Modeling Multiprocessor Computer Systems with Unbalanced Flows

Stuart E. Madnick
Sloan School of Management
Massachusetts Institute of Technology

Y. Richard Wang
Department of Management Information Systems
College of Business and Public Administration
University of Arizona

ABSTRACT:

A performance analysis methodology using certain aspects of queueing theory to evaluate computer system speed performance is presented. This methodology specifically focuses on modeling multiprocessor computer systems with unbalanced flows (i.e., number of transactions leaving a server is not the same as number of transactions entering that server) due to asynchronously spawned parallel tasks. This unbalanced flow phenomenon, which has a significant effect on performance, cannot be solved analytically by classical queueing network models.

A decomposition method is applied to decompose the unbalanced flows. Formulae for open queueing networks with unbalanced flows due to asynchronously spawned tasks are developed. Furthermore, an algorithm based on Buzen's convolution algorithm is developed to test the necessary and sufficient condition for closed system stability as well as to compute performance measures. An average of less than four iterations is reported for convergence with this algorithm.

A study of the INFOPLEX multiprocessor data storage hierarchy, comparing this rapid solution algorithm with simulations, has shown highly consistent results. A cost effective software tool, using this methodology, has been developed to analyze an architectural design, such as INFOPLEX, and to produce measures such as throughput, utilization, and response time so that potential performance problems can be identified.

1. INTRODUCTION

1.1 FOCUS OF THE RESEARCH

This paper models Unbalanced flows due to Asynchronously spawned Parallel tasks (UAP) in a multiprocessor computer system using generalized queueing network models. The acronym UAP will be used throughout the paper to refer to unbalanced flows (i.e. the number of transactions leaving a server is not the same as the number of transactions entering that server) due to asynchronously spawned parallel tasks which are assumed to run independently of each other except for resource contention.

The significance of this research is as follows: (1) A decomposition method is applied to incorporate the workload due to UAP, an important factor on speed performance, into queueing network models; (2) Formulae for open queueing networks with UAP are developed; (3) The infeasibility of a product form solution for closed queueing networks with UAP is shown; (4) An algorithm based on Buzen's convolution algorithm is developed to test the necessary and sufficient condition for closed system stability; and (5) Approximate solutions for closed queueing networks with UAP are developed based on the algorithm. As a result, performance measures are assessed more accurately and cost effectively. A software tool, using this methodology, has been developed (Wang and Madnick, 1984) to analyze system architectures with UAP so that performance implications can be identified early in the design process.

The discussion in this paper uses as an example the INFOPLEX data storage hierarchy (INFOPLEX is a database computer research project at the MIT Center for Information Systems Research; the theory of hierarchical decomposition is applied in this research to structure hundreds of microprocessors together to realize a data storage hierarchy with very large capacity and small access time. See Madnick, 1973, 1977, 1979, Lam and Madnick, 1979, and Wang and Madnick, 1981, 1984). However, the methodology employed in this research is generalizable to other distributed information systems (Trivedi and Sigmon, 1981, Geist and Trivedi, 1982, and Goyal and Agerwala, 1984).

1.2 PURPOSE AND BACKGROUND OF THE RESEARCH

The use of analytic performance models, instead of simulation models, is important because the analytic approach is more cost effective than simulation. For complex multiprocessor systems, such as INFOPLEX, hundreds of thousands of simulation iterations were needed to reach steady-state for a single design alternative analysis. In order to obtain accurate results, all significant factors effecting performance should be captured in the analytic model. UAP has been found to be a common situation that has a primary effect on performance (Wang and Madnick, 1981, 1984). For example, in response to receiving a block of data, a processor may send an acknowledgement transaction, and also forward the data to another processor. Thus, there are now two transactions departing the server from the original data transmission transaction. This is called unbalanced flow since the number of transactions arriving at the server (i.e. 1) does not equal the number departing (i.e. 2). As another example, a processor may broadcast a message, via communication lines, to other processors, thereby spawning multiple transactions. Unfortunately, networks with UAP did not have an analytically tractable solution because the input flow and the output flow are not balanced at the places where parallel tasks are spawned, a violation of the principle of job flow balance (Denning and Buzen, 1978: the principle of job flow balance says that the number of customers that flow into a service facility equals the number of customers that flow out of the facility when the system is in the steady-state).

1.3 LITERATURE REVIEW

Several studies have been attempted to generalize queueing network models to include parallel processing. Browne, Chandy, Horgarth, and Lee (1973) investigated the effect on throughput of multiprocessing in a multiprogramming environment using the central server model approach. Reiser and Chandy (1979) studied the impact of distributions and disciplines on multiple processor systems. Towsley, Chandy, and Browne (1978) developed approximate queueing models for internal parallel processing by individual programs in a multiprogrammed system based on the central model approach and "Norton theorem". Price (1975) analyzed models of multiple I/O buffering schemes. Maekawa (1976) and Peterson (1979) modeled a number of CPU:IO overlap cases. These studies, although valuable, do not fit systems which have a generalized topology and the UAP phenomenon.

Modeling the UAP phenomenon for generalized queueing network systems is a relatively new topic, first reported, to our knowledge, by Heidelberger and Trivedi in 1982. In that work, an approximate solution method was developed and results of the approximation were compared to those of simulations. Mean value analysis approximation techniques were proposed for local area distributed computer systems with UAP by Goldberg, Popek, and Lavenberg (1983).

It is perhaps interesting to note at this point that, quite independently from the above research, the authors of this paper developed what they have called "Distributed Systems with Unbalanced Flows" (Wang and Madnick, 1981, 1984) starting in 1981. The technique used to model UAP is very similar but a different algorithm is used to test the necessary and sufficient condition as well as to compute the closed network throughput. Moreover, the results for open networks with UAP, such as response time, have been analyzed. A syntactic definition has also been given to decompose a model uniquely.

A terminal-oriented system and a batch-oriented multiprogramming system are modeled by Heidelberger and Trivedi (1982), and local area distributed systems are modeled by Goldberg and others (1983) while a hierarchically decomposed architecture is modeled in the INFOPLEX research (Wang and Madnick, 1984). The consistency reported from modeling these different architectures provides further validation of the modeling technique. The UAP model is described in the next section.

2. MODEL STRUCTURE

Without loss of generality, let's assume that all customers are homogeneous, i.e. there is a single customer type. It would be easy to relax this assumption to include different types of customers. Without loss of generality, transaction examples will be drawn from the INFOPLEX data storage hierarchy analysis (e.g., read, write, and acknowledgement transactions).

Let there be M service facilities and C classes in a queueing network. A service facility may consist of several classes which allow customers to have different sets of routing probabilities for different visits. Assume that any source and sink belong to class 0. Let $p_{i,j}$ denote the routing probability which is the fraction of customers completing service in class i that join class j, i = 0, ..., C; j = 0, ..., C; and $p_{0,0} = 0$ by convention.

A main chain is defined as the path through which customers travel according to the defined routing probability and eventually go out of the system to return to the reference source. Since all customers have been assumed to be homogeneous, there is only one main chain in the system.

A class c customer of facility m in the queueing network is said to be UAP with degree b if its output splits into b branches where b is an integer number greater than one. Note that (a) UAP can occur in many classes within a queueing network, for instance acknowledgements may take place at different levels of a data storage hierarchy; and (b) the inputs to a class that cause UAP can be the outputs from other UAP classes. For instance, a split from an acknowledgement may split again to send more acknowledgements to other classes.

Consider a class which is UAP with degree b, the main task that eventually returns to the reference source is defined to belong to the main chain; on the other hand, the b-1 additional flows which cause that class to be unbalanced are perceived as "internal sources" (denoted as SOURCEU) which generate customers to travel within the network and eventually terminate at the "internal sink" (denoted as SINKU). It follows, as will be justified in section 3, that all the classes with UAP can be separated from the main chain to form the UAP chain where the UAP chain is defined as the additional path through which the "internally generated" customers (from SOURCEU) travel and eventually sink (at SINKU). Note that SOURCEU may combine multiple "internal sources".

By labeling the source and sink of the main chain as SOURCEM and SINKM, and the source and sink of the UAP chain as SOURCEU and SINKU, one can decompose the graph of a network model with UAP unambiguously without referring to the semantics of the model. In other words, given the labeled graph of an UAP network, it is impossible to interchange one of the UAP flows with a part of the main chain. Therefore, a unique syntactic definition exists for each UAP network.

Classical queueing network models cannot be applied to analyze UAP directly because of the unbalanced flows mentioned. An extended routing matrix is introduced below to accommodate the problem. Let R denote the extended routing matrix of an UAP network where a row-sum may be greater than one. Let R_c denote the unextended routing matrix which excludes the UAP chain of the network. Elements in R and R_c are the routing probabilities $p_{i,j}$'s.

Define the visit ratio of a class, V_c, as the mean number of requests of the class to a service facility per customer. Define the sum of visit ratios of all exogenous sources, V_o, in an open system to be one. In a closed system, the outputs feedback to the system inputs; the sum of visit ratios of the system inputs is also defined to be one.

The visit ratios of the classes in R_c can be obtained from the visit ratio equations (5, p.237), viz.,

$$V_j = p_{o,j} + \sum_{i=1}^{C} V_i * p_{i,j}$$
$$j = 1, \ldots, C.$$

Alternatively, the visit ratio equations can be applied directly to the extended routing matrix R to obtain all the visit ratios of the classes in R.

3. ANALYTIC FORMULATION

3.1 QUEUING NETWORKS WITH UAP

To present the paper concisely, service rate is assumed to be fixed, and typical queuing network notations are used (see the Appendix for a summary of notations). It was noted, in section 2, that a) UAP can occur in many classes within a queueing network; b) an input to a class that causes UAP may be the output from another class in the UAP chain; and c) all the additional unbalanced flows are defined to belong to the UAP chain -- a single chain. It is natural to ask whether the flows of the transformed network would be balanced, and what kind of relationship would exist between the main chain and the UAP chain. These questions are answered below:

If one cuts the additional b-1 unbalanced flows from a class which is UAP with degree b and inserts "internal sources" (SOURCEU) which generate customers with equivalent flow rates as those of the network before the cut, then following the assumption that unbalanced flows run independently of one another except for resource contention, the b-1 unbalanced flows will form b-1 new open chains which will not interact with the main chain. If all the additional unbalanced flows (spawned from the classes which are UAP and connected to the main chain) are cut from the main chain, then the flow in the main chain will be balanced.

Let {R} denote the set of classes in the network before the cuts and $\{R_c\}$ denote the set of classes in the main chain. It follows that we have the balanced main chain with its classes in the set $\{R_c\}$ and many open chains with their classes in the set $\{R\} - \{R_c\}$. Therefore, the classes in the main chain and the classes in the open chains are disjoint.

However, it has been pointed out, in section 2, that a split may split again, so the open chains may themselves be flow unbalanced. To solve the problem, it is logical to cut all the additional unbalanced flows in the open chains continuously (and insert "internal sources" which generate equivalent flow rates as those of the open chains before the cuts) until all flows are balanced, as a result, additional open chains are formed as part of the UAP chain.

It is assumed that service time distributions and service disciplines of the facilities in the network follow those of classical product form queueing networks (Baskett et al); in addition, the unbalanced flows which run independently of one another are assumed to arrive at their destinations as independent poisson processes (this assumption is also adopted by Goldberg (1983) and Heidelberger (1983)). However, as Burke (1972) pointed out, these processes are not Poisson in general. The simulation studies that the authors have conducted indicate that this is a fairly robust approximation. The validation reported by Goldberg, et al (1983) provides further support for this assumption. It follows that the open product form multiple chain queueing network (OPFMCQN) result can be applied to aggregate the additional open chains discussed in the last paragraph to a single open chain -- the UAP chain.

If the original network is an open network, then the OPFMCQN result can be applied again to make the overall network a single chain with its workload contributed from both the main chain and the UAP chain; section 3.2 discusses the formulation of useful performance measures for open queueing networks with UAP. On the other hand, if the original network is a closed network, then we have a mixed network with the closed main chain and the open UAP chain. Section 3.3 discusses the necessary and sufficient condition for the closed network to be stable and an iterative procedure which computes the system throughput. Let V(U), denote the internally generated visit rate of the UAP chain. Note that "(M)" will denote an open chain in section 3.2 and a closed chain in section 3.3.

3.2 OPEN QUEUEING NETWORKS WITH UAP

For an open queueing network with UAP, the network arrival process is assumed to be poisson with a constant rate λ_o. By solving the extended routing matrix introduced in section 2, one can obtain the visit ratios for all classes, hence V(U). Since λ_o is given, $X_o(U)$ is also determined, specifically, $X_o(M) = \lambda_o$ and $X_o(U) = \lambda_o * V(U)$. It can be shown (Sauer 1981, Lazowska 1984) that throughput, utilization, mean queue length, and response time for FCFS, PS, and LCFSPR disciplines are computed as shown in Table 3-1. The "system response time" in a flow unbalanced network is defined as the time to complete the main chain since that is the only observable completion seen by the external world.

Facility i	Formulae
$X_i(M)$	$X_o(M) * V_i(M)$
$X_i(U)$	$X_o(U) * V_i(U) / V(U)$
X_i	$X_i(M) + X_i(U)$
$U_i(M)$	$X_i(M) * S_i(M)$
$U_i(U)$	$X_i(U) * S_i(U)$
U_i	$U_i(M) + U_i(U)$
$N_i(M)$	$U_i(M) / (1-U_i)$
$N_i(U)$	$U_i(U) / (1-U_i)$
N_i	$N_i(M) + N_i(U)$
$R_i(M)$	$N_i(M) / X_i(M)$
$R_i(U)$	$N_i(U) / X_i(U)$
R_i	N_i / X_i
$R_o(M)$	$R_1(M) + \ldots + R_C(M)$
R_o	$R_1 + \ldots + R_C$

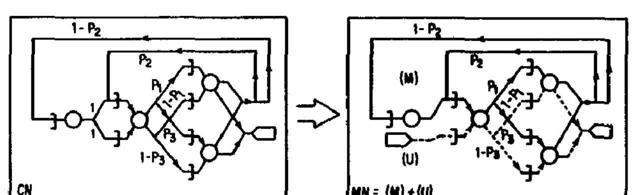
Table 3-1: Open Networks with UAP.

3.3 CLOSED QUEUEING NETWORKS WITH UAP

For closed queueing networks with UAP, a mixed network with the closed main chain and the open UAP chain, as illustrated in Figure 3-1, can be obtained following the discussion in section 3.1. Since $X_o(U) = X_o(M) * V(U)$ where $X_o(M)$ is evaluated through a nonlinear function of $X_o(U)$ (Reiser and Kobayashi, 1975). It follows that $X_o(U) = f(X_o(U)) * V(U)$ where f is a nonlinear function. To solve the nonlinear equation, a couple of issues have to be addressed first:

A) What are the properties of f?
B) what is the necessary and sufficient condition for the network to be stable?

In Wang and Madnick (1984) a corollary based on Reiser and Kobayashi's theorem (1975) on product form mixed queueing networks (PFMQN) is developed to settle issue A and a lemma is proven to settle issue B which leads to an iterative procedure for the closed network. The corollary and lemma are summarized below. The infinite server (IS) discipline is excluded from this subsection. Its difference from other disciplines will be discussed at the end of the section.



Original Closed Network with UAP (CN) — Mixed Network without UAP (MN)

Figure 3-1 Decomposition of CN to MN

A) **Corollary:** An equivalent closed network (EN) of the main chain for the mixed network(MN), as illustrated in Figure 3-2, can be obtained by inflating the main chain traffic intensities.

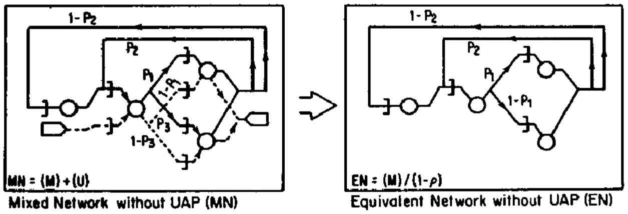


Figure 3-2 Transformation of MN to EN

From the marginal distribution above, it is not difficult to see (13,28) that f is continuous and monotonically decreasing (CMD), assuming that there exists at least a pair of $(D_i(M), D_i(U))$ such that $D_i(M)>0$ and $D_i(U)>0$. With the corollary and the CMD property, the convolution algorithm can be applied to solve the nonlinear equation iteratively. Let $()^i$ denote the ith iteration. For instance, $(EN(X_o))^5$ denotes the throughput of EN at the 5th iteration. In the iterative procedure, $(X_o(U))^0$ is estimated initially by the lemma described later and $(X_o(U))^{i+1}$is determined as follows:

$$(X_o(U))^{i+1} = (EN(X_o))^{i+1} * V(U)$$

where $(EN(X_o))^{i+1} = f((X_o(U))^i)$.
This relationship is used below to settle issue B.

B) The stability of PFMQN is unaffected by the presence of closed chains (Lazowska, 1984). Define the facility with maximum open chain utilization (i.e. the bottleneck facility) as facility I. It follows that a closed network with UAP is stable if and only if $U_I(U)<1$. Note that $U_I(U) = (X_o(U) / V(U)) * D_I(U)$. Denote $V(U)/D_I(U)$ as B, the maximum throughput of the bottleneck facility, i.e., at saturation. It follows that a closed queueing network with UAP is stable if and only if $X_o(U) < B$.

Denote $D_I(M)$ as the main chain D value at the bottleneck facility I.The stability condition of the closed network with UAP can then be identified with the following four mutually exclusive and collectively exhaustive cases:

I) $f(X_o(U)=0) * V(U) < B$;
II) $f(X_o(U)=0) * V(U) \geq B$, but $D_I(M) > 0$;
III) $f(X_o(U)=0) * V(U) \geq B$, $D_I(M) = 0$, but $f(X_o(U)=B) * V(U) < B$;
IV) $f(X_o(U)=0) * V(U) \geq B$, $D_I(M) = 0$, and $f(X_o(U)=B) * V(U) \geq B$.

Figure 3-3 depicts the four conditions. Let $a = f(X_o(U)=0)$, $b = a*V(U)$, $c = f(X_o(U)=B)$, and $d = c*V(U)$, then the four cases can be rewritten as follows:

I) $b < B$;
II) $b \geq B$, but $D_I(M) > 0$;
III) $b \geq B$, $D_I(M) = 0$, but $d < B$;
IV) $b \geq B$, $D_I(M) = 0$, and $d \geq B$.

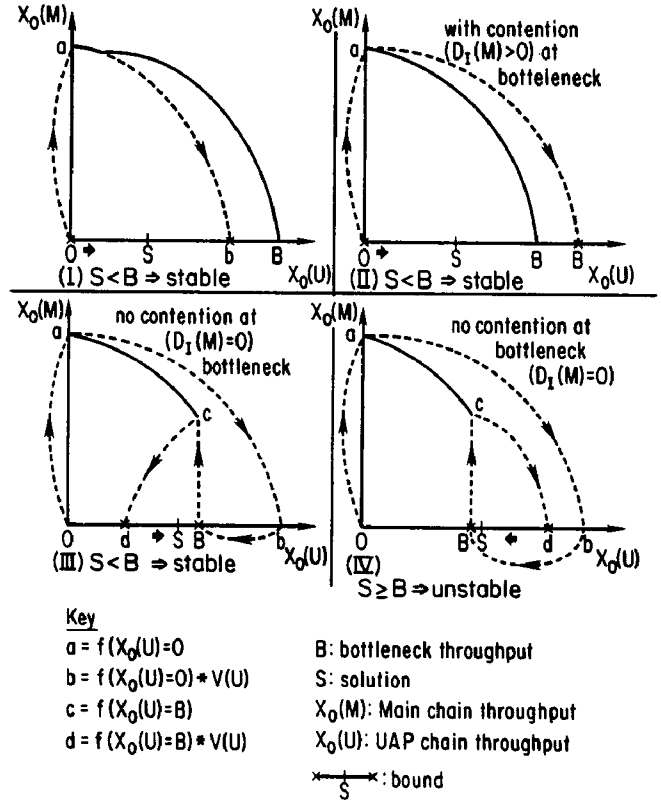


Figure 3-3 Stability Condition Test

It is proven (Wang and Madnick, 1984) that the network is stable if and only if it is not Case IV. Several points are worth noting:

a) The IS discipline was excluded since the main chain and the UAP chain do not interact with each other at the IS facility. For networks with mixed disciplines, the inflating factor for the IS facility is one. For networks with IS facilities only, the UAP chain has no impact on the main chain, therefore, can be ignored.
b) The Convolution algorithm, simple and efficient, is used to insure the stability condition as well as to locate the solution (Wang and Madnick, 1984).
c) The equivalent closed network obtained from the corollary is used to calculate the "system response time" perceived by the external world. Moreover, when the iterative procedure stops, G(1), ..., G(N) are also available as a byproduct for calculating useful performance measures.

d) It was found (Wang and Madnick, 1984) that a bounded interpolation algorithm takes an average of 2.4 iterations to produce relative errors less than 0.001 over 10,000 test cases.

e) A comparative study of the INFOPLEX data storage hierarchy has been conducted to assess the predictability of this technique. It has been observed (Wang and Madnick, 1984) that the analytic results are highly consistent with the simulations. A closer examination of the data shows that the analytic results deviate from the simulations by less than 2%.

4. CONCLUSIONS

An analytic approximation methodology has been developed to model multiprocessor computer systems with unbalanced flows due to asynchronously spawned tasks (UAP). The methodology allows us to assess useful performance measures that can be used by system designers to explore different design alternatives cost effectively. Whereas it has been costly to attain the steady-state results of a single design alternative of the INFOPLEX system using simulation economically, studies have shown (Wang and Madnick, 1984) that this methodology produces the same quality of results as simulation with less effort and at a fraction of the time and cost.

5. ACKNOWLEDGEMENTS

The authors wish to thank Professors Jeff Buzen, Bill Frank, Ugo Gagliardi, John Little, Anthony Wong, and the anonymous referees for their valuable comments on earlier versions of the paper. Work reported herein was supported, in part, by research contract N00039-83-C-0463 from the Naval Electronic Systems Command.

6. REFERENCES

1. Baskett, F., Chandy, K. M., Muntz, R. R., and Palacios, J. "Open, Closed, and Mixed Networks with Different Classes of Customers," JACM, April 1975, pp. 248-260.
2. Browne, J. C., Chandy, K. M., Hogarth, J., and Lee, C. "The Effect on Throughput in Multi-processing in a Multi-programming environment," IEEE Trans. Computers, Vol. 22, August 1973, pp. 728-735.
3. Burke, P. J. "Output Process and Tandem Queues," Symposium on Computer-Communications Networks and Teletraffic, Polytechnique Institute of Brooklyn, PP. 419-428, 1972.
4. Buzen, J. P. "Computational Algorithms for Closed Queueing Networks with Exponential Servers," CACM, Sept. 1973, pp. 527-531.
5. Denning P. J. and Buzen, J. P. "The Operational Analysis of Queueing Network Models," ACM Computing Surveys, Vol.10, No. 3, Sept. 1978, pp. 225-261.
6. Gagliardi, Ugo, Lectures on Software Engineering, Harvard University, 1982.
7. Geist, R. M., and Trivedi, K. S. "Optimal Design of Multilevel Storage Hierarchies," IEEE Transactions on Computers, March 1982.
8. Goldberg, A. ,Popek, G., and Lavenberg, S. "A Validated Distributed System Performance Model," Performance'83, pp. 251-268.
9. Gordon, W. J. and Newell, G. F. "Closed Queueing Systems with Exponential Servers," Operation Research 15 (1967), pp. 254-265.
10. Goyal, A. and Agerwala, T. "Performance Analysis of Future Shared Storage Systems," IBM J. Res. Develop., January 1984.
11. Graham, G. S. "Guest Editor's Overview: Queueing Network Models of Computer System Performance," ACM Computing Surveys, Vol. 10, #3, September 1978, pp 219-224.
12. Hamming, R. W. "Numerical Methods for Scientists and Engineers," 2nd ed. New York: Mc Graw-Hill, 1973.
13. Heidelberger, P. and Trivedi, K. S. "Analytic Queueing Models for Parallel Processing with Asynchronous Tasks," IEEE Transactions on Computers, November 1982.
14. Heidelberger, P. and Trivedi, K. S. "Analytic Queueing Models for Programs with Internal Concurrency," IEEE Transactions on Computers, January 1983.
15. Jackson, J. R. "Jobshop Like Queueing Systems," Management Science 10 (1963), pp. 131-142.
16. Kleinrock, L. "Queueing Systems I," John Wiley, New York, 1975
17. Kleinrock, L. "Queueing Systems II," John Wiley, New York, 1976
18. Lam, C. Y., "Data Storage Hierarchy Systems for Database Computers," Tech. Rep. #4, August 1979, MIT Sloan School.
19. Lam, C. Y. and Madnick, S. E., "Properties of Storage Hierarchy Systems with Multiple Page Sizes and Redundant Data," ACM Transactions on Database Systems, Vol. 4, No. 3, September 1979, pp. 345-367.
20. Lavenberg, S. S. "Computer Performance Modeling Handbook," Academic Press, 1983.
21. Lazowska, Zahorjan, Graham, and Sevcik, "Quantitative System Performance: Computer System Analysis Using Queueing Network Models," Prentice Hall, 1984.
22. Little, J. D. C. "A Proof of the Queueing Formula $L=\lambda W$," Operations Research 9, 383-387 (1961)

23. Madnick, S. E., "Storage Hierarchy Systems," Report No. TR-105, Project MAC, MIT, Cambridge, MA, 1973.
24. Madnick, S. E., "Trends in Computers and Computing: The Information Utility," Science, Vol. 185, March 1977, pp. 1191-1199.
25. Madnick, S. E., "The INFOPLEX Database Computer, Concepts and Directions," Proc. IEEE Comp. Con., February 1979, pp. 168-176.
26. Maekawa, M and Boyd, D. L. "Two Models of Task Overlap with Jobs of Multiprocessing Multiprogramming Systems," Proc. 1976 Int. Conf. on Parallel Processing, Detroit, August 1976, pp. 83-91.
27. Muntz, R. R. "Poisson Departure Processes and Queueing Networks," IBM Res. Rep. RC-4145, 1972.
28. Mayrhauser, A. V. and Kenevan, J. R. "Convexity of Reciprocal Throughput and Response Time for a Very General Class of Queueing Networks," Illinois Institute of Technology, June 1983.
29. Peterson, M. and Bulgren, W. "Studies in Markov Models of Computer Systems," Proc. 1975 ACM Annual Conf., Minneapolis, Minn., pp. 102-107.
30. Price, T. G. "Models of Multiprogrammed Computer Systems with I/O buffering," Proc. 4th Texas Conf. Comput. Syst , Austin, 1975.
31. Reiser, M. and Kobayashi, H. "Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms," IBM J. Res. Develop., May 1975, pp. 283-294.
32. Reiser, M. and Chandy, K. M. "The Impact of Distributions and Disciplines on Multiple Processor Systems," CACM, Vol.22, pp. 25-34, 1979.
33. Sauer, C. H. and Chandy, K. M. "Computer Systems Performance Modeling," Printice-Hall, Englewood Cliffs, New Jersey, 1981.
34. Sauer, C. H., MacNair, E. A., and Kurose, J. F. "The Research Queueing Package Version 2: Introduction and Examples," RA 138, 4/12/82 IBM Thomas J. Watson Research Center, Yorktown Heights, New York.
35. Towsley, D., Chandy, K. M., and Browne, J. C. "Models for Parallel Processing within Programs: Applications to CPU:I/O and I/O:I/O Overlap," CACM, Vol.21, pp. 821-831, 1978.
36. Trivedi, K. S. and Sigmon, T. M. "Optimal Design of Linear Storage Hierarchies," JACM April 1981, pp. 270-288.
37. Wang, Y. R. and Madnick, S. E. "Performance Evaluation of the INFOPLEX Database Computer," Tech. Rep. #13, MIT Sloan School, April 1981.
38. Wang, Y. R. and Madnick, S. E. "Performance Evaluation of Distributed Systems with Unbalanced Flows: An analysis of the INFOPLEX Data Storage Hierarchy," Tech. Rep. #15, MIT Sloan School, July 1984.

7. APPENDIX -- NOTATIONS USED

A) subscripts:

i	denotes an individual service facility.
o	denotes the overall network.
(M)	denotes the main chain.
(U)	denotes the UAP chain.
$()^i$	denotes the ith iteration.

B) notations:

B	bottleneck facility (therefore chain) throughput.
C	total number of classes in the network.
CMD	continuous and monotonically decreasing
D	V*S; the product of visit ratio and mean service time.
FCFS	first come first serve.
f	$X_o(M)=f(X_o(U))$; the main chain throughput as a nonlinear function of the UAP chain throughput.
IS	infinite server.
LCFSPR	last come first serve preemptive resumable.
M	number of service facilities in the network.
N	mean queue length including the one in service.
n	number of customers.
OPFMCQN	open product form multiple chain queueing networks
PFMQN	product form mixed queueing networks.
PS	processor sharing.
R	mean response time.
S	mean service time.
U	utilization.
V	visit ratio.
X	throughput.
λ	arrival rate.
ρ	traffic intensity.

Example: $S_i(M)$ means the mean service time of facility i for the main chain; $V_i(M)$ means the visit ratio to facility i due to the main chain; and $D_i(M) = S_i(M) * V_i(M)$ is the product of visit ratio and mean service time of facility i for the main chain.