j$ messages into $j$ destinations where $j$ is the number of different destinations.

b) $n_b = b$. In this case the $i-b$ messages are two kind of messages:

i) $j$ messages queued for $b$ nodes currently accessed;

ii) $i-b-j$ messages queued for $N-b$ free nodes but no bus is available.

We take all the possibilities of i) and ii) i.e. all the values $j$ can take. We obtain:
\[
\sum_{j=0}^{i-b} p_b(j+b) p_{N-b}(i-2b+j+N).
\]

The number of states at level \( i \) are:

\[
B(i) = \begin{cases} 
B_s(i) + B_b(i) & \text{if } i \neq N \\
B_s(i) + B_b(i) - 1 & \text{if } i = N
\end{cases}
\]
### TABLE I

Average queueing delay versus load factor (3 x 2 synchronous system) analytic results - exact and approximate models.

<table>
<thead>
<tr>
<th>( \rho = \frac{\lambda}{\mu} )</th>
<th>average queueing delay</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>exact model</td>
<td>approximate model</td>
</tr>
<tr>
<td>0.01</td>
<td>0.5024</td>
<td>0.5024</td>
</tr>
<tr>
<td>0.1</td>
<td>0.5223</td>
<td>0.5227</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5741</td>
<td>0.5794</td>
</tr>
<tr>
<td>1</td>
<td>0.5960</td>
<td>0.6066</td>
</tr>
<tr>
<td>3</td>
<td>0.6050</td>
<td>0.6068</td>
</tr>
</tbody>
</table>

### TABLE II

Average queueing delay comparison for 8 x 3 synchronous system. Approximate and simulation models.

<table>
<thead>
<tr>
<th>( \rho = \frac{\lambda}{\mu} )</th>
<th>average queueing delay</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>simulation</td>
<td>approximation</td>
</tr>
<tr>
<td>0.01</td>
<td>0.5107</td>
<td>0.5042</td>
</tr>
<tr>
<td>0.05</td>
<td>0.5344</td>
<td>0.5207</td>
</tr>
<tr>
<td>0.1</td>
<td>0.5643</td>
<td>0.5412</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7231</td>
<td>0.7081</td>
</tr>
<tr>
<td>1</td>
<td>0.9040</td>
<td>0.8781</td>
</tr>
</tbody>
</table>
TABLE III - Average queueing delay comparison for a 12 x 2 synchronous system: Approximate and simulation models.

<table>
<thead>
<tr>
<th>ρ = \frac{λ}{μ}</th>
<th>average queueing delay</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>simulation</td>
<td>approximation</td>
</tr>
<tr>
<td>0.01</td>
<td>0.5113</td>
<td>0.5050</td>
</tr>
<tr>
<td>0.05</td>
<td>0.5637</td>
<td>0.5510</td>
</tr>
<tr>
<td>0.1</td>
<td>0.6765</td>
<td>0.6550</td>
</tr>
<tr>
<td>0.5</td>
<td>3.0020</td>
<td>2.9590</td>
</tr>
<tr>
<td>1</td>
<td>4.0310</td>
<td>3.9180</td>
</tr>
</tbody>
</table>

TABLE IV - Average queueing delay comparison for a 8x3 asynchronous system. Approximate and simulation models.

<table>
<thead>
<tr>
<th>ρ = \frac{λ}{μ}</th>
<th>average queueing delay</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>simulation</td>
<td>approximation</td>
</tr>
<tr>
<td>0.01</td>
<td>0.0231</td>
<td>0.0231</td>
</tr>
<tr>
<td>0.05</td>
<td>0.1164</td>
<td>0.1169</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2103</td>
<td>0.2157</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7913</td>
<td>0.8303</td>
</tr>
<tr>
<td>1</td>
<td>1.1418</td>
<td>1.1846</td>
</tr>
</tbody>
</table>
TABLE V - Average queuing delay comparison for a \(12 \times 2\) asynchronous system. Approximate and simulation models.

| \(\rho = \frac{\lambda}{\mu}\) | \begin{tabular}{c|c|c} \hline average queueing delay & simulation & approximation \\ \hline 0.01 & 0.0384 & 0.0384 & 0. \\ 0.05 & 0.2285 & 0.2289 & 0.17 \\ 0.1 & 0.5393 & 0.5458 & 1.19 \\ 0.5 & 3.0032 & 3.1122 & 3.50 \\ 1 & 3.9332 & 4.0320 & 2.47 \\ \hline \end{tabular} |

% error
Figure 4 - Average queueing delay vs. the load factor for a $8 \times 8$ synchronous system. Approximation results.
Figure 5 - Average queuing delay vs. the load factor for a $b \times b$ asynchronous system.

Approximation results.
Figure 6 - Average queueing delay vs. the load factor for a $12 \times b$ synchronous and asynchronous systems. Approximation results.


