

AD-A059 508

DELAWARE UNIV NEWARK DEPT OF STATISTICS AND COMPUTER--ETC F/G 12/1
EXPONENTIAL SERVERS SHARING A FINITE STORAGE: COMPARISON OF SPA--ETC(U)
MAY 78 G LATOUCHE

AFOSR-77-3236

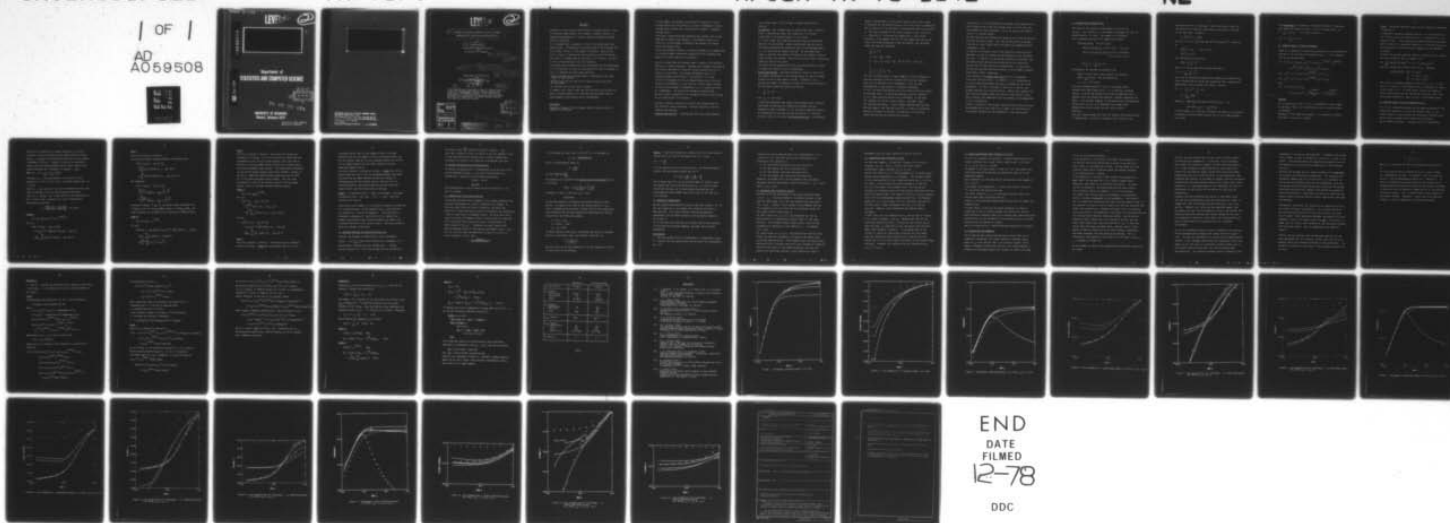
UNCLASSIFIED

TR-78/9

AFOSR-TR-78-1142

NL

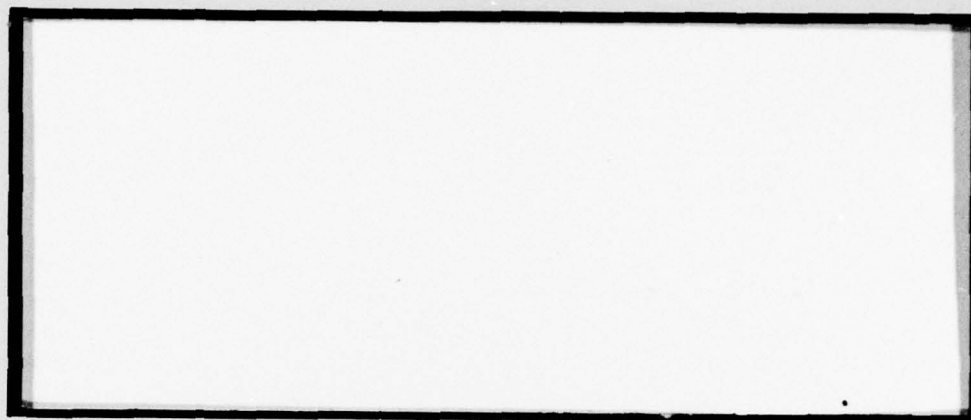
1 OF 1
AD
A059508



LEVEL II

2

AD A059508



DDC FILE COPY

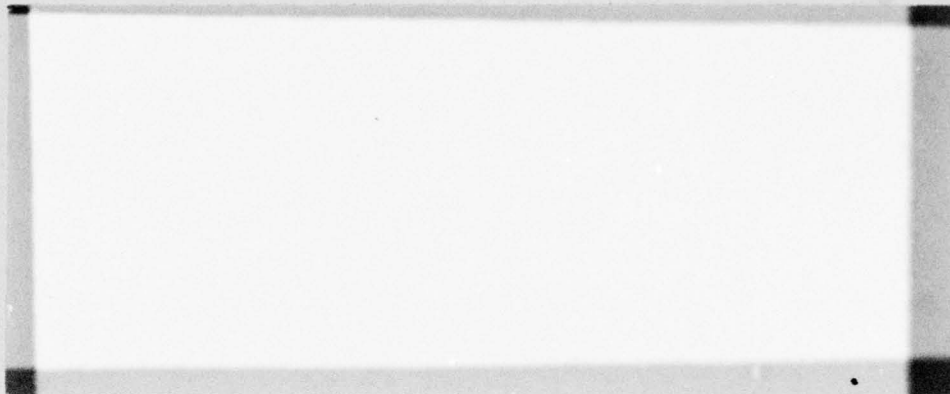
Department of
STATISTICS AND COMPUTER SCIENCE

DDC
RECEIVED
SEP 25 1978
D

78 07 20 151

UNIVERSITY OF DELAWARE
Newark, Delaware 19711

Approved for public release;
distribution unlimited.



AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC

This technical report has been reviewed and is
approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.

A. D. BLOSH
Technical Information Officer

LEVEL II

2

6 EXPONENTIAL SERVERS SHARING A FINITE STORAGE:
COMPARISON OF SPACE ALLOCATION POLICIES.

by

10 Guy/Latouche

Université Libre de Bruxelles
and
University of Delaware

16 2344

17 A5

18 AFOSR 19 TR-78-1142

14 TR-78/9

Department of Statistics
and

9 Computer Science
Technical Report, No. 78/9

15 11 May 1978

12 47p.

This research was sponsored by the Air Force Office of Scientific Research Air Force Systems Command USAF, under Grant No. AFOSR-77-3236. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

ACCESSION for	
DTIC	White Section ☒
DDC	Buff Section ☐
UNANNOUNCED	☐
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

DDC
RECEIVED
SEP 25 1978
D

78 07 20 151
391 807

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

mt

ABSTRACT

Consider a finite waiting room shared by several servers. Such a system may approximate, for instance, a packet switch in a communications network or a buffer shared by producer-consumer parallel processes.

It is assumed that a customer is lost if he cannot enter the waiting room. Furthermore, the waiting room is partitioned into $(L+1)$ areas, where L is the number of servers: each of the first L areas is reserved for the exclusive use of one server, the $(L+1)$ st is common and may be used by all servers. We refer to this allocation policy as $(L+1)$ -sharing.

The steady-state probabilities are determined and the performance of $(L+1)$ -sharing, in terms of the probability that a customer is lost, is compared to three other policies:

- restricted sharing (an upper bound is imposed for the queue length of each server),
- unrestricted sharing (the $(L+1)$ st area occupies the whole waiting room),
- no sharing (the $(L+1)$ st area is empty).

It appears very clearly that the last two policies are not the best; a choice between $(L+1)$ -sharing and restricted sharing would depend on the objectives of the designer.

Key words:

Exponential queue, finite storage, space allocation policy, numerical comparisons

In this paper, we consider the problem of managing a finite storage space used concurrently by several independent servers. Systems which present this characteristic appear in computer related areas.

-Consider a packet-switched communications network such as the ARPA [8] or CIGALE [10] networks. Each packet switch has a store of a finite number of buffers, the buffers are shared by all the output links.

-Parallel processes linked in producer-consumer pairs communicate via a buffer where information generated by the producers may queue until being used by the consumers.

We shall assume that the storage space is made up of M buffers; there are L output processes. Messages are produced from outside the system, destined to a specific output process and can fit exactly in one buffer. We assume furthermore that if an incoming message cannot enter a free buffer, it is lost.

Arrivals of messages to the j th output process form a Poisson process with parameter λ_j ; the time for the j th output process to service one message is exponential μ_j ; all random variables are independent. We analyze the system in steady-state:

We shall defer discussion of our assumptions and of their effect on our conclusions till Section 6.

One has to define a policy to allocate the storage space to the different output processes. We shall describe first the simplest two policies.

Unrestricted sharing: a message may join the system whenever

the storage space is not filled; no other restriction is imposed.

No sharing: the storage space is partitioned into L disjoint areas; a message may enter the system only if the area associated to its output process is not filled.

It is well known that these policies may lead to undesirable behavior for the system. Under unrestricted sharing, one of the output processes may monopolize most of the storage space, if it is very slow or highly utilized. On the other hand, under no sharing, the buffers allocated to an (almost) inactive output process are wasted: they are not used by their process and cannot be used by the others.

Two policies for storage allocation have been defined in order to reduce the impact of such circumstances.

Restricted sharing has been defined by Irland in 1976 [6,7]: the number of messages in the storage space for the j th output process can be at most equal to $K(\leq M)$. Therefore, if k_j denotes the number of messages in queue for the output process j , the following conditions must be satisfied:

$$\sum_{j=1}^L k_j \leq M$$

and $k_j \leq K$, for $1 \leq j \leq L$.

Irland [6] determines the steady-state probabilities, proposes algorithms to evaluate the normalizing constant and the probability that a message is lost and gives numerical results. The second policy has been defined by Dijkstra in 1972 [4].

We shall refer to it as the (L+1)-sharing policy: the storage

space is partitioned in $(L+1)$ areas, each of the first L areas is reserved for one output process, the $(L+1)$ st area is common and can be used by all the processes. A message may enter the system if

- a. the area allocated to its output process is not filled, or
- b. that area is filled and the common area is not filled.

Therefore, if N_0 denotes the size of the common area and N_j the size of the area associated to the j th process, the following conditions must be satisfied:

$$\sum_{j=0}^L N_j = M,$$

$$k_j \leq N_j + N_0, \quad 1 \leq j \leq L,$$

$$k_j + k_{j'} \leq N_j + N_{j'} + N_0, \quad \text{for all } j \neq j', \quad 0 \leq j, j' \leq L,$$

$$\vdots$$

$$\sum_{j=1}^L k_j \leq \sum_{j=1}^L N_j + N_0.$$

The $(L+1)$ -sharing policy is used commonly in the literature on parallel processes (see for instance Devillers and Louchard [3]), but to the best of our knowledge, it has not been analyzed yet in the context of stochastic processes.

If $L=2$, the restricted sharing and $(L+1)$ -sharing policies belong to the same class: if $N_1=N_2$ and $K=N_1+N_0$, they are identical.

If $L>2$, those two policies are quite different, if one does not consider the trivial cases, $N_1=N_2=\dots=N_L=K=M/L$, $N_0=0$ (no sharing) and $N_1=N_2=\dots=N_L=0$, $K=N_0=M$ (unrestricted sharing).

One may roughly rank the four policies, in order of increasing degree of sharing, as follows: no sharing, $(L+1)$ -sharing, restricted sharing and unrestricted sharing.

In Sections 1 to 3, we determine the steady-state probabilities, which always exist since the storage space is finite and also the probability that a message is lost, we analyze the special case $L=2$ in greater detail.

We then restrict our attention to balanced $(L+1)$ -sharing policies: policies for which $N_1=N_2=\dots=N_L (=N)$ and we define an optimal solution as a pair (N_0, N) which minimizes the probability that a message is lost.

In Section 5 we present some of the numerical results we have obtained in order to compare the performances of the different policies. It appears, as was noted already by Irland [6], that the unrestricted sharing and no sharing policies should be omitted from further practical consideration. In particular, the unrestricted sharing policy proves to be extremely unstable under heavy traffic conditions.

Irland defines the square root rule which is a suboptimal restricted sharing policy, i.e. $K \approx M/\sqrt{L}$. It is interesting as it yields good performances and does not depend on the system parameters. We shall introduce in Section 4 a similar sub-optimal balanced $(L+1)$ -sharing policy: $N \approx M/(L+\sqrt{L})$, $N_0 \approx M/(1+\sqrt{L})$. The relations between those four policies (two optimal within their classes, two suboptimal) are too complex to be summarized in this introduction.

Finally in Section 6, we present some conclusions and we discuss the assumptions we have made in our model, the limitations of the present approach and the robustness of our conclusions.

1. Steady-state Probabilities

The state of the system is represented by a vector $\underline{k} = (k_1, k_2, \dots, k_L)$, where k_j is the number of messages for the j th output process (in short: the number of j -messages).

Let S represent the set of all admissible states:

$$\begin{aligned} S = \{ \underline{k} \mid & 0 \leq k_j \leq N_j + N_0 \quad \text{for all } 1 \leq j \leq L, \\ & 0 \leq k_j + k_{j'} \leq N_j + N_{j'} + N_0 \quad \text{for all } j \neq j', \quad 1 \leq j, j' \leq L; \\ & 0 \leq k_j + k_{j'} + k_{j''} \leq N_j + N_{j'} + N_{j''} + N_0 \quad \text{for all different} \\ & \quad j, j', j'', 1 \leq j, j', j'' \leq L; \\ & \vdots \\ & 0 \leq \sum_{j=1}^L k_j \leq \sum_{j=0}^L N_j = M \}. \end{aligned}$$

Furthermore, we consider the probabilities

$$P[\underline{k}, t] = P[\text{at time } t, \text{ the system is in state } \underline{k}],$$

$$P[\underline{k}] = \lim_{t \rightarrow \infty} P[\underline{k}, t], \quad \text{and the quantities}$$

$$\rho_j = \lambda_j / \mu_j, \quad \text{for } 1 \leq j \leq L.$$

Since the storage space M is finite, the steady-state probabilities $P[\underline{k}]$ always exist. It is possible, although tedious, to write the system of equilibrium equations which, together with the normalizing equation, uniquely determine the $P[\underline{k}]$'s. This system, however, is not particularly enlightening and we shall not use it to determine the steady-state probabilities but we rather use the local balance equations.

The local balance equations (see [1]) express the condition that in steady-state, the rate of entrance into a state of the

system by arrival of a message of any given type is equal to the rate of exit from the same state by processing a message of the same type. Formally

$$\lambda_j P[\underline{k}^{(j)}] = \mu_j P[\underline{k}]$$

for all j , for all $\underline{k} \in S$ such that $\underline{k}^{(j)} \in S$, where $\underline{k}^{(j)}$ is defined by

$$(\underline{k}^{(j)})_i = k_i, \quad \text{for all } i \neq j,$$

$$(\underline{k}^{(j)})_j = k_j - 1.$$

This system has the solution

$$P[\underline{k}] = \frac{1}{C} \prod_{j=1}^L \rho_j^{k_j},$$

where C is the normalizing constant defined by

$$(1) \quad C = \sum_{\underline{k} \in S} \prod_{j=1}^L \rho_j^{k_j}.$$

Let π and π_j respectively denote the steady-state probabilities that a message may not enter the system and that a message with destination j may not enter the system. One has

$$(2) \quad \pi = \left(\sum_{j=1}^L \lambda_j \right)^{-1} \sum_{j=1}^L \lambda_j \pi_j,$$

$$\pi_j = \sum_{\underline{k} \in S_j''} P[\underline{k}],$$

where $S_j'' = \{\underline{k} \in S \mid k_j \geq N_j \text{ and there exists } D \subset \{1, 2, \dots, N\}$

$$\text{such that } \sum_{i \in D} k_i = \sum_{i \in D} N_i + N_0\},$$

in other words, S_j'' is the set of states such that all the buffers allocated to the j th output process and all the common buffers are filled.

The throughput T is defined as the expected number of messages that enter the system per unit of time in steady-state, i.e.

$$T = \sum_{j=1}^L \lambda_j (1 - \pi_j). \quad T \text{ is clearly given by}$$

$$(3) \quad T = \left(\sum_{j=1}^L \lambda_j \right) (1 - \pi).$$

2. Special case: 2 Output Processes

There is in general no closed form expression for the normalizing constant C , nor for the loss probabilities π_j . If $L=2$ however, one can easily show that

$$(4) \quad C = \{ (1 - \rho_1)(1 - \rho_2)(\rho_1 - \rho_2) \}^{-1} \{ (\rho_1 - \rho_2) (1 - \rho_1)^{N_1 + N_0 + 1} \rho_2^{N_2 + N_0 + 1} + \rho_1^{N_1 + 1} \rho_2^{N_2 + 1} (\rho_2^{N_0} (1 - \rho_2) - \rho_1^{N_0} (1 - \rho_1)) \},$$

$$(5) \quad \pi_1 = C^{-1} \rho_1^{-1} (1 - \rho_2)^{-1} (1 - \rho_2^{N_2 + N_0 + 1}) - \rho_1^{-1} (1 - \rho_1),$$

$$(6) \quad \pi_2 = C^{-1} \rho_2^{-1} (1 - \rho_1)^{-1} (1 - \rho_1^{N_1 + N_0 + 1}) - \rho_2^{-1} (1 - \rho_2),$$

and

$$(7) \quad \pi = 1 - (\lambda_1 + \lambda_2)^{-1} (\mu_1 + \mu_2) + \{ C(\lambda_1 + \lambda_2) \}^{-1} \left\{ \mu_1 \frac{1 - \rho_2^{N_2 + N_0 + 1}}{1 - \rho_2} + \mu_2 \frac{1 - \rho_1^{N_1 + N_0 + 1}}{1 - \rho_1} \right\}.$$

Theorem

For a fixed value of N_2 , if N_1 is considered as a real number in $[0, M - N_2]$, then π as a function of N_1 has a unique minimum in $[0, M - N_2]$.

Moreover, if for some value $N_2 = N_2^0$, π is minimum for $N_1 = M - N_2$, the same is true for all $N_2 \geq N_2^0$.

Proof: The purely technical proof of this theorem is presented in Appendix A.

A completely analogous theorem holds for π as a function of N_2 for fixed N_1 since the expression for π is symmetrical in the indexes 1 and 2. These theorems and other technical properties make it possible to determine numerically the values (N_0, N_1, N_2) which minimize π without having to compare all the possible solutions.

We shall not present detailed numerical results for this special case but we mention that it appears that

$$(8) \quad \lim_{M \rightarrow \infty} (\max\{\tilde{N}_0, \tilde{N}_1, \tilde{N}_2\} M^{-1}) \geq 0.5 \quad \text{if } \rho_1 \neq \rho_2, \rho_1 \neq \rho_2^{-1},$$

where $\tilde{N}_i$ denotes the optimal value for N_i . Moreover,

$$\begin{aligned} \max\{\tilde{N}_0, \tilde{N}_1, \tilde{N}_2\} &= \tilde{N}_0 \quad \text{if } \rho_1 \rho_2 < 1, \\ &\tilde{N}_1 \quad \text{if } \rho_1 \rho_2 > 1, \rho_1 < \rho_2, \\ &\tilde{N}_2 \quad \text{if } \rho_1 \rho_2 > 1, \rho_2 < \rho_1. \end{aligned}$$

The rate of convergence depends on the ratio μ_1/μ_2 . If that ratio is close to 1, the inequality (8) is observed for values of M as small as 10. In other words, we have an asymptotic result which shows that for the optimal solution, one of the three areas occupies at least one half of the buffer space.

3. General Case: The Loss Probabilities π_j

For $L > 2$, no closed form expression has been determined for the normalizing constant C or the loss probabilities π_j . We rather present in this section the recurrence relations which may be used to compute those quantities efficiently.

Let us denote by $S[K; N_1, \dots, N_K; N_0]$, the set of admissible

states for a system with K output processes, N_j buffers reserved to the j th process and N_0 common buffers, and by $C[K; N_1, \dots, N_K; N_0]$, the normalizing constant for this system. Furthermore, $S'[K; N_1, \dots, N_K; N_0]$ is the set of admissible states such that some messages may not enter (some of the output processes are "blocked"), $S' = \{k \in S[K; \dots; N_0] \mid$

$$\exists D \subset \{1, 2, \dots, K\}, \sum_{j \in D} k_j = \sum_{j \in D} N_j + N_0\}.$$

Also, $H[K; N_1, \dots, N_K; N_0]$ denotes a sum similar to the expression (1) for C , with S and L replaced respectively by S' and K ;

$S''_j[K; N_1, \dots, N_K; N_0]$, $j \leq K$, the set of admissible states such that no message to the j th output process may enter;

$G_j[K; N_1, \dots, N_K; N_0]$, $j \leq K$, a sum similar to the expression (1) for C , with S and L replaced by S'' and K respectively.

We obviously have that

$$\pi_j = \frac{G_j[L; N_1, \dots, N_L; N_0]}{C[L; N_1, \dots, N_L; N_0]}, \text{ for } 0 \leq j \leq L.$$

Lemma 1:

$$C[1; N_1; N_0] = (1 - \rho_1)^{-1} (1 - \rho_1^{N_1 + N_0 + 1}).$$

For $K \geq 1$,

$$\begin{aligned} & C[K+1; N_1, N_2, \dots, N_K, W; N_0] \\ &= (1 - \rho_{K+1})^{-1} (1 - \rho_{K+1}^W) C[K; N_1, N_2, \dots, N_K; N_0] \\ &+ \rho_{K+1}^W \sum_{i=0}^{N_0} \rho_{K+1}^i C[K; N_1, N_2, \dots, N_K; N_0 - i]. \end{aligned}$$

Proof:

The first relation is obvious.

To prove the second relation formally, we observe that

$$\begin{aligned}
 & S[K+1; N_1, N_2, \dots, N_K, W; N_0] \\
 &= \left[\bigcup_{i=0}^{W-1} \{(\underline{k}, i) \mid \underline{k} \in S[K; N_1, \dots, N_K; N_0]\} \right] \\
 &\quad \cup \left[\bigcup_{i=0}^{N_0} \{(\underline{k}, W+i) \mid \underline{k} \in S[K; N_1, \dots, N_K; N_0-i]\} \right]
 \end{aligned}$$

and therefore

$$\begin{aligned}
 & C[K+1; N_1, N_2, \dots, N_K, W; N_0] \\
 &= \sum_{i=0}^{W-1} \rho_{K+1}^i \sum_{\underline{k} \in S[K; \dots; N_0]} \prod_{j=1}^K \rho_j^{k_j} \\
 &\quad + \sum_{i=0}^{N_0} \rho_{K+1}^{W+i} \sum_{\underline{k} \in S[K; \dots; N_0-i]} \prod_{j=1}^K \rho_j^{k_j}.
 \end{aligned}$$

Informally stated, if $k_{K+1} < W$, the other output processes are allowed to use all the N_0 common buffers, if $W \leq k_{K+1} \leq W+N_0$, the other processes are allowed to use $(N_0 - k_{K+1} + W)$ common buffers.

Lemma 2:

$$H[1; N_1; N_0] = \rho_1^{N_1 + N_0},$$

for $K \geq 1$,

$$\begin{aligned}
 & H[K+1; N_1, \dots, N_K, W; N_0] = (1 - \rho_{K+1})^{-1} (1 - \rho_{K+1}^W) H[K; N_1, \dots, N_K; N_0] \\
 &\quad + \rho_{K+1}^W \sum_{i=0}^{N_0} \rho_{K+1}^i H[K; N_1, \dots, N_K; N_0-i] \\
 &\quad + \rho_{K+1}^{W+N_0} \prod_{j=1}^K (1 - \rho_j)^{-1} (1 - \rho_j^{N_j}).
 \end{aligned}$$

Proof:

The first relation is obvious. One proves the second one informally as follows. The first term on the right-hand side corresponds to the $(K+1)$ st output process using less than W buffers, all the N_0 common buffers being used, thereby one of the other processes being blocked. The second term corresponds to the $(K+1)$ st output process using $(W+i)$ buffers, $0 \leq i \leq N_0$, all the (N_0-i) remaining common buffers being used and at least one of the other processes being blocked. The last term corresponds to the $(K+1)$ st process using all the N_0 common buffers, none of the other processes being blocked.

Lemma 3:

$$G_1[1; N_1; N_0] = \rho_1^{N_1+N_0},$$

for $j \geq 2$,

$$\begin{aligned} G_j[j; N_1, \dots, N_{j-1}, W; N_0] \\ = \rho_j^{W+N_0} C[j-1; N_1, \dots, N_{j-1}; 0] \\ + \rho_j^W \sum_{i=0}^{N_0-1} \rho_j^i H[j-1; N_1, \dots, N_{j-1}; N_0-i], \end{aligned}$$

for $K \geq j$,

$$\begin{aligned} G_j[K+1; N_1, \dots, N_K, W; N_0] \\ = (1-\rho_{K+1})^{-1} (1-\rho_{K+1}^W) G_j[K; N_1, \dots, N_K; N_0] \\ + \rho_{K+1}^W \sum_{i=0}^{N_0} \rho_{K+1}^i G_j[K; N_1, \dots, N_K; N_0-i]. \end{aligned}$$

Proof:

The first relation is obvious. The second can be informally proved as follows: either the j th process uses all of its

allocated buffers and all the common buffers, the other processes may use any number of their allocated buffers, or the j th process uses all of its allocated buffers but not all of the common buffers and the other processes use the remaining common buffers.

The third relation is proved as follows: either the $(K+1)$ st process does not use all of its allocated buffers, the other processes use N_0 common buffers and the j th process is blocked or the $(K+1)$ st process uses at least all of its allocated buffers, the other processes use the remaining common buffers and the j th process is blocked.

Remark: To determine $C[K+1; \dots; N_0]$, using Lemma 1, one needs to determine $C[K; \dots; 0]$, $C[K; \dots; 1]$, ..., $C[K; \dots; N_0]$ and similarly for H and G .

Using those three Lemmas, it is possible to design an algorithm for the determination of the probabilities π_j and therefore the probability π and the throughput T . The algorithm is presented in Appendix B. We did not try to identify the largest problem which the resulting program can handle, but a system with $M=100$, $L=20$, $N_0=20$, $N_1=N_2=\dots=N_L=4$ was solved in about 0.5 seconds of CPU time.

4. Balanced Optimum and Square Root Policies

Consider the problem of determining a set of parameters $(N_0, N_1, \dots, N_L)$ ($\sum_{j=0}^L N_j = M$) which maximize the throughput T or, equivalently, minimize the loss probability π . Without further analysis of π , this can be done only by evaluating π

for each of the $\binom{M+L}{M}$ possible values for $(N_0, N_1, \dots, N_L)$. Such small systems as $M=20$, $L=5$ begin to tax the computer, even if the algorithm avoids repetition of lengthy computations. Therefore, we restrict our attention to suboptimal policies.

A. Optimal balanced $(L+1)$ -sharing policy

An optimal balanced policy is represented by (N_0, N) and is the set of parameters $(N_0, N_1, \dots, N_L)$ which minimize the loss probability π subject to the following constraints:

$$N_1 = N_2 = \dots = N_L = N,$$

$$N_0 + LN = M.$$

The determination of (N_0, N) requires the evaluation of π for $[M/L]+1$ systems.

B. Square-root $(L+1)$ -sharing policy

The optimal balanced policy depends on the system parameters and requires readjustment as traffic characteristics change. As this may prove difficult to do in practice, one would set (N_0, N) to a fixed value which yields a reasonable loss probability π over a large range of parameter values. We find such a pair (N_0, N) by using an argument similar to that given by Irland [6]. First we observe that losses are usually insignificant for small values of the ρ_j 's. Therefore, we shall analyze the optimal balanced policy in the special case where $\rho_1 = \rho_2 = \dots = \rho_L = 1$. For $L=2$, one can show from Equations (2) and (4) to (6), by using l'Hospital's rule, that

$$\pi = \frac{2(M-N+1)}{(M+1)(M+2)-2N(N+1)}.$$

If N can take any real value in $[0, ML^{-1}]$, π is minimum for

$$N = M+1 - \sqrt{(M+1)(M+2)/2},$$

which is approximately equal to

$$N \approx \frac{M}{2+\sqrt{2}},$$

so that $N_0 \approx M - 2N = \frac{M}{1+\sqrt{2}}$.

For general $L \geq 2$, we define the square-root $(L+1)$ -sharing policy as follows:

$$N = N_1 = \dots = N_L = \left\lfloor \frac{M}{L+\sqrt{L}} \right\rfloor \text{ or } \left\lceil \frac{M}{L+\sqrt{L}} \right\rceil,$$

whichever is best in the case $\rho_j = 1$, $1 \leq j \leq L$;

$$N_0 = M - LN \geq 0.$$

In the next section, we examine the overall quality of this policy. It seems to be close to the optimal policy if $\rho_j = 1$, $1 \leq j \leq L$, although we do not have a formal proof of this assertion. Let us mention that we have numerically determined the optimal solution for the following systems:

$$L = 3, 4, \dots, 20;$$

$$M = 20, 30, \dots, 100;$$

$$\lambda_j = \mu_j, \quad 1 \leq j \leq L.$$

In three fourth of the cases, the square-root policy is optimal and in all the other cases, the optimal value for N is

$$N = \left\lfloor \frac{M}{L+\sqrt{L}} \right\rfloor - 1,$$

and the value of the loss probability for the square-root policy is very close to the minimum.

Remark: Irland [6] determines a square-root restricted sharing policy which is close to the optimal for $\rho_j=1$, $1 \leq j \leq L$:

$$(9) \quad K \approx \frac{M}{\sqrt{L}}.$$

For the square-root $(L+1)$ -sharing policy, the maximum number of buffers that any output process may use is

$$N + N_0 \approx \frac{M}{L+\sqrt{L}} + \frac{M}{1+\sqrt{L}} = \frac{M}{\sqrt{L}}$$

and is equal (see (9)) to the maximum number of buffers under the square-root restricted sharing policy. Of course, under restricted sharing, two or more output processes may use simultaneously $M/\sqrt{L}$ buffers each, while this is not true for $(L+1)$ -sharing.

5. Numerical Comparisons

We now present some numerical results (see also Figures 1 to 14). Our main objective is to compare the different policies we have defined. This leaves several interesting questions unanswered, such as those related to the asymptotic behavior for M and $L \rightarrow \infty$.

As six policies are involved, the figures tend to get confused. To alleviate this problem somewhat, we adopt the following conventions.

Conventions:

- i. The loss probabilities are presented on a logarithmic scale.
- ii. Results for the unrestricted sharing policy are represented by a "0",

-results for the no sharing policy are represented by a "+",
 -results for all the other policies are represented by a continuous line, marked by a

"1" for the optimal balanced $(L+1)$ -sharing policy,

"2" for the square root $(L+1)$ -sharing policy,

"3" for the optimal restricted sharing policy,

"4" for the square root restricted sharing policy.

Moreover, we assume in all of the examples that the output processes have the same speed, chosen as the unit: $\mu_j=1$, $1 \leq j \leq L$.
 Hence, $\rho_j=\lambda_j$, $1 \leq j \leq L$.

A. Balanced load (figures 1 and 2)

Consider first the case where all the input rates are equal: $\lambda_1=\lambda_2=\dots=\lambda_L$. Since the number of accessible buffers is the same for each output link, the loss probabilities are equal: $\pi=\pi_j$, $1 \leq j \leq L$. Figures 1 and 2 present the throughput and the loss probability for $L=3$, $M=10$ and for different values for the common traffic coefficient ρ_1 .

For small values of ρ_1 ($\rho_1 < 1$), the throughput for the no sharing policy is substantially smaller than for the other policies. This property seems rather general since we have systematically observed it (see figures 3, 7, 11) whenever $\rho_j < 1$, $1 \leq j \leq L$.

For large values of ρ_1 ($\rho_1 > 1$), the unrestricted sharing policy offers the smallest throughput, followed by the square root restricted sharing policy. This suggests that the unrestricted sharing policy is the most sensitive to an increase of congestion affecting uniformly all the output processes; and that the square root restricted sharing is more sensitive to such a

phenomenon than the other controlled sharing policies.

B. Unbalanced load (figures 3 to 10)

For the next example, $L=3$ and $M=10$ (figures 3 to 6) and 20 (figures 7 to 10). Only ρ_1 varies, the other traffic coefficients remain constant ($\rho_2=.5$, $\rho_3=.6$).

Notice first that for $\rho_1 > 1$, the throughput for the unrestricted sharing policy decreases as ρ_1 increases. This property seems to be a characteristic of the situations where the ρ_j 's are not all equal, a justification has been proposed by Irland [6].

For $\rho_1 < 1$, the global loss probabilities π for the optimal balanced $(L+1)$ -sharing policy and the optimal restricted sharing policy are nearly equal (see figures 4 and 8); a similar property holds for the corresponding square-root policies. Differences appear for $\rho_1 > 1$ (see figures 3 and 7); the restricted sharing policies have a larger throughput than the $(L+1)$ -sharing policies. However, those differences get smaller when M increases.

In this case, the loss probabilities π_j are not equal; figures 5, 6, 9 and 10 present π_1 and π_2 . Observe that for all the controlled sharing policies, π_2 varies on the interval $0 \leq \rho_1 \leq 1$ much less than π_1 , especially so for the square root policies; again, this property is general. It indicates that if the traffic is stable for some output links but not for others, both the restricted sharing and the $(L+1)$ -sharing policies reduce the effect of such traffic variations on the stable output processes. Moreover, one observes that for those policies, $\pi_j > \pi_{j'}$, if $\rho_j > \rho_{j'}$.

C. Highly unbalanced load (figures 11 to 14)

For our last example, we consider a situation where most of the output processes have a high traffic coefficient: $L=4$, $M=20$, $\rho_2=.7$, $\rho_3=.8$, $\rho_4=.9$, ρ_1 varies.

It appears first (see figure 11) that for $\rho_1 > 1$, the square root restricted sharing policy does not perform as well as in the preceding example.

Most interesting is the observation one can make from figures 12 to 14 ($\rho_1 \leq 1$):

- the global loss probability, π , takes very similar values for all the controlled sharing policies,
- for small values of ρ_1 , π_1 is noticeably smaller under $(L+1)$ -sharing than under restricted sharing,
- there is less difference among those policies for the other loss probabilities.

In other words, if most of the output processes are heavily loaded and a few are lightly loaded, the latter will receive more favorable treatment under $(L+1)$ -sharing than under restricted sharing with no large increase of loss probability for the other processes nor the system as a whole.

Again, as noted earlier, those differences decrease for larger M .

6. Conclusions and Comments

The no sharing and unrestricted sharing policies prove to be completely inadequate, the former because it performs poorly when all ρ_j 's are smaller than 1 (or slightly larger), which seems a frequently prevailing condition, the latter because it presents no robustness against an increase of traffic for some

or all of the output processes.

If the objective is to minimize the global loss probability π , the optimal restricted sharing policy proves to be best by a small margin in the majority of cases: we have found very few (usually unrealistic) situations where the optimal balanced $(L+1)$ -sharing policy is better.

If the objective is to have a small global loss probability π and a small loss probability π_j for lightly loaded output processes, then there is no uniformly best policy. In cases similar to example B, restricted sharing gives best results, for situations similar to example C, it is $(L+1)$ -sharing. The square root policies have the advantage, as we mentioned earlier, of being independent of the parameters. They perform well in general and there does not seem to be any general answer to the question of which of the two square root policies minimizes the global loss probability π . However the square root $(L+1)$ -sharing policy may prove to be preferable if one takes into consideration the risk of coalitions (defined informally below): coalitions are impossible under square root $(L+1)$ -sharing if $M > L + \sqrt{L}$, but may occur under square root restricted sharing if $L \geq 3$. A coalition occurs when the output processes have a behavior which allows a subset of them to always use all of the M buffers, preventing the other processes from ever getting any message. This is reflected to some degree by the values of π_1 for small values of ρ_1 in example C (figure 13).

We now comment on some of the assumptions we have made in defining our model.

Firstly, we have assumed that arrivals form a Poisson process and services are exponential. To the best of our knowledge, only Fisher [5] considers a general service time distribution, $L=2$. Fisher proposes an approximation for the steady-state probability distribution under unrestricted sharing and makes comparisons with the no sharing policy in the exponential case. Extending the analysis of our model to general service time and/or interarrival time distributions seems to be a major task; before engaging in it, some measurement of the robustness (or lack of robustness) of the results for the Markovian case might be appropriate.

Secondly, we have considered the system in stationary state. This may not be appropriate since the conditions are likely to change over time for the applications described in the introduction. However, the results we have obtained suggest that the square root policies are generally close to the optimal and should be good ones provided that abrupt changes do not occur too frequently, especially if one takes into consideration the costs of adjusting the allocation policy to a changing environment.

Two of our assumptions have a crucial influence on our analysis. We have assumed that a message that cannot join the system is lost. This may be legitimate for a packet-switch communications network: such a message would either be resubmitted later (and appear to be a new one) or transmitted through another path in the network (and be effectively lost for the system under consideration). For producer-consumer parallel processes, this

assumption is at best an approximation. A producer that cannot find a buffer to enter a message will wait until it gets a free buffer. This should affect the steady-state probabilities but not necessarily our qualitative comparisons of the different policies.

We have also assumed that all random variables are independent. This affects our qualitative conclusions. For instance, this assumption reduces the risk of coalition. It is possible to define arrival processes which are not independent, such that the square root restricted sharing policy and even the optimal restricted sharing policy (based on the Markovian assumption) perform disastrously: coalitions are certain to occur and the output process with the smallest traffic coefficient never gets any messages.

Furthermore, situations, not covered by our numerical analysis, are worthy of future investigation. For instance, would our qualitative conclusions hold if the service rates μ_j 's were not all equal? More importantly perhaps, one should try to get asymptotic results for large values of M and L ; intuitively, it seems that for fixed L and increasing M , the differences between policies would reduce. This is supported by our numerical results.

Finally, one may consider that the messages have not all the same importance and, for instance, assign costs for loss of a message, which are different according to the destination. The function to minimize in this case is the expected cost per unit of time:

$$\Gamma = \sum_{j=1}^L \gamma_j \lambda_j \pi_j$$

where γ_j is the cost for loss of a message with destination j .

In [9] we analyze the no sharing policy in a small system: $L=2$, exponential services. For general L and $(L+1)$ -sharing policy, Γ can be determined as easily as π (Section 3). If costs are different, one cannot restrict the search for an optimum among the balanced policies; we have already indicated in Section 4 that an exhaustive comparison of all the possible values for $(N_0, N_1, \dots, N_L)$ is prohibitively costly except for very small systems. Therefore, Γ would have to be analyzed further in order to determine an efficient algorithm.

Appendix A

1. Let $L=2$. Consider N_2 fixed and let N_1 take any real value in $[0, M-N_2]$. π as a function of N_1 has a unique minimum in $[0, M-N_2]$.

Proof:

As $N_2 + N_1 + N_0 = M$, the expression (7) for π can be written as

$$\pi = \gamma + \delta g_1(N_1; N_2, M) / g_2(N_1; N_2, M)$$

where

$$\gamma = 1 - (\lambda_1 + \lambda_2)^{-1} (\mu_1 + \mu_2) \text{ is independent of } N_1,$$

$$\delta = (\lambda_1 + \lambda_2)^{-1} \text{ is positive and independent of } N_1,$$

$$g_1 = \mu_1 (1 - \rho_1) (1 - \rho_2)^{M-N_1+1} + \mu_2 (1 - \rho_2) (1 - \rho_1)^{M-N_2+1},$$

$$g_2 = (\rho_1 - \rho_2)^{-1} \{ (\rho_1 - \rho_2) (1 - \rho_1)^{M-N_2+1} - \rho_2^{M-N_1+1} \\ + \rho_1^{N_1+1} \rho_2^{M-N_1+1} (1 - \rho_2) - \rho_1^{M-N_2+1} \rho_2^{N_2+1} (1 - \rho_1) \}.$$

Replacing N_1 by x , it is easy to check that

$$\pi'(x) = (h_1(x))^2 h(x)$$

where $h_1(x)$ is a function of no interest to us and $h(x)$ is given by

$$(A.1) \quad h(x) = (\rho_1 - \rho_2)^{-1} \{ -\mu_1 (1 - \rho_1) (\rho_1 - \rho_2) \rho_1^{M-N_2+1} \log \rho_2 \\ - \mu_1 (1 - \rho_1)^2 \rho_2^{N_2+1} \rho_1^{M-N_2+1} \log \rho_2 \\ - \mu_2 (1 - \rho_2) (\rho_1 - \rho_2) (1 - \rho_1)^{M-N_2+1} \log \rho_2 \\ + \mu_1 (1 - \rho_1) (1 - \rho_2) \rho_1^{x+1} \rho_2^{M-x+1} \log \rho_2 \\ - [\mu_1 (1 - \rho_1) (1 - \rho_2)^{M-x+1} + \mu_2 (1 - \rho_2) (1 - \rho_1)^{M-N_2+1}] \\ (1 - \rho_2) \rho_1^{x+1} (\log \rho_1 - \log \rho_2) \}.$$

The derivative of $h(x)$ is

$$h'(x) = -\rho_1^{x+1} (\log \rho_1 - \log \rho_2) (\rho_1 - \rho_2)^{-1} \times \\ \{ \mu_1 (1 - \rho_2) (1 - \rho_2^{M-x+1}) (1 - \rho_1) \log \rho_1 \\ + \mu_2 (1 - \rho_2)^2 (1 - \rho_1^{M-N_2+1}) \log \rho_1 \}$$

and is positive since $(1 - \rho^\alpha) \log \rho \leq 0$, $\alpha > 0$; hence $h(x)$ is increasing and it is trivial to conclude that

π is minimum for $N_1 = 0$ if $h(0) \geq 0$,

π has a unique minimum in $(0, M - N_2)$ if $h(0) < 0 < h(M - N_2)$,

π is minimum for $N_1 = M - N_2$ if $h(M - N_2) \leq 0$.

2. If $h(M - N_2^0) \leq 0$, then $h(M - N_2) \leq 0$ for all $N_2 \geq N_2^0$.

Proof:

From (A.1), $h(M - N_2) \leq 0$ if and only if

$$(A.2) \quad \mu_1 (1 - \rho_1) (1 - \rho_2^{N_2+1}) (\rho_1 - \rho_2)^{-1} \rho_1^{M-N_2+1} [(1 - \rho_1) \log \rho_2 - (1 - \rho_2) \log \rho_1] \\ - \mu_2 (1 - \rho_2) (1 - \rho_1^{M-N_2+1}) (\rho_1 - \rho_2)^{-1} [(\rho_1 - \rho_2) \log \rho_2 + \\ (1 - \rho_2) \rho_1^{M-N_2+1} (\log \rho_1 - \log \rho_2)] \leq 0.$$

As $(1 - \rho)^{-1} \log \rho$ is an increasing function of ρ , it is easy to verify that the coefficient of μ_1 in (A.2) is negative.

The coefficient of $(-\mu_2)$ is negative; it can be written as

$(1 - \rho_1)^{-1} (1 - \rho_1^{M-N_2+1}) g(N_2)$ where

$$g(x) = (1 - \rho_1) (1 - \rho_2) (\rho_1 - \rho_2)^{-1} [(\rho_1 - \rho_2) \log \rho_2 + \\ (1 - \rho_2) \rho_1^{M-x+1} (\log \rho_1 - \log \rho_2)].$$

As $g'(x) = -(1-\rho_1)(1-\rho_2)^2(\rho_1-\rho_2)^{-1}\rho_1^{M-x+1}\log\rho_1(\log\rho_1-\log\rho_2)$ is positive and $g(M) = (1-\rho_1)^2(1-\rho_2)^2(\rho_1-\rho_2)^{-1}[(1-\rho_1)^{-1}\rho_1\log\rho_1 - (1-\rho_2)^{-1}\rho_2\log\rho_2]$ is negative since $(1-\rho)^{-1}\rho\log\rho$ is a decreasing function of ρ , $g(x)$ is negative for all $x \leq M$.

Hence, $h(M-N_2) \leq 0$ if and only if $\mu_1/\mu_2 \geq b(N_2)$, where

$$b(x) = \{(1-\rho_1)(1-\rho_2)^{x+1}\rho_1^{M-x+1}[(1-\rho_1)\log\rho_2 - (1-\rho_2)\log\rho_1]\}^{-1} \times \\ \{(1-\rho_2)(1-\rho_1)^{M-x+1}[(\rho_1-\rho_2)\log\rho_2 + (1-\rho_2)\rho_1^{M-x+1}(\log\rho_1-\log\rho_2)]\} \geq 0$$

After tedious algebraic manipulations, one can write $b'(x)$ as

$$b'(x) = \rho_1^{M-x+1} \{(1-\rho_1)^4(1-\rho_2)^2(\rho_1-\rho_2)^{-1}[(1-\rho_1)^{-1}\log\rho_1 - (1-\rho_2)^{-1}\log\rho_2]\}^{-1} \times \{-(1-\rho_2)(1-\rho_2)^{x+1}(1-\rho_1)\log\rho_1 g(2x-M-1) \\ -(1-\rho_1)(1-\rho_1)^{M-x+1}\rho_2^{x+1}(1-\rho_2)\log\rho_2 g(x)\}$$

which is clearly negative for all $x \leq M$. Therefore, $b(x)$ is decreasing and $b(N_2^0) \leq \mu_1/\mu_2$ implies $b(N_2) \leq \mu_1/\mu_2$ for all $N_2 \geq N_2^0$, which completes the proof.

Appendix B

Denote by $\underline{x}$ an infinite sequence $(x_0, x_1, x_2, \dots)$ and let the operator $*$ denote the convolution:

$$(\underline{x} * \underline{y})_j = \sum_{i=0}^j x_i y_{j-i} \quad 0 \leq j.$$

The Lemmas 1 to 3 (Section 3) can be written as follows, using N_{K+1} instead of W . To simplify notations, we use $C_K(N_0)$ instead of $C[K; N_1, N_2, \dots, N_K; N_0]$ and we let $\underline{C}_K$ represent the sequence $(C_K(0), C_K(1), \dots)$, similarly for H and G . Moreover,

$$\underline{\rho}_j = (1, \rho_j, \rho_j^2, \dots), \quad 1 \leq j \leq L$$

and we define the sequence $\underline{v}_j$ as follows:

$$\underline{v}_j(n) = \sum_{i=0}^n \rho_j^i \quad 1 \leq j \leq L, \quad 0 \leq n.$$

Lemma 1'

$$C_1(N_0) = \underline{v}_1(N_1 + N_0) \quad 0 \leq N_0,$$

$$\underline{C}_K = \underline{v}_K(N_{K-1}) \underline{C}_{K-1} + \rho_K^{N_K} (\underline{\rho}_K * \underline{C}_{K-1}) \quad 2 \leq K \leq L.$$

Lemma 2'

$$H_1(N_0) = \rho_1^{N_1 + N_0} \quad 0 \leq N_0,$$

$$\begin{aligned} \underline{H}_K &= \underline{v}_K(N_{K-1}) \underline{H}_{K-1} + \rho_K^{N_K} (\underline{\rho}_K * \underline{H}_{K-1}) \\ &\quad + \rho_K^{N_K} \underline{\rho}_K \prod_{j=1}^{K-1} \underline{v}_j(N_j - 1), \quad 2 \leq K \leq L. \end{aligned}$$

Lemma 3' .

$$\underline{G}_{1,1} = \underline{H}_1,$$

$$\underline{G}_{K,K} = \rho_K^{N_K} (\underline{C}_{K-1}(0) - \underline{H}_{K-1}(0)) \underline{\rho}_K + \rho_K^{N_K} (\underline{\rho}_K * \underline{H}_{K-1}), \quad 2 \leq K \leq L,$$

$$\underline{G}_{j,K} = \underline{\nu}_K^{(N_K-1)} \underline{G}_{j,K-1} + \rho_K^{N_K} (\underline{\rho}_K * \underline{G}_{j,K-1}), \quad 1 \leq j < K \leq L.$$

To compute the first N components of $\underline{y} = \underline{x} * \underline{\rho}$ where $\underline{\rho} = (1, \rho, \rho^2, \dots)$, we use the following algorithm (Buzen [2]):

```

CONVOL (y,x, $\rho$ ,N);
  real array y,x; real  $\rho$ ; integer N;
  begin integer k;
    y(0):=x(0);
    for k:=1 step 1 until N do
      y(k):= $\rho$ y(k-1)+x(k);
  end;

```

This algorithm requires N multiplications and N additions.

Therefore, to determine $\underline{C}$ and $\underline{G}_{j,L}$, $1 \leq j \leq L$, one has to perform

$N_0(L-1)(L+7)+M+2L$ additions

and $N_0(L-1)(3L/2+5)+M+L$ multiplications.

(Details are indicated in Table 1). Moreover, memory requirements are not very large if one discards intermediate results when they are no longer needed.

	additions	multiplications
C_1 ($\underline{p}_1$ and $\underline{v}_1$)	$N_1 + N_0 + 1$	$N_1 + N_0$
C_K ($2 \leq K \leq L$) ($\underline{p}_K$ and $\underline{v}_K$) convolution total	$N_K + N_0 + 1$ N_0 $N_K + 3N_0 + 1$	$N_K + N_0$ N_0 $N_K + 4N_0$
H_K ($2 \leq K \leq L$) convolution last term total	N_0 $3N_0$	N_0 N_0 $4N_0$
$G_{K,K}$ ($2 \leq K \leq L$)	$N_0 + 1$	$N_0 + 1$
$G_{j,K}$ ($2 \leq K \leq L$) convolution total total all j	N_0 $2N_0$ $2(K-1)N_0$	N_0 $3N_0$ $3(K-1)N_0$
H_1 and $G_{1,1}$	0	0

Table 1

References

- [1] F. Baskett, K. M. Chandy, R. R. Muntz and F. G. Palacios (1975)
Open, Closed and Mixed Networks of Queues with Different Classes of Customers
Journal of the ACM, 22, 248-260.
- [2] J. P. Buzen (1973)
Computational Algorithms for Closed Queueing Networks with Exponential Servers
Communications of the ACM, 16, 527-531.
- [3] R. Devillers and G. Louchard (1977)
Improvement of Parallelism in a Finite Buffer Sharing Policy
The Computer Journal, 19, 238-245.
- [4] E. W. Dijkstra (1972)
Information Streams Sharing a Finite Buffer
Information Processing Letters, 1, 179-180.
- [5] M. J. Fisher (1977)
Two Queueing Systems Sharing the Same Finite Waiting Room
Defense Communications Engineering Center, Reston, Virginia
Presented at the ORSA/TIMS Meeting, Atlanta
- [6] M. I. Irland (1978)
Buffer Management in a Packet-Switch
IEEE Transactions on Communications, COM-26.
- [7] M. I. Irland (1976)
Computational Algorithms for a System of Exponential Servers Sharing a Common Finite Storage
Report E-46, Computer Communications Network Group, University of Waterloo
- [8] S. R. Kimbleton and G. M. Schneider (1975)
Computer Communication Networks: Approaches, Objectives and Performance Considerations
ACM Computing Surveys, 7, 129-173.
- [9] G. Latouche (1977)
Optimal Partitioning of a Finite Buffer Between Two Pairs of Producer-Consumer
Proceedings of ICS 77 (ACM), Liège, Belgium
- [10] L. Pouzin (1973)
Presentation and Major Design Aspects of the CYCLADES Computer Network
Proceedings of the IEEE-ACM Third Data Communications Symposium, St. Petersburg, Florida

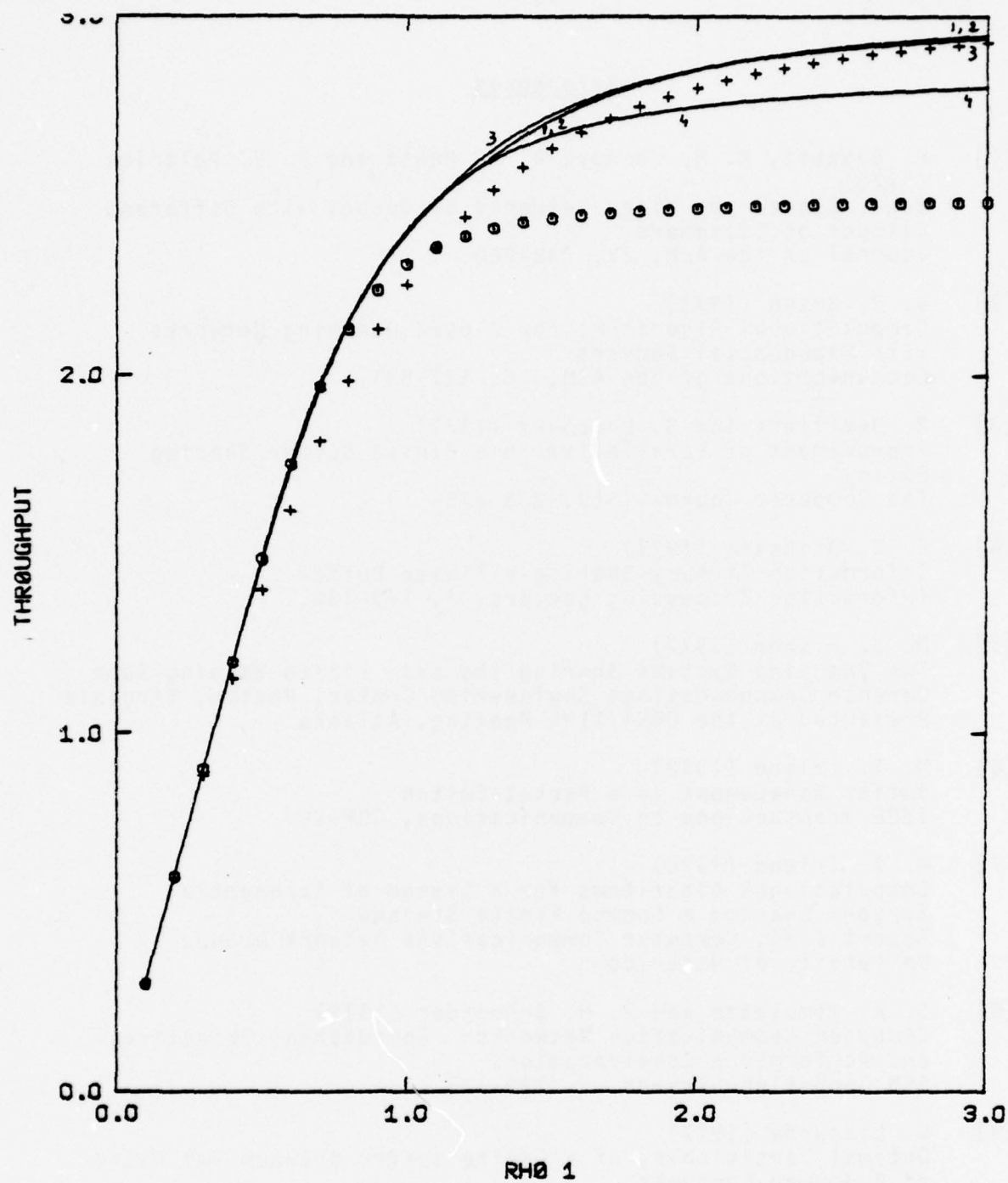


Figure 1 Throughput; balanced queue, $L=3$, $M=10$

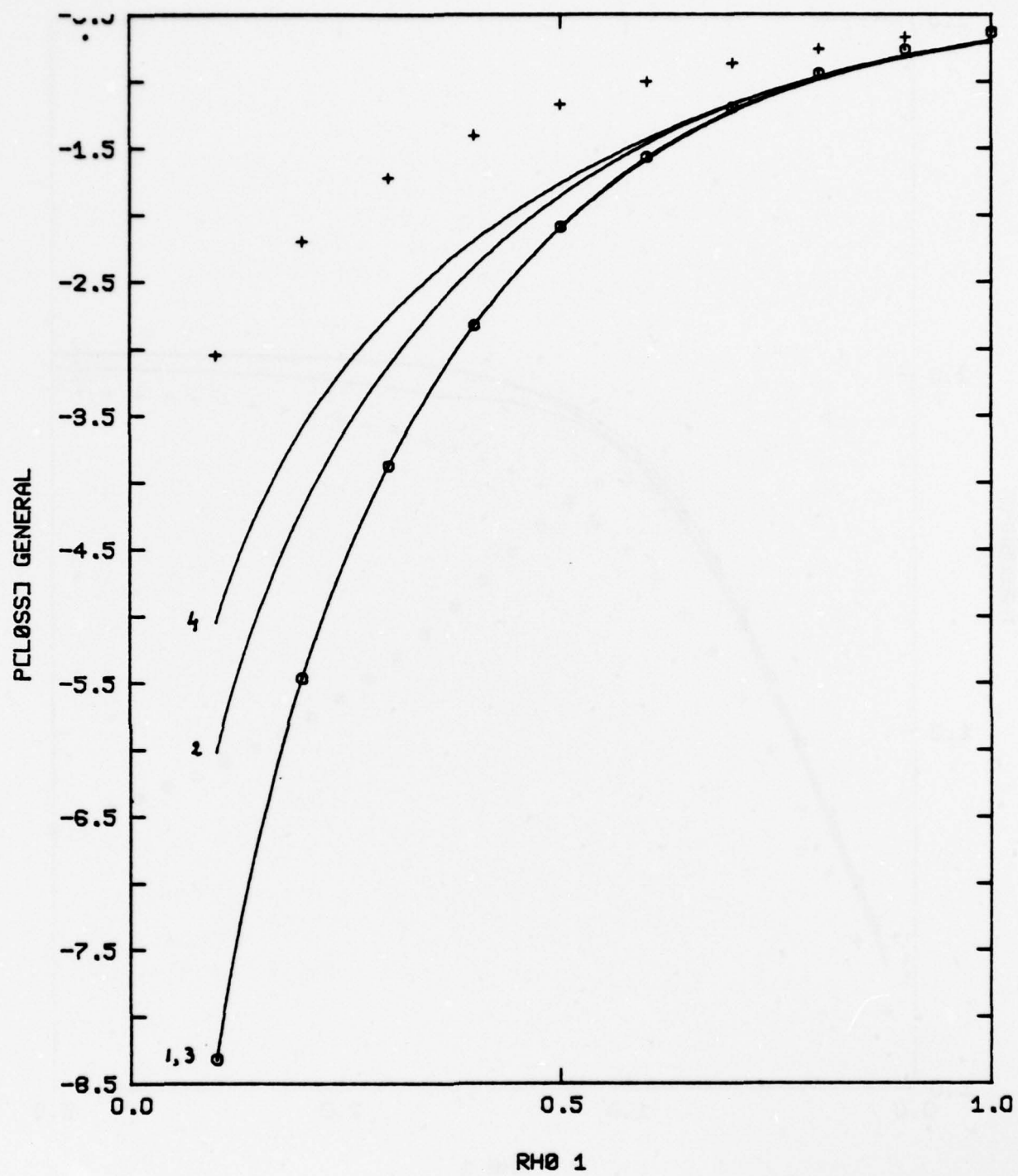


Figure 2 Loss probability π ; balanced queue, $L=3$, $M=10$

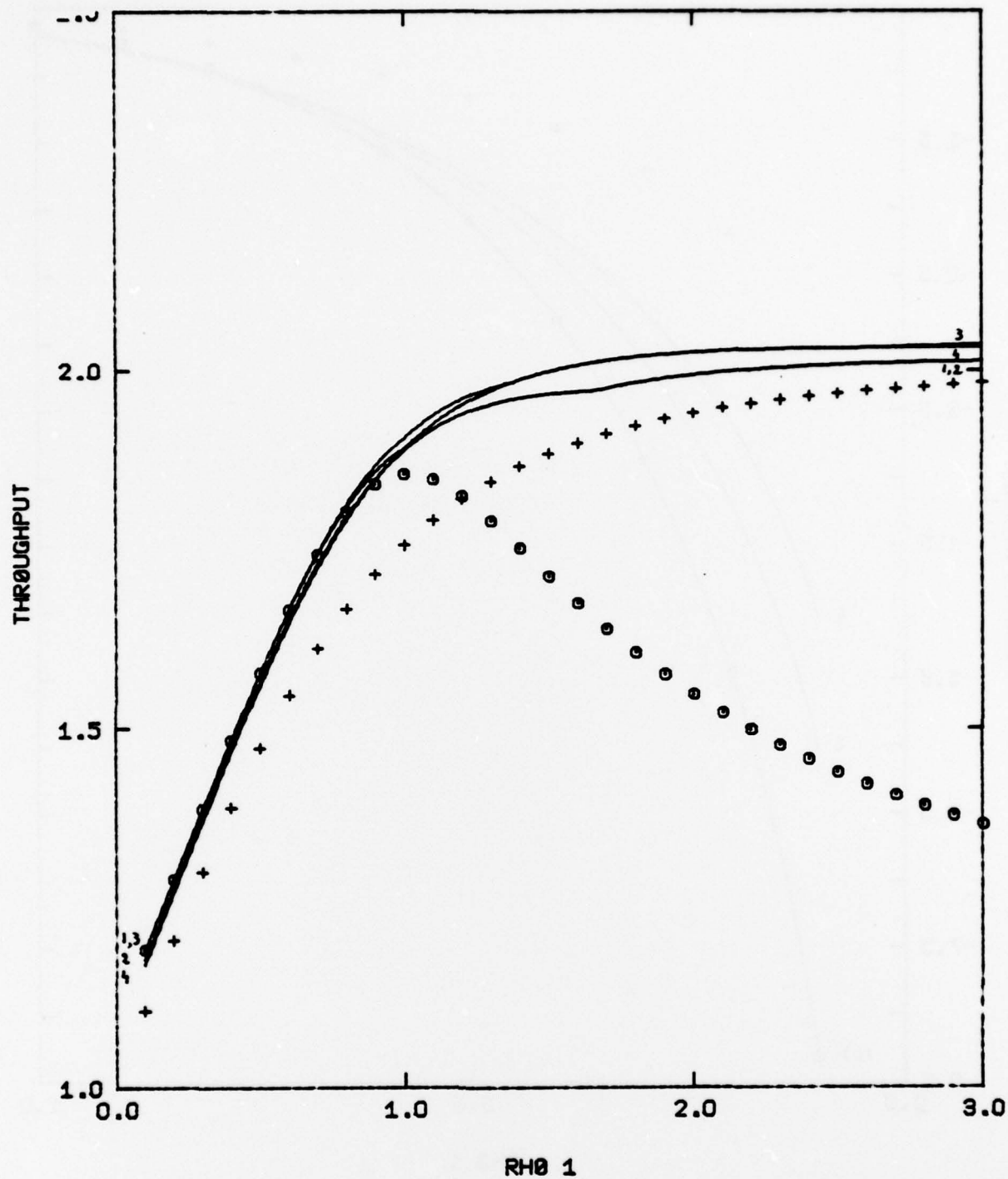


Figure 3 Throughput; unbalanced queue, $L=3$, $M=10$, $\rho_2=0.5$, $\rho_3=0.6$

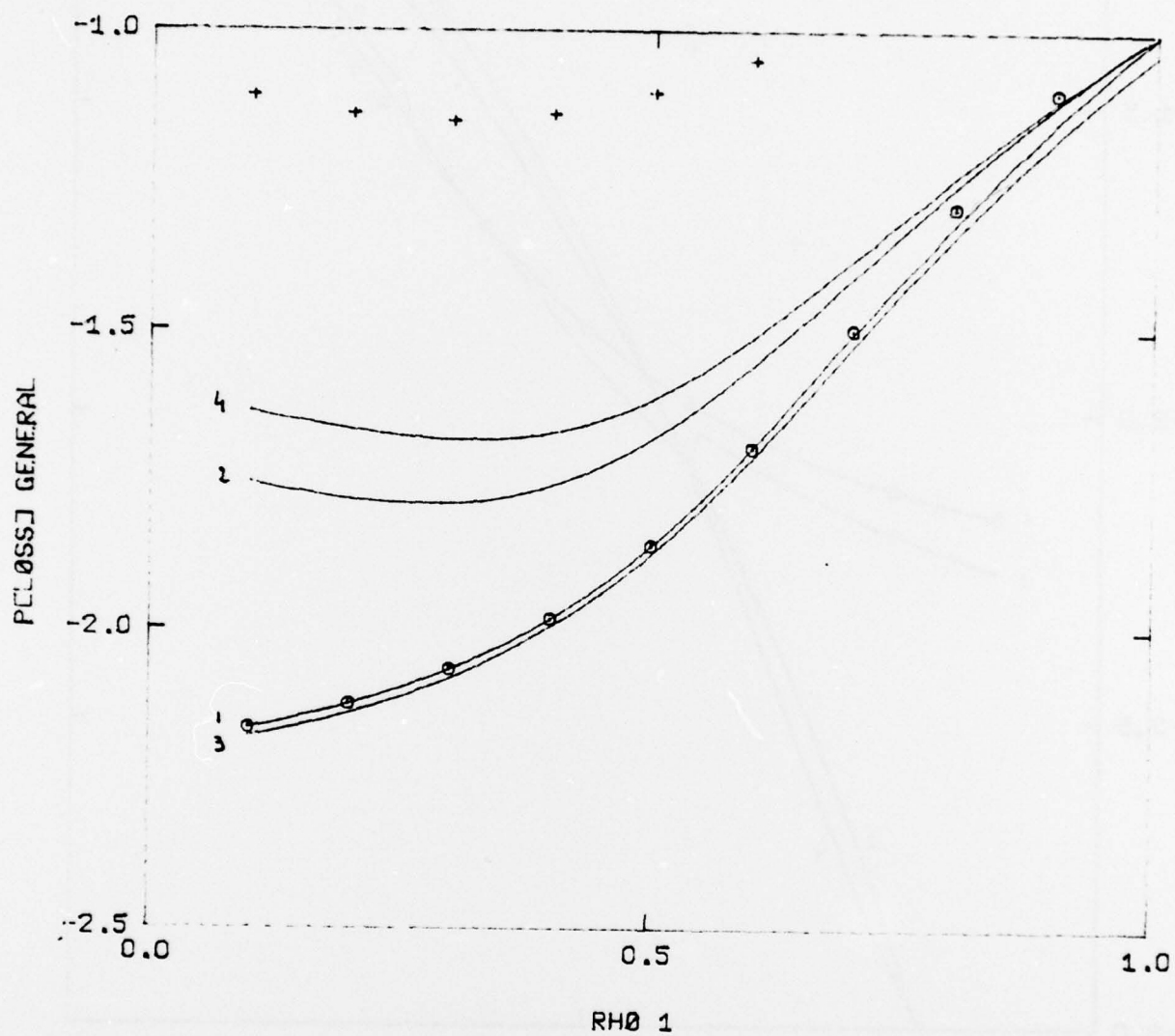


Figure 4 Loss probability π ; unbalanced queue; $L=3$, $M=10$, $\rho_2=.5$, $\rho_3=.6$

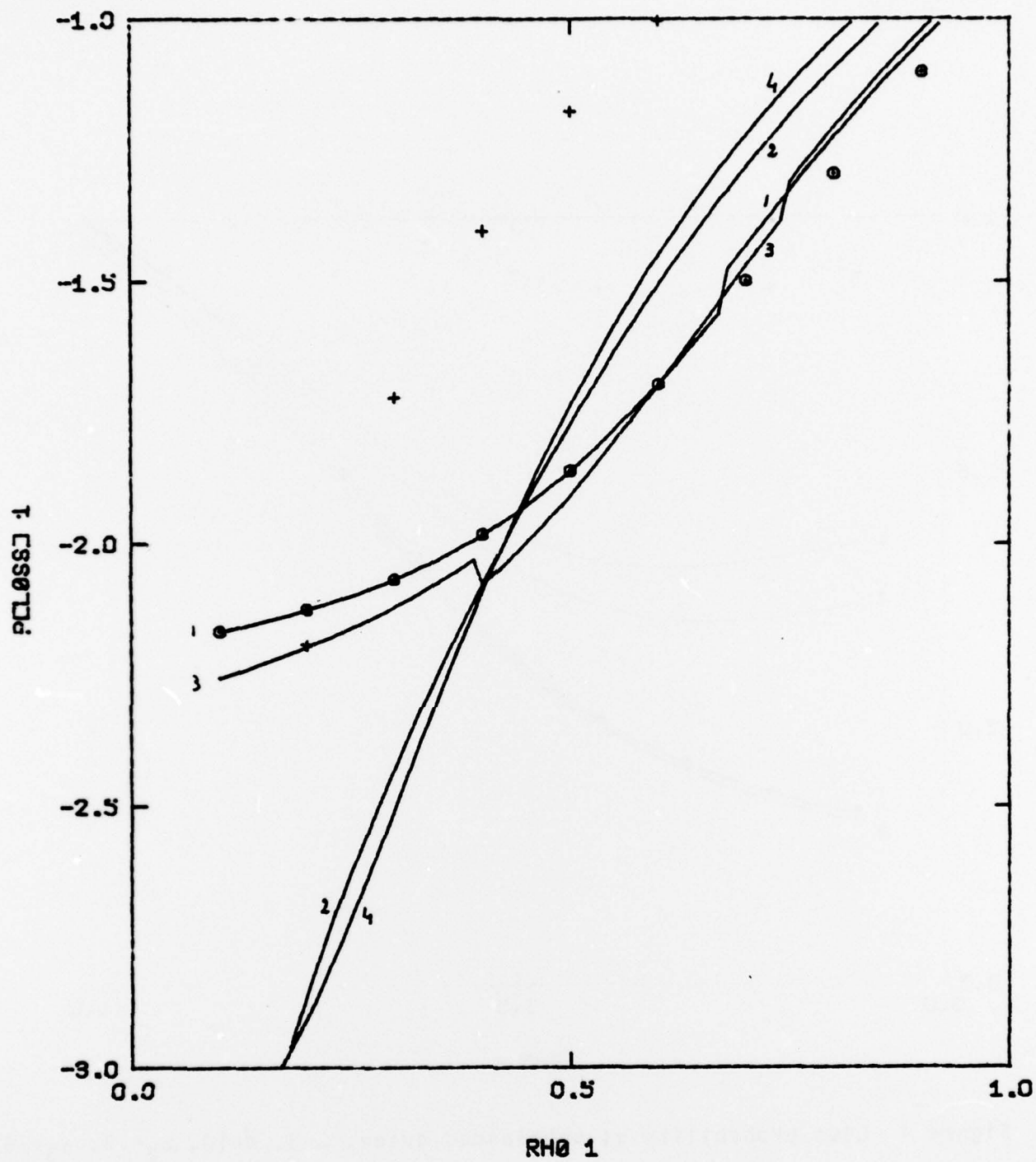


Figure 5 Loss probability for 1-messages: π_1 ; unbalanced queue; $L=3$, $M=10$, $\rho_2=.5$, $\rho_3=.6$

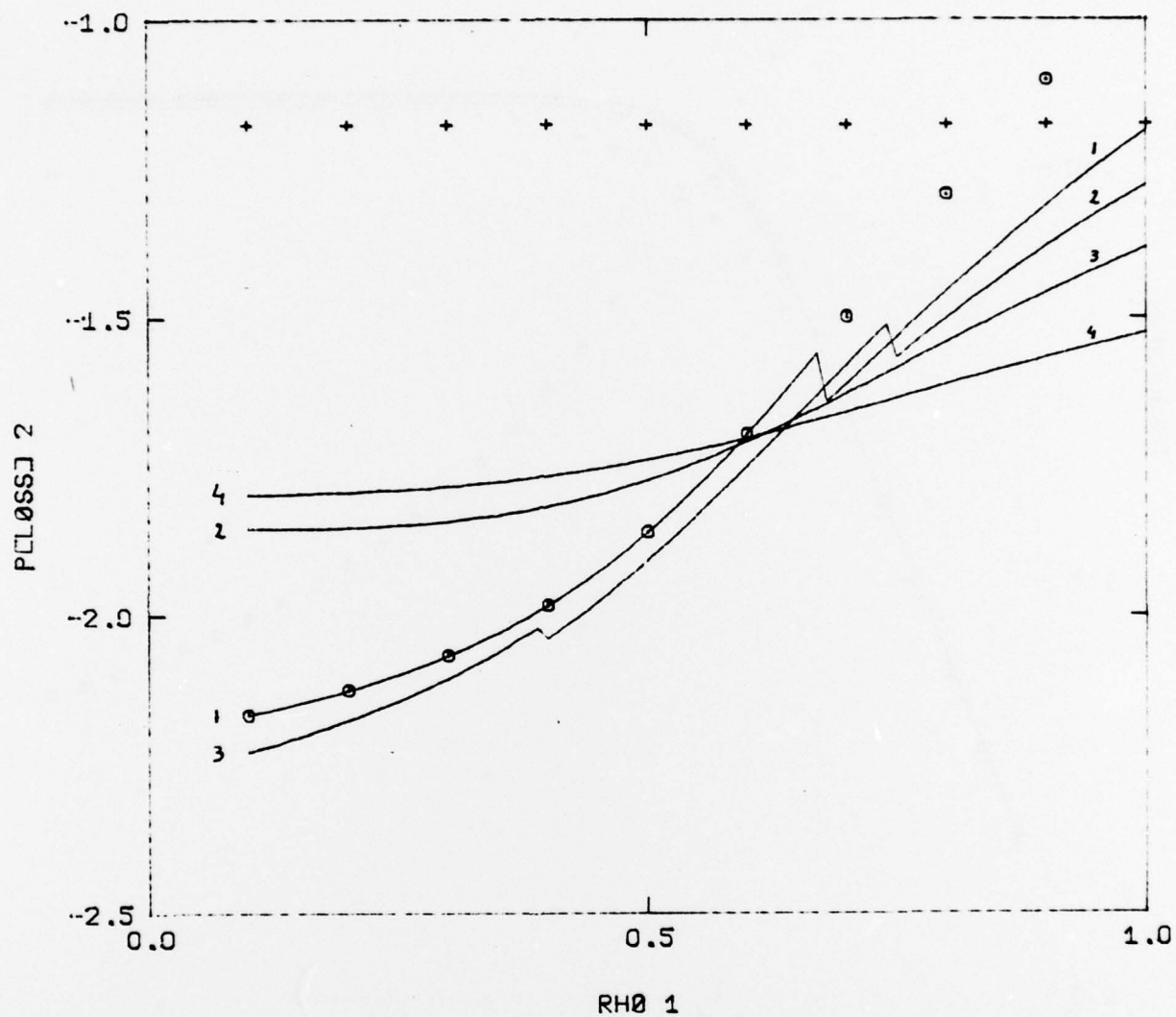


Figure 6 Loss probability for 2-messages: π_2 ; unbalanced queue;
 $L=3$, $M=10$, $\rho_2=.5$, $\rho_3=.6$

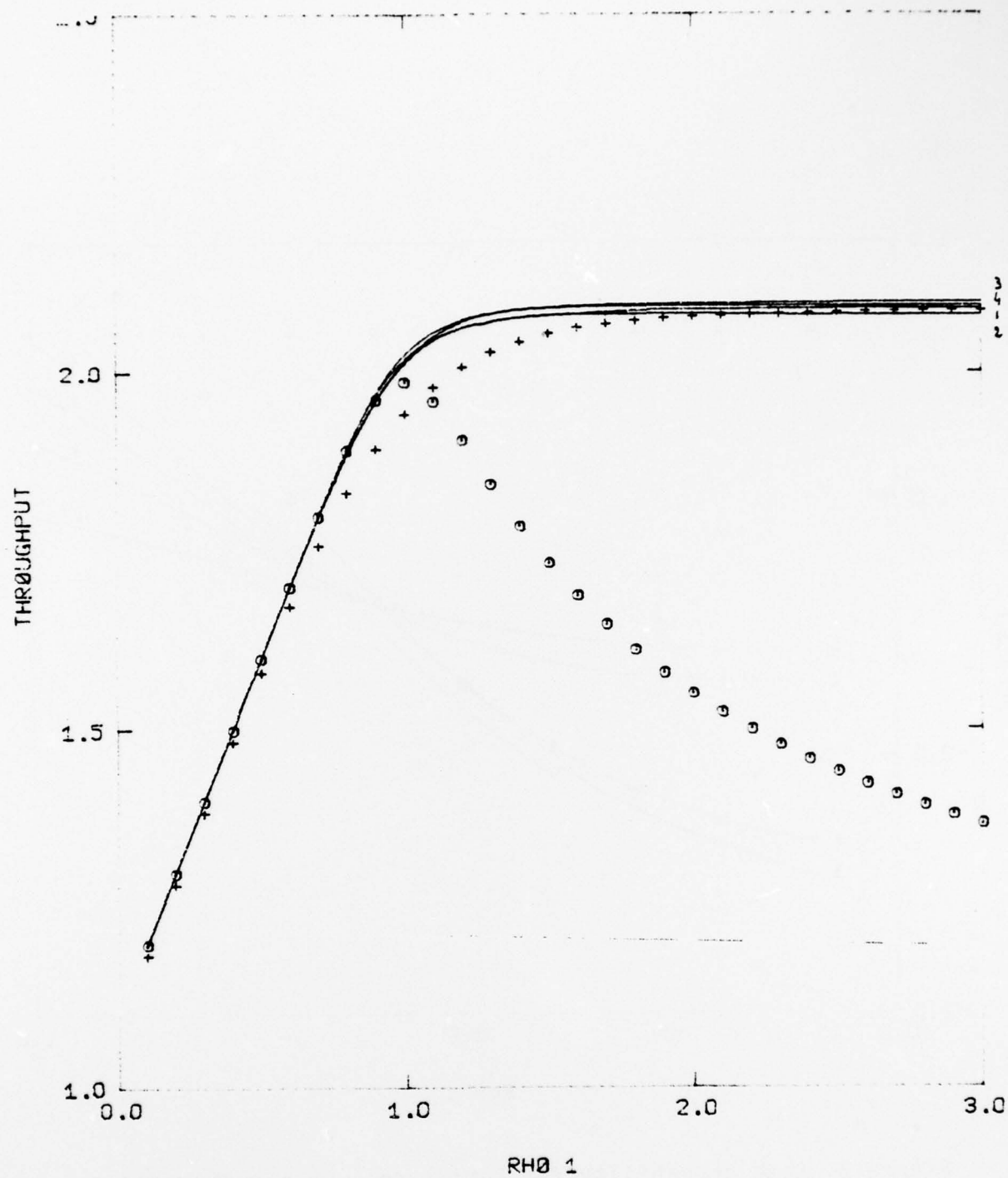


Figure 7 Throughput; unbalanced queue; $L=3$, $M=20$, $\rho_2=.5$, $\rho_3=.6$

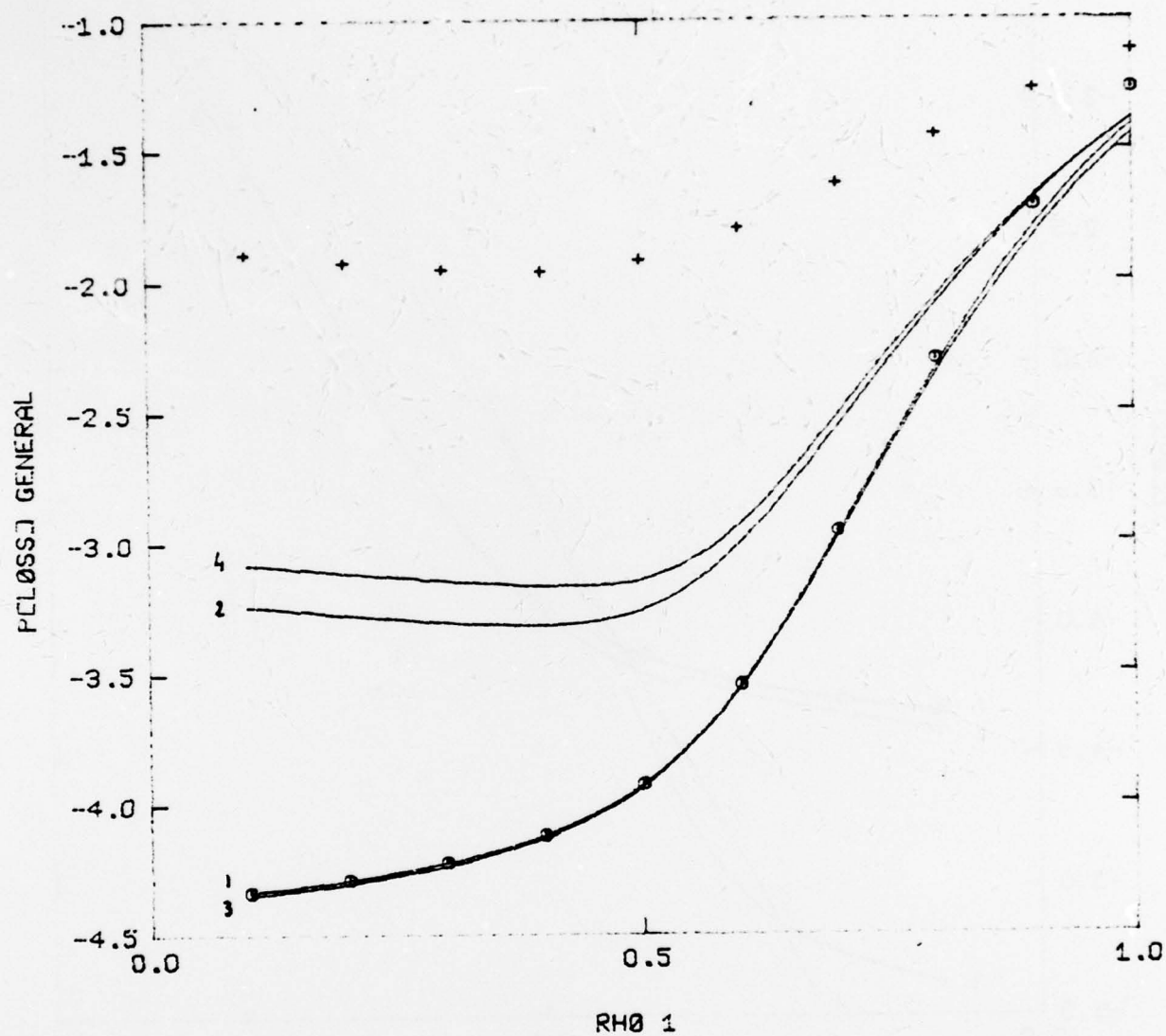


Figure 8 Loss probability π ; unbalanced queue; $L=3$, $M=20$, $\rho_2=.5$, $\rho_3=.6$

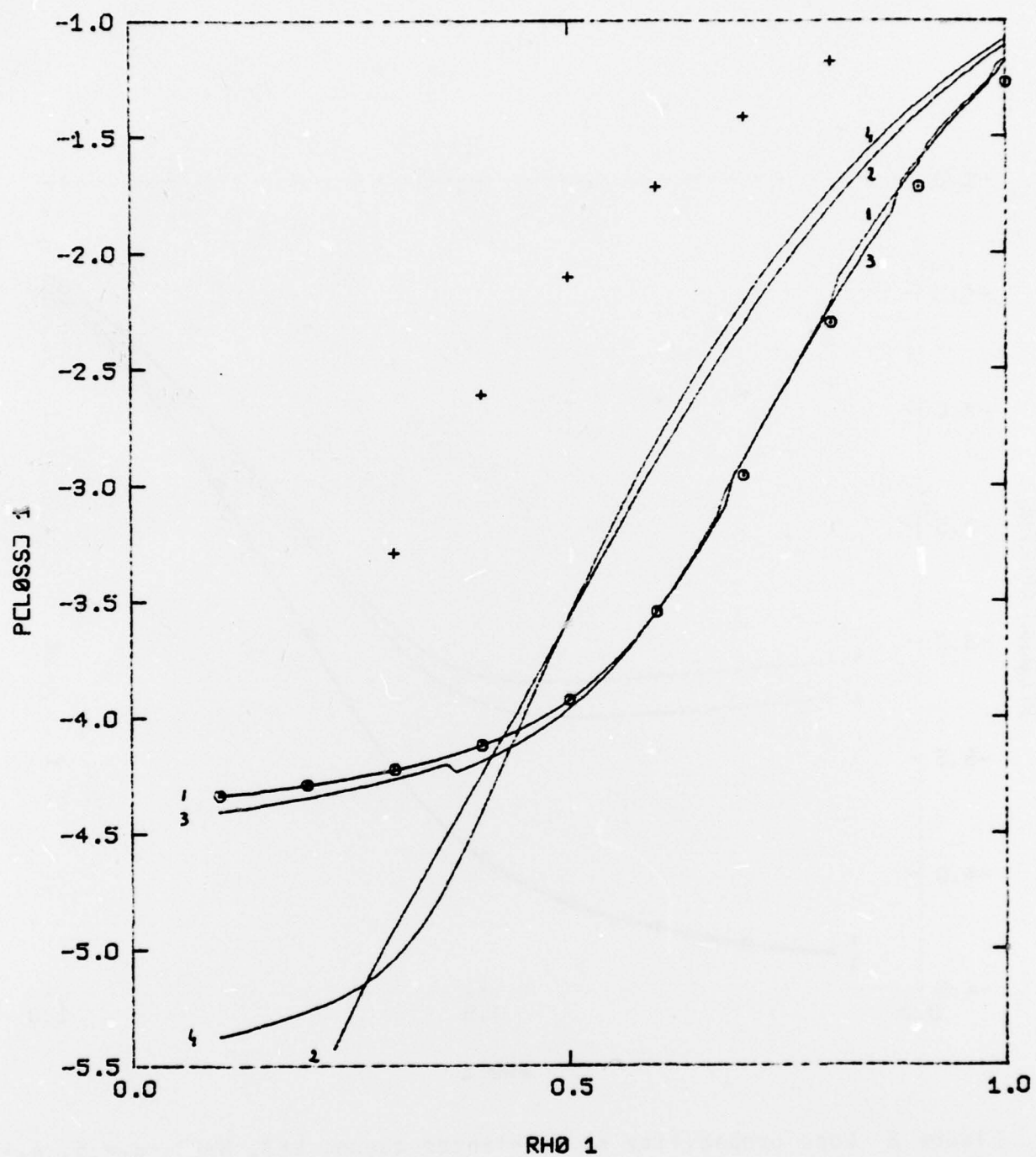


Figure 9 Loss probability for 1-messages: π_1 ; unbalanced queue; $L=3$, $M=20$, $\rho_2=.5$, $\rho_3=.6$

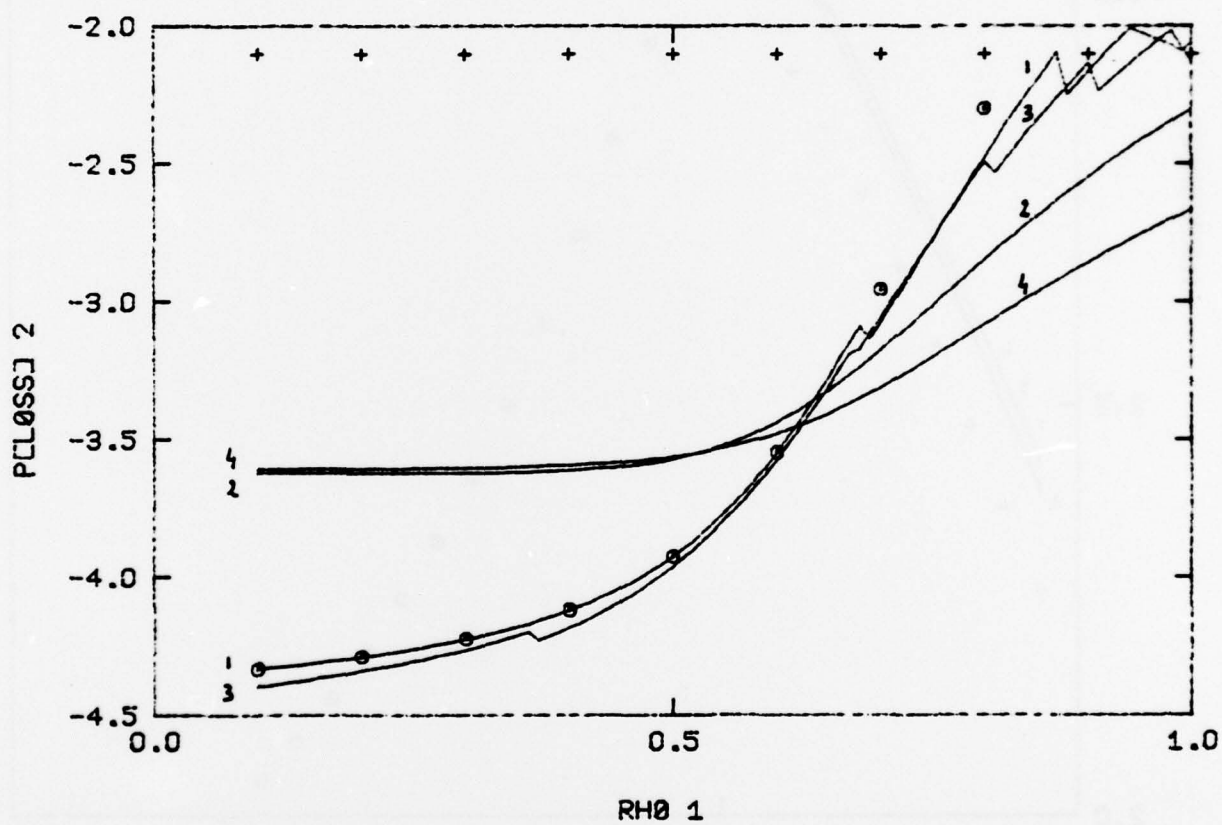


Figure 10 Loss probability for 2-messages: π_2 ; unbalanced queue;
 $L=3$, $M=20$, $\rho_2=.5$, $\rho_3=.6$

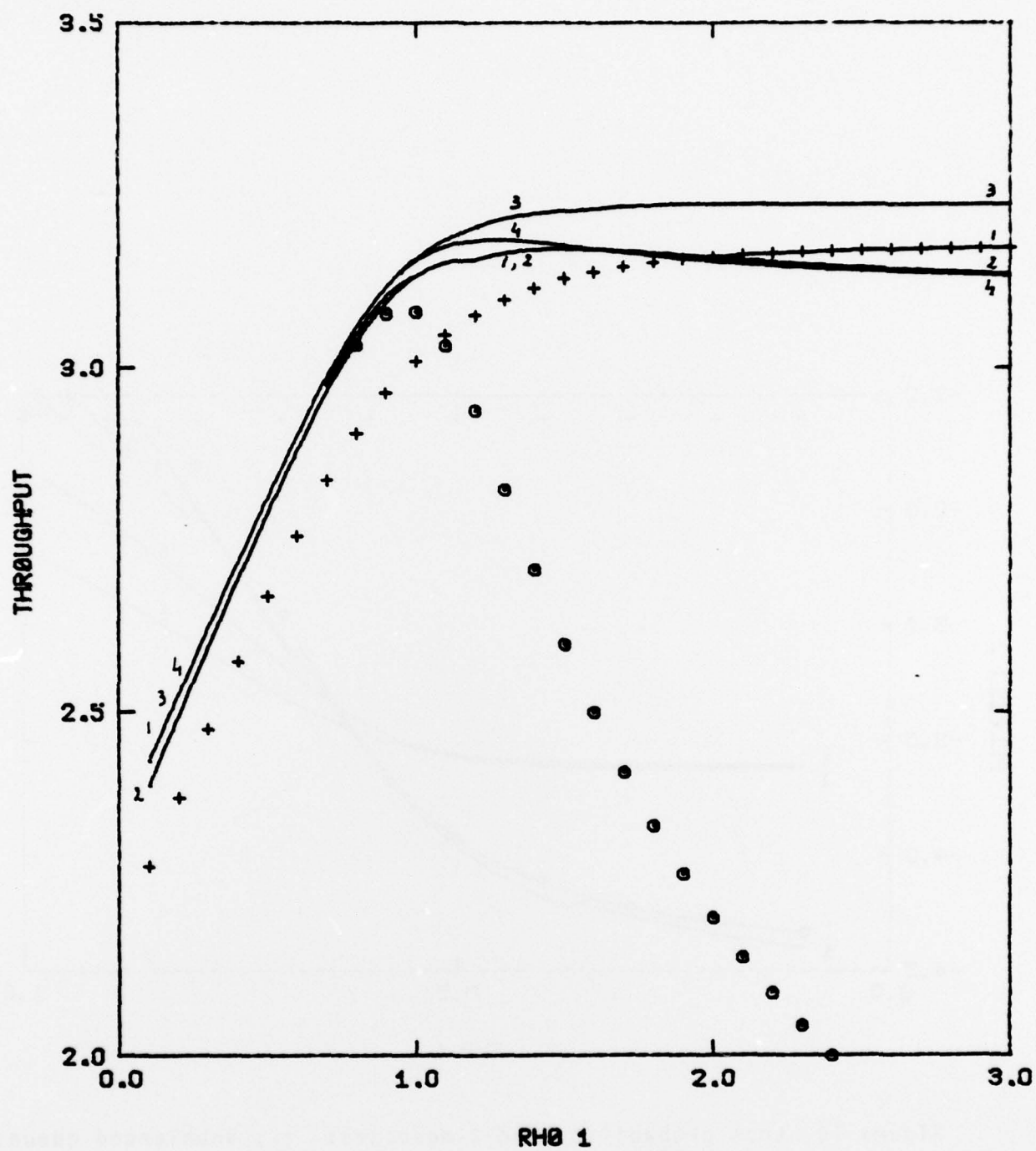


Figure 11 Throughput; highly unbalanced queue;
 $L=4$, $M=20$, $\rho_2=.7$, $\rho_3=.8$, $\rho_4=.9$

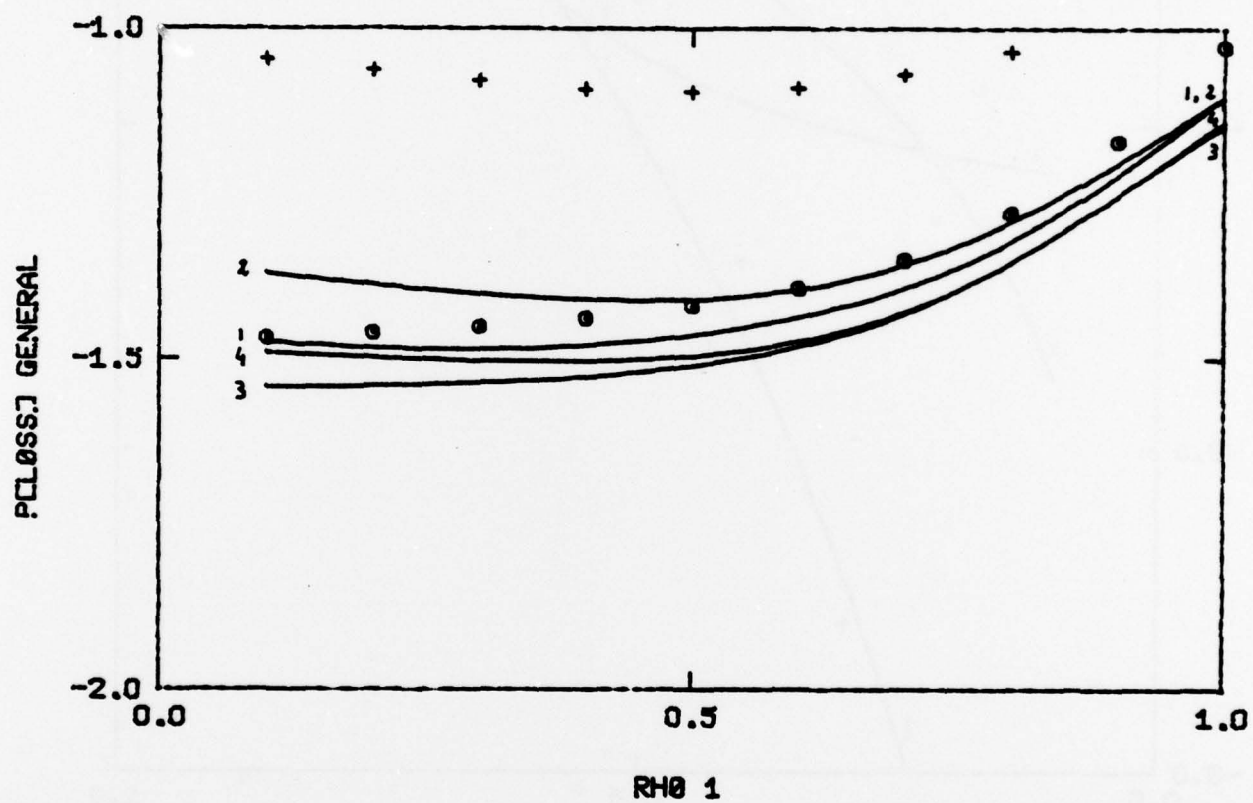


Figure 12 Loss probability π ; highly unbalanced queue;
 $L=4$, $M=20$, $\rho_2=.7$, $\rho_3=.8$, $\rho_4=.9$.

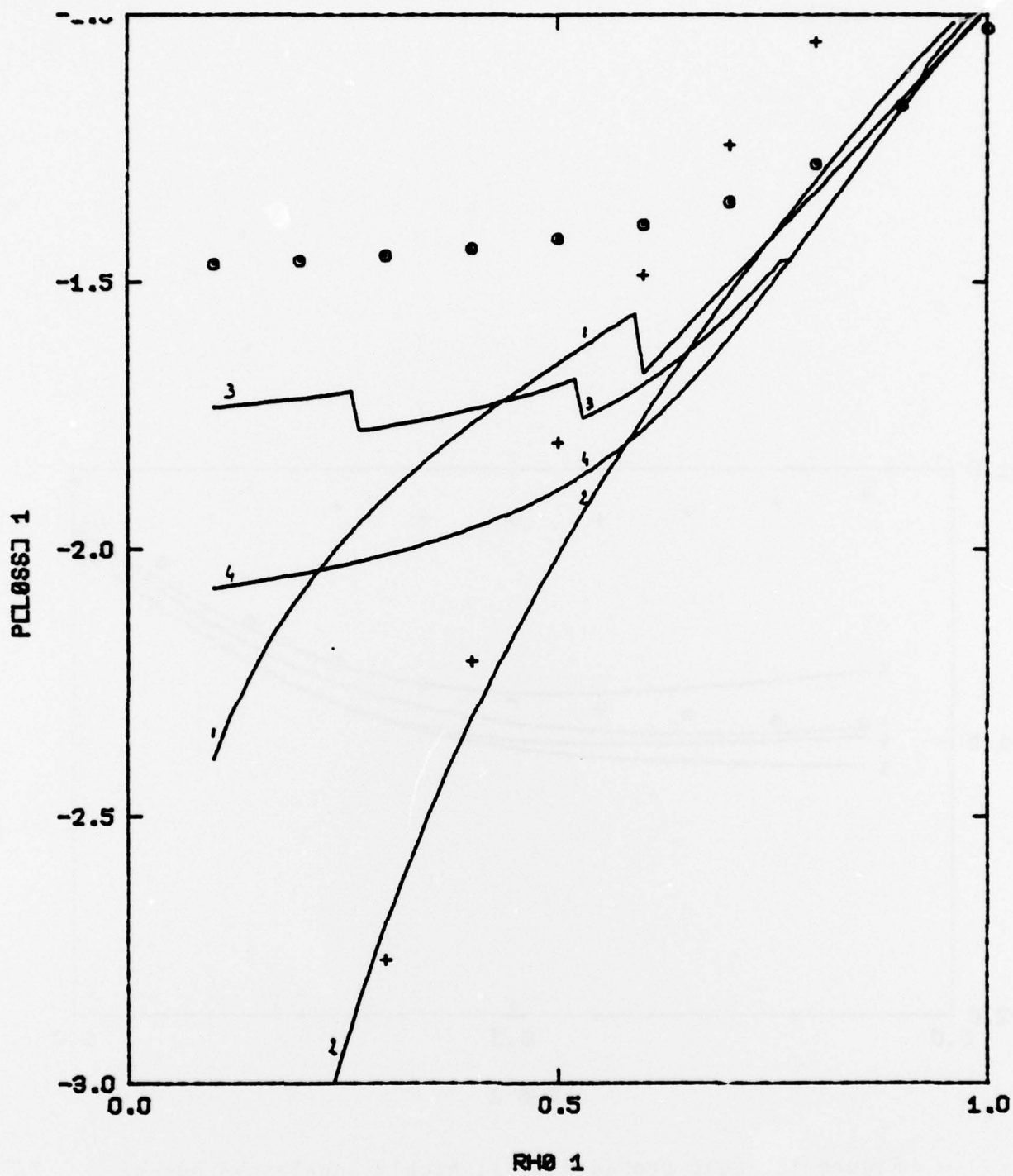


Figure 13 Loss probability for 1-messages: π_1 ;
highly unbalanced queue;
 $L=4$, $M=20$, $\rho_2=.7$, $\rho_3=.8$, $\rho_4=.9$

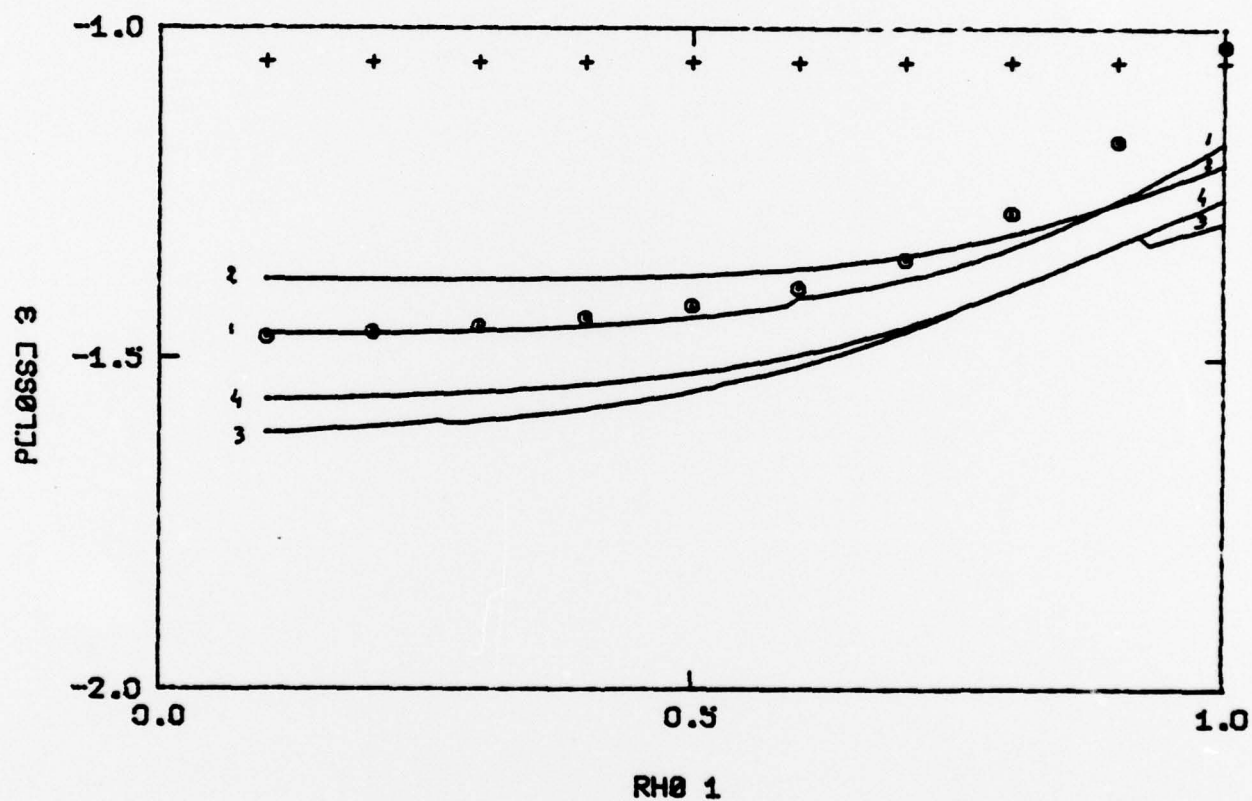


Figure 14 Loss probability for 3-messages: π_3 ;
highly unbalanced queue;
 $L=4$, $M=20$, $\rho_2=.7$, $\rho_3=.8$, $\rho_4=.9$

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 78-1142	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) EXPONENTIAL SERVERS SHARING A FINITE STORAGE: COMPARISON OF SPACE ALLOCATION POLICIES		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Guy Latouche		8. CONTRACT OR GRANT NUMBER(s) AFOSR-77-3236
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Delaware Department of Statistics & Computer Science Newark, Delaware 19711		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A5
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NM Bolling AFB, Washington, DC 20332		12. REPORT DATE May 1978
		13. NUMBER OF PAGES 45
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Exponential queue, finite storage, space allocation policy numerical comparisons		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Consider a finite waiting room shared by several servers. Such a system may approximate, for instance, a packet switch in a communications network or a buffer shared by producer-consumer parallel processes. It is assumed that a customer is lost if he cannot enter the waiting room. Furthermore, the waiting room is partitioned into (L+1) areas, where L is the number of servers: each of the first L areas is reserved		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract

for the exclusive use of one server, the $(L+1)$ st is common and may be used by all servers. ~~We refer to this allocation policy as~~ $(L+1)$ -sharing. *is referred to*

The steady-state probabilities are determined and the performance of $(L+1)$ -sharing, in terms of the probability that a customer is lost, is compared to three other policies:

-restricted sharing (an upper bound is imposed for the queue length of each server),

-unrestricted sharing (the $(L+1)$ st area occupies the whole waiting room),

-no sharing (the $(L+1)$ st area is empty).

It appears very clearly that the last two policies are not the best; a choice between $(L+1)$ -sharing and unrestricted sharing would depend on the objectives of the designer.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)